This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Design of General Entropy-constrained Successively Refinable Unrestricted Polar Quantizer

Huihui Wu and Sorina Dumitrescu, *Senior Member, IEEE*

***Abstract*—This paper presents an algorithm for the optimal design of general entropy-constrained successively refinable unrestricted polar quantizer, i.e., with arbitrary number L of refinement levels, for bivariate circularly symmetric sources. The optimization problem is formulated as the minimization of a weighted sum of distortions and entropies for the scenario where the magnitude quantizers' thresholds are confined to a predefined finite set. The proposed solution algorithm is globally optimal. It involves L stages, where each stage corresponds to an unrestricted polar quantizer (UPQ) level, and includes solving the minimum-weight path problem for multiple node pairs in a series of weighted directed acyclic graphs. Additionally, we derive an upper bound $P_{\max}^{(l)},\ l \in [1:L]$, on the possible number of phase levels in any phase quantizer of the l-th level UPQ, which grows linearly with l. The time complexity of the proposed approach is $O(L^2K^3P_{\max}^{(1)})$, where K is the cardinality of the predefined set of possible magnitude thresholds. Finally, the experimental results for $L = 3$ demonstrate the effectiveness in practice of the proposed scheme.**

***Index Terms*—Successively refinable quantizer, entropy-constrained unrestricted polar quantization, globally optimal algorithm, minimum-weight path problem.**

I. Introduction

Polar coordinates are a natural choice when representing two-dimensional circularly symmetric probability densities. Accordingly, the quantization of bivariate sources with such densities can be performed in polar coordinates by the so-called polar quantizers. Specifically, a polar quantizer consists of two sequential scalar quantizers, i.e., a magnitude quantizer that may be nonuniform, and a uniform phase quantizer. There is a large body of literature focusing on the design of polar quantizers, such as [1]– [20] and references therein. Polar quantizers can be applied, for instance, in the encoding of discrete Fourier transform coefficients [1], [2], in holographic image processing [5], as well as in audio coding [6] for quantizing the sinusoid signals. Recently, the use of polar transmitters in wireless communication systems has gained increased attention [21]–[23], which also calls for the application of polar quantization in wireless receivers design [10], [12], [15].

In polar quantization, the phase quantizer can be designed either separately or dependently on the magnitude quantizer. The former one is called strictly polar quantizer (SPQ), while the latter one is known as unrestricted polar quantizer (UPQ). Note that the UPQ is of higher interest since it possesses better rate-distortion performance [3]. Most of the prior work on UPQ design focuses on the design of single-description fixed-rate UPQ (FRUPQ), aiming at minimizing the distortion for a fixed number of quantization regions. In particular, the design of uniform FRUPQ, i.e., where the magnitude quantizer is also uniform, was investigated in [13], [19]. The design of nonuniform FRUPQ was considered in [4], [8], [9], while the work [11] addresses the variance mismatch analysis for FRUPQ for Gaussian sources. Note that in the aforementioned literature, the optimal designs were derived using the asymptotic analysis, thus the optimality holds as the rate approaches infinity. The design of optimal FRUPQ for finite rates, i.e., without the high resolution assumption, was firstly conducted in [3], but the solution suffers from high complexity. Recently, in order to make the practical FRUPQ design more feasible for larger rates, an optimal algorithm based on dynamic programming was proposed in [18].

It is known that it is inefficient to assign equal length bitstreams to all quantized outputs (which is the case in FRUPQ), since they may have non-equal probabilities. Thus, Wilson applied entropy coding to the FRUPQ's output in [3], in order to improve its performance. On the other hand, he also pointed out that for optimal rate-distortion performance, the entropy-constrained optimization has to be accomplished, i.e. where the distortion is minimized with a constraint on the entropy, but he did not pursue it. The entropy-constrained UPQ (ECUPQ) design was addressed in [6] for the high rate scenario and in [17] for the finite rate case. The optimization criterion in both works is the minimization of a weighted sum of the distortion and entropy.

A successively refinable (or progressive/embedded/scalable) quantizer encodes the input signal into a base layer followed by several refinement (or enhancement) layers. The decoding of the base layer guarantees a coarser reconstruction of the signal, while the decoding of any additional enhancement layer gradually improves the reconstruction. This characteristic is useful for data transmission over the Internet and mobile networks, in order to maintain the quality of service when the bandwidth fluctuates in time due to network congestion and/or channel noise. For instance, in case of network congestion, the packets containing the last refinement layer can be dropped leading to only a slight decrease in the quality of reconstruction at the decoder. Thus, a successively refinable code enables a graceful degradation of the reconstruction when channel conditions deteriorate. On the other hand, if a non-successively refinable code is used, the loss of a portion of the bitstream may worsen dramatically the reconstruction of

This work was supported in part by an NSERC Discovery Grant.

H. Wu was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada. He is now with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (huihui.wu.phd@gmail.com).

S. Dumitrescu is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (e-mail: sorina@mail.ece.mcmaster.ca).

This article has been accepted for publication in a future issue of IEEE Transactions on Communications, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the signal. Much effort has been put into the investigation of successively refinable quantizers (see [24]– [31] and references therein), with applications in JPEG 2000 image compression standard [32], [33] and H. 264 scalable video coding standard [34], among others.

Therefore, it is of interest to study the design of successively refinable UPQs (SRUPQ). To the best of our knowledge, there are only two papers, [7] and [20], dealing with the practical design of SRUPQ. The authors of [7] address the design of fixed-rate SRUPQ with a general number L of refinement levels, where each component UPQ $Q^{(k)}$ consists of exactly 2^k quantization bins, in other words, the rate is $k/2$ bits/sample. The design of [7] employs a greedy algorithm. To be specific, $Q^{(1)}$ is the optimal two-level FRUPQ. Further, for each $k \geq 2$, after constructing $Q^{(1)}, \cdots, Q^{(k-1)}$, the FRUPQ $Q^{(k)}$, is obtained as the (asymptotically) best one-bit refinement of $Q^{(k-1)}$. Work [20] presents design algorithms for both fixed-rate SRUPQs (FR-SRUPQ) and entropy-constrained SRUPQs (EC-SRUPQ), but with only $L = 2$ refinement levels. In [20], the EC-SRUPQ design problem is formulated as the minimization of a weighted sum of distortions and entropies of UPQs at both levels for the case when the thresholds of the magnitude quantizers are restricted to some predefined finite sets and a globally optimal solution is proposed.

This work is concerned with the design of EC-SRUPQs with an arbitrary number L of refinement levels and is related to [20]. Therefore, we will review the EC-SRUPQ design algorithm proposed in [20] more closely. The solution algorithm extends ideas from the design of optimal single-level ECUPQ of [17], but is much more involved. In particular, the algorithm of [17] first finds the optimal number of phase levels for each possible cell of the magnitude quantizer, then solves the minimum-weight path (MWP) problem in a certain weighted directed acyclic graph (WDAG) in order to find the optimal partition of the magnitude quantizer. Adding one refinement level to the ECUPQ leads to the need of finding the best refinement of each possible quantization region of the coarse UPQ first, which is a problem similar in spirit to the problem of single-level ECUPQ design. This leads to the addition of another stage in the solution algorithm of [20], whose computationally dominant portion consists of solving the MWP problem between multiple pairs of nodes in multiple WDAGs. Each pair of nodes represents a possible magnitude cell of the coarse UPQ. Multiple WDAGs are needed since each of them is connected to one possible value of the number of phase levels of the phase quantizer corresponding to a magnitude cell in the coarse UPQ. Finally, this addition leads to the increase of the time complexity from $O(K^2)$ in the single-level case to $O(K^3 P_{\max}^{(1)})$ in the two-level case, where K is the size of the set of possible magnitude thresholds and $P_{\max}^{(1)}$ is an upper bound on the possible number of phase levels of the phase quantizers involved in the coarse UPQ.

However, it is worth pointing out that from [20], it is not clear how the solution approach can be generalized to solve the problem of optimal EC-SRUPQ design for higher values of L. It is also not clear how big the impact of adding refinement levels beyond two would be on the time complexity. The fact that adding just one refinement level increases the running time by a factor of $O(KP_{\max}^{(1)})$ naturally leads to the question whether such an increase should be expected for each additional level of refinement.

In this work, we settle the aforementioned two problems by presenting an algorithm for the design of an EC-SRUPQ with general number $L \geq 2$ of refinement levels, for which the algorithm of [20] is a special case corresponding to $L = 2$. Similarly to [20], we consider the scenario where the magnitude quantizer thresholds are confined to some predefined finite set and propose a globally optimal solution for this problem. Namely, we show that the solution algorithm for general L consists of L stages, where each stage is related to a UPQ level. The algorithm starts with the stage associated to the L-th level (i.e., the finest level) and proceeds in decreasing order of levels. The stage corresponding to each level $l, l > 1$, is similar in spirit to the stage corresponding to the finest level in the design of [20] and its time complexity is $O(K^3 P_{\max}^{(l-1)})$, where $P_{\max}^{(l-1)}$ is an upper bound on the possible number of phase levels of the phase quantizers at the $(l-1)$-th refinement level. The challenge here was to develop an upper bound $P_{\max}^{(l)}$ that does not increase exponentially with l. Interestingly, we found a value for $P_{\max}^{(l)}$ that increases at most linearly with l, i.e., it satisfies the inequality $P_{\max}^{(l)} \leq l P_{\max}^{(1)}$. This leads to the conclusion that the time complexity of the proposed algorithm is $O(L^2 K^3 P_{\max}^{(1)})$, i.e., only a factor of $O(L^2)$ higher than for the case of two levels.

The remaining of this paper is structured as follows. The following section introduces the notations, definitions and the problem formulation. In Section III, the major steps of the solution algorithm are described for the case of $L = 3$. Section IV shows how the process revealed in the previous section can be generalized to the case of arbitrary number of levels $L \geq 3$. Section V presents the details of each step of the proposed algorithm for general L. Finally, the experimental results are presented in Section VI, while Section VII concludes the paper.

II. Notation, Definitions and Problem Formulation

A. Notation and Definitions

Consider a bivariate random variable with the following circularly symmetric density, as a function of the polar coordinates r and θ,

$$p(r, \theta) = \frac{1}{2\pi} g(r), \ 0 \leq r < \infty, \ 0 \leq \theta < 2\pi.$$

Note that $g(r)$ is the marginal probability density function (pdf) of the magnitude variable, and the phase variable is uniformly distributed over the interval $[0, 2\pi)$. In addition, notice that the magnitude and phase variables are independent. An example of such a bivariate random variable is a two-dimensional memoryless Gaussian vector (X_1, X_2), with X_1 and X_2 following independent and identical marginal pdfs.

Let us first define, for any integer $n \geq 2$, an increasing n-sequence as any n-tuple $\mathbf{r} = (r_0, r_1, \cdots, r_{n-1})$, where $0 \leq r_0 < r_1 < \cdots < r_{n-2} < r_{n-1} \leq \infty$. Additionally, for any

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$n \geq 2$, $a \in [0, \infty)$ and $b \in (0, \infty]$, with $a < b$, let $\mathcal{S}_n(a, b)$ denote the set of all increasing n-sequences such that $r_0 = a$ and $r_{n-1} = b$.

For any integer $L \geq 2$, an EC-SRUPQ with L refinement levels is a sequence of L progressively refinable ECUPQs $\mathbf{Q}_L = (Q_1, Q_2, \cdots, Q_L)$, where Q_l is a refinement of Q_{l-1}, for $l \in [2 : L]$[1].

We will first discuss the notations for the coarse ECUPQ Q_1. Let M_1 denote its number of magnitude levels, while $\mathbf{r} = (r_0, r_1, \cdots, r_{M_1})$ denotes the increasing sequence of thresholds of its magnitude quantizer. Then the bins of the magnitude quantizer are $C_{i_1} = [r_{i_1 - 1}, r_{i_1})$, for $1 \leq i_1 \leq M_1$. Each magnitude quantizer is associated with a uniform phase quantizer, and we denote by P_{i_1} the number of phase regions of the phase quantizer associated to cell C_{i_1}, $1 \leq i_1 \leq M_1$. Note that $P_{i_1} \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ is the set of positive integers. Finally, each quantization bin of the ECUPQ Q_1 is given by

$$\mathcal{R}(i_1, k) = \left\{ re^{\jmath\theta} | r \in C_{i_1}, (k-1)\frac{2\pi}{P_{i_1}} \leq \theta < k\frac{2\pi}{P_{i_1}} \right\},$$

for some $1 \leq i_1 \leq M_1$ and $1 \leq k \leq P_{i_1}$, where $\jmath$ is the imaginary unit, i.e., $\jmath^2 = -1$. The total number of quantization bins of Q_1 is $N_1 = \sum_{i_1=1}^{M_1} P_{i_1}$. For each quantization bin of Q_1, the reconstructed magnitude-phase pair that minimizes the distortion (measured using the squared error) is $A_{i_1} e^{\jmath\theta_{i_1,k}}$ given by [1], [3]

$$\theta_{i_1,k} = (2k-1)\frac{\pi}{P_{i_1}}, \; A_{i_1} = \operatorname{sinc}\left(\frac{1}{P_{i_1}}\right) x(C_{i_1}),$$

where $\operatorname{sinc}\left(\frac{1}{P_{i_1}}\right) = \frac{\sin(\pi/P_{i_1})}{\pi/P_{i_1}}$, and for $C \subseteq [0, \infty)$, $x(C) = \frac{\int_C rg(r)dr}{\int_C g(r)dr}$.

For each l, $l \in [2 : L]$, ECUPQ Q_l is a refinement of Q_{l-1}. In other words, each magnitude cell C of Q_l is a subset of some magnitude cell C' of Q_{l-1}, and the number of phase levels of the phase quantizer corresponding to C is a multiple of the number of phase levels of the phase quantizer associated to C'. We will index the magnitude cells of Q_l using l-tuples of positive integers. We use the notation $\mathbf{i}_l$ for such an l-tuple, i.e., $\mathbf{i}_l = (i_1, i_2, \cdots, i_l)$. Given the l-tuple $\mathbf{i}_l = (i_1, i_2, \cdots, i_l)$, $\mathbf{i}_{l-1}$ denotes its $(l-1)$-length prefix, i.e., $\mathbf{i}_{l-1} = (i_1, i_2, \cdots, i_{l-1})$. The magnitude cell $C_{\mathbf{i}_l}$ of Q_l is a subset of the magnitude cell $C_{\mathbf{i}_{l-1}}$ of Q_{l-1}. More specifically, each cell $C_{\mathbf{i}_{l-1}}$ of the magnitude quantizer of Q_{l-1} is partitioned into $M_{l,\mathbf{i}_{l-1}}$ cells of the magnitude quantizer of Q_l.

Moreover, for any magnitude cell $C_{\mathbf{i}_l}$ of Q_l, $l \in [2 : L]$, let us denote by $\widetilde{P}_{\mathbf{i}_l}$ the number of phase regions of the phase quantizer associated to $C_{\mathbf{i}_l}$. As Q_l is a refinement of Q_{l-1}, $\widetilde{P}_{\mathbf{i}_l}$ must be a multiple of $\widetilde{P}_{\mathbf{i}_{l-1}}$, i.e., one has $\widetilde{P}_{\mathbf{i}_l} = P_{\mathbf{i}_l}\widetilde{P}_{\mathbf{i}_{l-1}}$, for some $P_{\mathbf{i}_l} \in \mathbb{Z}_+$, where $\widetilde{P}_{\mathbf{i}_1} = P_{i_1}$. It follows that $\widetilde{P}_{\mathbf{i}_l}$ can be computed by $\widetilde{P}_{\mathbf{i}_l} = \Pi_{j=1}^{l} P_{\mathbf{i}_j}$. Here we make the convention that $\mathbf{i}_j$, for $1 \leq j \leq l-1$, denotes the j-length prefix of the l-tuple $\mathbf{i}_l$, and $P_{\mathbf{i}_1} = P_{i_1}$. We will use this convention in the sequel without explicitly specifying it. Accordingly, each quantization bin of the refined ECUPQ Q_l can be represented as

$$\mathcal{R}(\mathbf{i}_l, k) = \left\{ re^{\jmath\theta} | r \in C_{\mathbf{i}_l}, (k-1)\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}} \leq \theta < k\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}} \right\},$$

for $1 \leq k \leq \widetilde{P}_{\mathbf{i}_l}$, $1 \leq i_1 \leq M_1$ and $1 \leq i_l \leq M_{l,\mathbf{i}_{l-1}}$ for $l \geq 2$. The total number of quantization bins of Q_l is thus $N_l = \sum_{\mathbf{i}_l} \widetilde{P}_{\mathbf{i}_l}$, where the summation is over all l-tuples $\mathbf{i}_l$ labeling the magnitude cells of Q_l. It follows that each quantization region $\mathcal{R}(\mathbf{i}_l, k)$ of UPQ Q_l is a subset of the quantization region $\mathcal{R}(\mathbf{i}_{l-1}, \lceil \frac{k}{P_{\mathbf{i}_l}} \rceil)$ of UPQ Q_{l-1}, where $\lceil \cdot \rceil$ denotes the ceiling function.

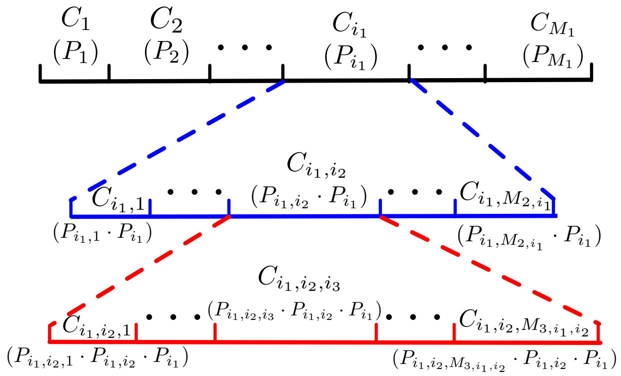

C_1
(P_1)
C_2
(P_2)
C_{i_1}
(P_{i_1})
C_{M_1}
(P_{M_1})
C_{i_1,i_2}
$(P_{i_1,i_2} \cdot P_{i_1})$
$C_{i_1,1}$
$(P_{i_1,1} \cdot P_{i_1})$
$C_{i_1,M_{2,i_1}}$
$(P_{i_1,M_{2,i_1}} \cdot P_{i_1})$
C_{i_1,i_2,i_3}
$(P_{i_1,i_2,i_3} \cdot P_{i_1,i_2} \cdot P_{i_1})$
$C_{i_1,i_2,1}$
$(P_{i_1,i_2,1} \cdot P_{i_1,i_2} \cdot P_{i_1})$
$C_{i_1,i_2,M_{3,i_1,i_2}}$
$(P_{i_1,i_2,M_{3,i_1,i_2}} \cdot P_{i_1,i_2} \cdot P_{i_1})$

Fig. 1. An illustration of the partitions of the magnitude quantizer for the case $L = 3$, as described in Section II.

To facilitate the understanding of the notations and structure of the EC-SRUPQ, we illustrate in Figure 1 the partitions of the magnitude quantizers for the case with $L = 3$ refinement levels. Notation C_{i_1}, $1 \leq i_1 \leq M_1$, is used for the magnitude cell $[r_{i_1-1}, r_{i_1})$ of the coarse ECUPQ Q_1, while P_{i_1} represents the number of phase levels of the phase quantizer associated to C_{i_1}. Further, the cell C_{i_1} is partitioned into M_{2,i_1} refined cells of the magnitude quantizer of Q_2. Each such refined magnitude bin is denoted by C_{i_1,i_2}, $1 \leq i_2 \leq M_{2,i_1}$. The corresponding number of phase regions for this bin is then $\widetilde{P}_{i_1,i_2} = P_{i_1,i_2} P_{i_1}$. Similarly, the cell C_{i_1,i_2} is divided into M_{3,i_1,i_2} bins of the magnitude quantizer of Q_3. We denote by C_{i_1,i_2,i_3}, $1 \leq i_3 \leq M_{3,i_1,i_2}$, each such refined magnitude bin and by $\widetilde{P}_{i_1,i_2,i_3} = P_{i_1,i_2,i_3} P_{i_1,i_2} P_{i_1}$ the corresponding number of phase regions. In addition, Figure 3 in Section VI shows the quantization regions of an actual EC-SRUPQ with 3 levels obtained with the optimization algorithm proposed in this work.

The squared error is utilized as the distortion measure. As a consequence, the expected distortion (per sample) of the

[1] Note that technically there are only $L-1$ levels of refinement, since level 1 can be considered as the base level. However, since the total number of levels, i.e., component UPQs, is L, we refer to such an EC-SRUPQ as having L refinement levels.

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

ECUPQ $Q_l,\ l \in [1:L]$, can be expressed as [1], [3]

$$\begin{aligned} &D(Q_l) \\ &= \frac{1}{2}\sum_{\mathbf{i}_l}\sum_{k'=1}^{\widetilde{P}_{\mathbf{i}_l}}\int_{C_{\mathbf{i}_l}}\int_{(k'-1)\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}}}^{k'\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}}} ||re^{\jmath\theta} - A_{\mathbf{i}_l}e^{\jmath\theta_{\mathbf{i}_l,k'}}||^2\ p(r,\theta)\ d\theta dr \\ &= \frac{1}{2}\sum_{\mathbf{i}_l}\int_{C_{\mathbf{i}_l}}\left\{\sum_{k'=1}^{\widetilde{P}_{\mathbf{i}_l}}\int_{(k'-1)\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}}}^{k'\frac{2\pi}{\widetilde{P}_{\mathbf{i}_l}}}\Big[r^2 - 2rA_{\mathbf{i}_l}\cos(\theta - \theta_{\mathbf{i}_l,k'}) \right. \\ &\qquad \left. + A_{\mathbf{i}_l}^2\Big]\ \frac{g(r)}{2\pi}\ d\theta\right\}dr, \end{aligned} \tag{1}$$

where the outer summation is over all l-tuples $\mathbf{i}_l$ labeling the magnitude cells of Q_l. According to [1], [3], for each bin $\mathcal{R}(\mathbf{i}_l, k')$, the reconstructed magnitude-phase pair that minimizes the distortion is

$$A_{\mathbf{i}_l} = \operatorname{sinc}\left(\frac{1}{\widetilde{P}_{\mathbf{i}_l}}\right)x(C_{\mathbf{i}_l}),\ \theta_{\mathbf{i}_l,k'} = (2k'-1)\frac{\pi}{\widetilde{P}_{\mathbf{i}_l}}. \tag{2}$$

Substituting (2) into (1), we obtain the following simplified expression

$$\begin{aligned} D(Q_l) &= \frac{1}{2}\left(\sum_{\mathbf{i}_l}\int_{C_{\mathbf{i}_l}} r^2 g(r)dr - \sum_{\mathbf{i}_l} A_{\mathbf{i}_l}^2 q(C_{\mathbf{i}_l})\right) \\ &= \frac{1}{2}\left(\int_0^{+\infty} r^2 g(r)dr - \sum_{\mathbf{i}_l} A_{\mathbf{i}_l}^2 q(C_{\mathbf{i}_l})\right), \end{aligned} \tag{3}$$

where, for $C \subseteq [0,\infty)$, $q(C) = \int_C g(r)dr$.

The encoder generates a bitstream with L refinement layers as follows. Each input pair (r,θ) is quantized using the UPQ Q_L, i.e., the region $\mathcal{R}(\mathbf{i}_L, k)$ satisfying $(r,\theta) \in \mathcal{R}(\mathbf{i}_L, k)$ is determined. Next the L-tuple $(k_1,\cdots,k_L)$ satisfying $k_L = k$ and $\mathcal{R}(\mathbf{i}_L, k_L) \subseteq \mathcal{R}(\mathbf{i}_{L-1}, k_{L-1}) \subseteq \cdots \subseteq \mathcal{R}(\mathbf{i}_1, k_1)$ is computed. The first layer is obtained by applying entropy coding to the pairs $(\mathbf{i}_1, k_1)$. Further, for each $l, l \in [2:L]$, the l-th refinement layer is generated by encoding the pairs $(\mathbf{i}_l, k_l)$ conditionally on $(\mathbf{i}_{l-1}, k_{l-1})$ also using an entropy coder. Since practical entropy coders, such as the arithmetic coder or the block Huffman coder, are able to approach the entropy of the random variable being encoded as the block dimension increases to infinity, we assume that the bitrate (per sample) of the l-th refinement layer equals $\frac{1}{2}H(\mathbf{I}_l, K_l|\mathbf{I}_{l-1}, K_{l-1})$, where $\mathbf{I}_l$ denotes the random vector representing the l-tuple $\mathbf{i}_l$, K_l denotes the random variable representing the integer k_l, and $H(\cdot|\cdot)$ denotes the conditional entropy function[2]. For each $l, l \in [1:L]$, denote by $R(Q_l)$ the bitrate (per sample) of the prefix formed of the first l layers. It follows that $R(Q_l)$ equals half of the entropy of the output of UPQ Q_l, i.e.,

$$\begin{aligned} R(Q_l) &= \frac{1}{2}H(\mathbf{I}_l, K_l) = \frac{1}{2}\left(H(\mathbf{I}_l) + H(K_l|\mathbf{I}_l)\right) \\ &= \frac{1}{2}\sum_{\mathbf{i}_l} q(C_{\mathbf{i}_l})\left(-\log_2 q(C_{\mathbf{i}_l}) + \log_2(\widetilde{P}_{\mathbf{i}_l})\right), \end{aligned} \tag{4}$$

where the summation is over all l-tuples $\mathbf{i}_l$ labeling the magnitude cells of Q_l.

We will further assume that the thresholds of the magnitude quantizer at each level take values in some predefined set $\mathcal{A} = \{a_1,\cdots,a_K\}$. In practice, this set can be obtained by finely discretizing the interval $[0,B]$, for some B chosen such that the probability that $r \notin [0,B]$ is sufficiently small. Assume that the elements of $\mathcal{A}$ are labeled in increasing order, i.e., $a_i < a_{i+1}$, for $1 \le i \le K-1$. Additionally, let us denote $a_0 = 0$, $a_{K+1} = \infty$ and $\bar{\mathcal{A}} = \mathcal{A} \cup \{a_0, a_{K+1}\}$. We emphasize that the algorithm proposed in this work can be easily generalized to the case where the predefined sets of possible thresholds of the magnitude quantizer of Q_l is a subset of that for Q_{l+1}, for $l \in [1:L-1]$.

B. Problem Formulation

We are interested in the minimization of a weighted sum of distortions and rates. Therefore, the cost in the optimization problem is

$$\mathcal{L}(\mathbf{Q}_L) \triangleq \sum_{l=1}^{L}\Big[\phi_l D(Q_l) + \lambda_l R(Q_l)\Big], \tag{5}$$

where $0 \le \phi_l \le 1$, $\lambda_l > 0$, for $l \in [1:L]$, and $\sum_{l=1}^{L}\phi_l = 1$. Let us denote by $\mathfrak{Q}_L(\mathcal{A})$ the set of all EC-SRUPQs with the magnitude quantizers' thresholds taken from the set $\bar{\mathcal{A}}$. We formulate the problem of optimal design of an EC-SRUPQ with L refinement levels as follows

$$\min_{\mathbf{Q}_L \in \mathfrak{Q}_L(\mathcal{A})} \mathcal{L}(\mathbf{Q}_L). \tag{6}$$

It is known [37], [38] that the solution to problem (6) corresponds to an EC-SRUPQ whose $2L$-tuple of rates and distortions $(R(Q_1),\cdots,R(Q_L),D(Q_1),\cdots,D(Q_L))$ lies on the lower boundary of the convex hull of the set of all such $2L$-tuples obtained when $\mathbf{Q}_L \in \mathfrak{Q}_L(\mathcal{A})$.

III. Proposed EC-SRUPQ Design when $L = 3$

In order to facilitate the understanding of the proposed solution to the optimization problem (6) for general L, we first treat the simpler case when $L = 3$. This section describes explicitly the major steps of the proposed design algorithm for the case of $L = 3$ refinement levels.

When $L = 3$, the optimal EC-SRUPQ design problem is

$$\min_{\mathbf{Q}_3 \in \mathfrak{Q}_3(\mathcal{A})} \mathcal{L}(\mathbf{Q}_3). \tag{7}$$

Notice that the first term in (3) is always constant, and hence it can be removed from the cost function (7) for each $l, l \in [1:3]$. Further, by substituting relations (3) and (4) in (7), the problem (7) becomes equivalent to minimizing the cost $\mathcal{F}(\mathbf{Q}_3)$, which is given in (8) at the top of the next page.

Let us assume that we know some integers $P_{\max}^{(1)}$ and $P_{\max}^{(2)}$, such that there is an optimal EC-SRUPQ satisfying

$$\begin{gathered} P_{i_1} \le P_{\max}^{(1)} \text{ and } P_{i_1}P_{i_1,i_2} \le P_{\max}^{(2)}, \\ \text{for any } 1 \le i_1 \le M_1 \text{ and } 1 \le i_2 \le M_{2,i_1}. \end{gathered} \tag{9}$$

By examining the cost function $\mathcal{F}(\mathbf{Q}_3)$, it can be noticed that for each triplet $\mathbf{i}_3 = (i_1, i_2, i_3)$, the variable $P_{\mathbf{i}_3}$ only

[2] Note that the assumption that the rate of an entropy-constrained quantizer equals the entropy of the quantized output divided by the quantizer dimension is very common in the entropy-constrained quantizer design literature [6], [17], [20], [35], [36].

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$$\begin{aligned}\mathcal{F}(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1} \Bigg\{ & \underbrace{q(C_{i_1})\Bigg(- \phi_1 \ \mathrm{sinc}^2\left(\frac{1}{P_{i_1}}\right) x^2(C_{i_1}) - \lambda_1 \log_2 q(C_{i_1}) + (\lambda_1+\lambda_2+\lambda_3)\log_2 P_{i_1}\Bigg)}_{\varphi_1(C_{i_1},P_{i_1})} + \\ \sum_{i_2=1}^{M_{2,i_1}} \Bigg[& \underbrace{q(C_{i_1,i_2})\Bigg(- \phi_2 \ \mathrm{sinc}^2\left(\frac{1}{P_{i_1}P_{i_1,i_2}}\right) x^2(C_{i_1,i_2}) - \lambda_2 \log_2 q(C_{i_1,i_2}) + (\lambda_2+\lambda_3)\log_2 P_{i_1,i_2}\Bigg)}_{\varphi_2(C_{i_1,i_2},P_{i_1},P_{i_1,i_2})} + \\ & \sum_{i_3=1}^{M_{3,i_1,i_2}} \underbrace{q(C_{i_1,i_2,i_3})\Bigg(- \phi_3 \ \mathrm{sinc}^2\left(\frac{1}{P_{i_1}P_{i_1,i_2}P_{i_1,i_2,i_3}}\right) x^2(C_{i_1,i_2,i_3}) - \lambda_3 \log_2 q(C_{i_1,i_2,i_3}) + \lambda_3\log_2 P_{i_1,i_2,i_3}\Bigg)}_{\varphi_3(C_{i_1,i_2,i_3},P_{i_1}\cdot P_{i_1,i_2},P_{i_1,i_2,i_3})} \Bigg]\Bigg\}. \end{aligned} \tag{8}$$

appears in the term $\varphi_3(C_{i_1,i_2,i_3}, P_{i_1} \cdot P_{i_1,i_2}, P_{i_1,i_2,i_3})$. Therefore, P_{i_1,i_2,i_3} can be optimized separately for fixed C_{i_1,i_2,i_3} and $\widetilde{P}_{i_1,i_2} = P_{i_1}P_{i_1,i_2}$. Note that the final C_{i_1,i_2,i_3} and $\widetilde{P}_{i_1,i_2}$ are not known in advance. Nevertheless, we can compute the optimal P_{i_1,i_2,i_3} for each possible choice of C_{i_1,i_2,i_3}, i.e., for each interval $[c, d)$, $c, d \in \bar{\mathcal{A}}$, $c < d$, and for each possible $\widetilde{P}_{i_1,i_2}$, i.e., for each positive integer $P \leq P^{(2)}_{\max}$. Let us denote by $P^*_3([c, d), P)$ the optimal P_{i_1,i_2,i_3}, namely

$$P^*_3([c, d), P) = \arg \min_{P' \in \mathbb{Z}_+} \varphi_3([c, d), P, P'), \tag{10}$$

where the smallest minimizer is taken if there are more solutions. Notice that the problem (10) is identical to problem (10) in [20] up to a change of parameters. Thus, according to [20, Proposition 1], there is a finite integer achieving the minimum in (10). Further, let

$$\varphi^*_3([c, d), P) = \varphi_3([c, d), P, P^*_3([c, d), P)), \tag{11}$$

for each c, d and P as in (10).

Next, by replacing P_{i_1,i_2,i_3} in $\mathcal{F}(\mathbf{Q}_3)$ with $P^*_3(C_{i_1,i_2,i_3}, \widetilde{P}_{i_1,i_2})$, for each $1 \leq i_1 \leq M_1$, $1 \leq i_2 \leq M_{2,i_1}$ and $1 \leq i_3 \leq M_{3,i_1,i_2}$, we obtain the following cost function

$$\begin{aligned}\mathcal{F}_{3,1}(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1} \Bigg\{ & \varphi_1(C_{i_1}, P_{i_1}) + \\ \sum_{i_2=1}^{M_{2,i_1}} \Bigg[& \varphi_2(C_{i_1,i_2}, P_{i_1}, P_{i_1,i_2}) + \sum_{i_3=1}^{M_{3,i_1,i_2}} \varphi^*_3(C_{i_1,i_2,i_3}, \widetilde{P}_{i_1,i_2})\Bigg]\Bigg\}.\end{aligned}$$

As $\mathcal{F}(\mathbf{Q}_3) \geq \mathcal{F}_{3,1}(\mathbf{Q}_3)$, it follows that the problem (7) is equivalent to minimizing $\mathcal{F}_{3,1}(\mathbf{Q}_3)$. It can also be noticed that, if the values $\varphi^*_3(C_{i_1,i_2,i_3}, \widetilde{P}_{i_1,i_2})$ are known for each possible pair $(C_{i_1,i_2,i_3}, \widetilde{P}_{i_1,i_2})$, then the partition of C_{i_1,i_2} into cells C_{i_1,i_2,i_3} can be optimized separately for each pair $(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})$. We will then denote by $\mathbf{r}^*_3(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})$ this optimal partition. The optimal partition can be computed for each possible magnitude cell C_{i_1,i_2}, i.e., for each interval $[c, d)$, $c, d \in \bar{\mathcal{A}}$, $c < d$, and for each positive integer $P \leq P^{(2)}_{\max}$, where P represents the value $\widetilde{P}_{i_1,i_2}$. In other words, we compute

$$\mathbf{r}^*_3([c, d), P) = \arg \min_{M,\mathbf{r}} \sum_{i=1}^{M} \varphi^*_3([r_{i-1}, r_i), P), \tag{12}$$

where $\mathbf{r} = (r_0, \cdots, r_M) \in \mathcal{S}_{M+1}(c, d) \cap \bar{\mathcal{A}}^{M+1}$. Then, let us denote by $\tau^*_2([c, d), P)$ the cost obtained at optimality in (12), i.e.,

$$\tau^*_2([c, d), P) = \sum_{i=1}^{M^*} \varphi^*_3([r^*_{i-1}, r^*_i), P), \tag{13}$$

where $\mathbf{r}^*_3([c, d), P) = (r^*_0, \cdots, r^*_{M^*})$. Subsequently, by replacing, for each (i_1, i_2), the partition of C_{i_1,i_2} in $\mathcal{F}_{3,1}(\mathbf{Q}_3)$ with the optimal partition $\mathbf{r}^*_3(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})$, a new cost function is obtained, which depends only on the ECUPQs at levels 1 and 2, namely

$$\begin{aligned}\mathcal{F}_{3,2}(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1} \Bigg\{ & \varphi_1(C_{i_1}, P_{i_1}) + \\ \sum_{i_2=1}^{M_{2,i_1}} \Bigg[& \varphi_2(C_{i_1,i_2}, P_{i_1}, P_{i_1,i_2}) + \tau^*_2(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})\Bigg]\Bigg\}.\end{aligned}$$

Now it can be observed that, if the values $\tau^*_2(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})$ are known for all possible pairs $(C_{i_1,i_2}, \widetilde{P}_{i_1,i_2})$, then the optimal P_{i_1,i_2} can be found independently for each possible pair (C_{i_1,i_2}, P_{i_1}). Similarly to the evaluation of the optimal P_{i_1,i_2,i_3}, the optimal P_{i_1,i_2} will be computed for each possible cell C_{i_1,i_2}, i.e., for each interval $[c, d)$, $c, d \in \bar{\mathcal{A}}$, $c < d$, and for each possible choice of P_{i_1}, i.e., for each positive integer $P \leq P^{(1)}_{\max}$. This optimal P_{i_1,i_2} will be denoted by $P^*_2([c, d), P)$ and computed as follows

$$\begin{aligned}& P^*_2([c, d), P) = \\ & \quad \arg \min_{\substack{P' \in \mathbb{Z}_+ \\ P' \leq \frac{P^{(2)}_{\max}}{P}}} \Big[\varphi_2([c, d), P, P') + \tau^*_2([c, d), P \cdot P')\Big],\end{aligned} \tag{14}$$

where the smallest one is taken if there are multiple minimizers. Recall that the term $\tau^*_2([c, d), P \cdot P')$ has already been

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

computed by (13). Further, let

$$\varphi_2^*([c,d),P) = \varphi_2([c,d),P,P_2^*([c,d),P)) + \tau_2^*([c,d),P\cdot P_2^*([c,d),P)). \tag{15}$$

We then replace in $\mathcal{F}_{3,2}(\mathbf{Q}_3)$ each P_{i_1,i_2} by the optimum $P_2^*(C_{i_1,i_2},P_{i_1})$, for each $1\le i_1\le M_1$ and $1\le i_2\le M_{2,i_1}$, and thus obtain

$$\mathcal{F}_{2,1}(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1}\left[\varphi_1(C_{i_1},P_{i_1}) + \sum_{i_2=1}^{M_{2,i_1}}\varphi_2^*(C_{i_1,i_2},P_{i_1})\right].$$

We conclude that the problem (7) can be further reduced to minimizing $\mathcal{F}_{2,1}(\mathbf{Q}_3)$, as $\mathcal{F}_{3,2}(\mathbf{Q}_3)\ge\mathcal{F}_{2,1}(\mathbf{Q}_3)$. Now it can be seen that, if the values $\varphi_2^*(C_{i_1,i_2},P_{i_1})$ are already known for each possible pair (C_{i_1,i_2},P_{i_1}), then the refined partition of C_{i_1} into cells C_{i_1,i_2} can be optimized separately for each pair (C_{i_1},P_{i_1}). Thus, we can determine this optimal partition for each possible choice of C_{i_1}, i.e., for each interval $[c,d)$ with $c,d\in\bar{\mathcal{A}}$, $c<d$, and for each possible choice of P_{i_1}, i.e., for each positive integer $P\le P_{\max}^{(1)}$. This optimal partition will be denoted by $\mathbf{r}_2^*(C_{i_1},P_{i_1})$, i.e.,

$$\mathbf{r}_2^*([c,d),P) = \arg\min_{M,\mathbf{r}}\sum_{i=1}^{M}\varphi_2^*([r_{i-1},r_i),P), \tag{16}$$

where $\mathbf{r}=(r_0,\cdots,r_M)\in\mathcal{S}_{M+1}(c,d)\cap\bar{\mathcal{A}}^{M+1}$. Further, the cost obtained at optimality in (16) is

$$\tau_1^*([c,d),P) = \sum_{i=1}^{M^*}\varphi_2^*([r_{i-1}^*,r_i^*),P), \tag{17}$$

where $\mathbf{r}_2^*([c,d),P) = (r_0^*,\cdots,r_{M^*}^*)$. At the next step, we replace, for each i_1, the partition of C_{i_1} in $\mathcal{F}_{2,1}(\mathbf{Q}_3)$ with the optimum $\mathbf{r}_2^*(C_{i_1},P_{i_1})$, and obtain a new cost as a function of only C_{i_1} and P_{i_1}, namely

$$\mathcal{F}_{2,2}(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1}\Big[\varphi_1(C_{i_1},P_{i_1}) + \tau_1^*(C_{i_1},P_{i_1})\Big].$$

Now it can be pointed out that, if the values $\tau_1^*(C_{i_1},P_{i_1})$ are known for all possible pairs (C_{i_1},P_{i_1}), then the optimal P_{i_1} can be evaluated independently for each C_{i_1}. As a consequence, we can compute the optimal P_{i_1} for each possible cell C_{i_1}, i.e., for each interval $[c,d)$, $c,d\in\bar{\mathcal{A}}$, $c<d$. We further denote by $P_1^*([c,d))$ this optimal P_{i_1}, i.e.,

$$P_1^*([c,d)) = \arg\min_{\substack{P\in\mathbb{Z}_+\\ P\le P_{\max}^{(1)}}}\Big[\varphi_1([c,d),P) + \tau_1^*([c,d),P)\Big], \tag{18}$$

where the smallest one is taken if there are multiple minimizers. Finally, by replacing P_{i_1} in $\mathcal{F}_{2,2}(\mathbf{Q}_3)$ with $P_1^*(C_{i_1})$, a new cost function is obtained, which depends only on C_{i_1}, namely

$$\mathcal{F}_1(\mathbf{Q}_3) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1}\left[\varphi_1\Big(C_{i_1},P_1^*(C_{i_1})\Big) + \tau_1^*\Big(C_{i_1},P_1^*(C_{i_1})\Big)\right].$$

Therefore, the optimization problem (7) reduces to

$$\begin{aligned}&\min_{M_1,\mathbf{r}} \quad \mathcal{F}_1(\mathbf{Q}_3)\\ &\text{subject to} \quad \mathbf{r}\in\mathcal{S}_{M_1+1}(0,\infty)\cap\bar{\mathcal{A}}^{M_1+1}.\end{aligned} \tag{19}$$

The above discussion suggests the following procedure to solve the problem (7).

i) Determine some integers $P_{\max}^{(1)}$ and $P_{\max}^{(2)}$ such that there is an optimal EC-SRUPQ satisfying condition (9).

ii) For each pair $(c,d)\in\bar{\mathcal{A}}^2$, with $c<d$, and any positive integer $P\le P_{\max}^{(2)}$, evaluate $P_3^*([c,d),P)$ according to (10).

iii) For each pair $(c,d)\in\bar{\mathcal{A}}^2$, with $c<d$, and each positive integer $P\le P_{\max}^{(2)}$, compute the optimal partition $\mathbf{r}_3^*([c,d),P)$ defined in (12) and the corresponding cost $\tau_2^*([c,d),P)$ given in (13).

iv) For each pair $(c,d)\in\bar{\mathcal{A}}^2$, with $c<d$, and each positive integer $P\le P_{\max}^{(1)}$, evaluate $P_2^*([c,d),P)$ according to (14).

v) For each pair $(c,d)\in\bar{\mathcal{A}}^2$, with $c<d$, and each positive integer $P\le P_{\max}^{(1)}$, compute the best partition $\mathbf{r}_2^*([c,d),P)$ defined in (16) and the corresponding cost $\tau_1^*([c,d),P)$ given in (17).

vi) For each pair $(c,d)\in\bar{\mathcal{A}}^2$, with $c<d$, compute $P_1^*([c,d))$ according to (18).

vii) Solve the problem (19).

The detailed solutions for each step will be described in the context of general L in Section V. In the following section, we explain how the above procedure can be generalized to find the optimal EC-SRUPQ for any $L\ge 3$.

IV. Proposed EC-SRUPQ Design for General L

By removing the constant terms in (3) and substituting relations (2), (3) and (4) into the cost (5), the problem (6) becomes equivalent to minimizing $\mathcal{F}'(\mathbf{Q}_L)$, which is shown in equation (20) at the top of next page.

We will assume that we know some integers $P_{\max}^{(l)}$, for $l\in[1:L-1]$, such that there is an optimal EC-SRUPQ satisfying

$$\begin{aligned}\widetilde{P}_{\mathbf{i}_l}\le P_{\max}^{(l)},\ &\text{for any } \mathbf{i}_l=(i_1,\cdots,i_l),\\ 1\le i_1\le M_1,\ &1\le i_k\le M_{k,\mathbf{i}_{k-1}},\\ 1\le k\le l,\ &l\in[1:L-1],\end{aligned} \tag{21}$$

where $\widetilde{P}_{\mathbf{i}_1}=P_{i_1}$. It can be noted from (20) that for each L-tuple $\mathbf{i}_L$, the variable $P_{\mathbf{i}_L}$ appears only in the last term $\varphi_L(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}},P_{\mathbf{i}_L})$. Consequently, $P_{\mathbf{i}_L}$ can be optimized separately for fixed cell $C_{\mathbf{i}_L}$ and positive integer P representing the value $\widetilde{P}_{\mathbf{i}_{L-1}}=\Pi_{j=1}^{L-1}P_{\mathbf{i}_j}$. In order to achieve this, we need to evaluate the optimal $P_{\mathbf{i}_L}$ for each possible choice of $C_{\mathbf{i}_L}$, i.e., for each interval $[c,d)$, where $c,d\in\bar{\mathcal{A}}$, with $c<d$, and for each positive integer $P\le P_{\max}^{(L-1)}$. Then the optimal $P_{\mathbf{i}_L}$ can be obtained as

$$P_L^*([c,d),P) = \arg\min_{P'\in\mathbb{Z}_+}\varphi_L([c,d),P,P'), \tag{22}$$

where the smallest value is chosen in case of multiple minimizers. The above problem is identical to problem (10) in [20] up to a change of parameters. Thus, according to [20, Proposition

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$$\begin{aligned}\mathcal{F}'(\mathbf{Q}_L) = \frac{1}{2}\sum_{i_1=1}^{M_1} \Bigg(& \underbrace{q(C_{i_1})\Big(-\phi_1 \ \mathrm{sinc}^2\left(\frac{1}{P_{i_1}}\right) x^2(C_{i_1}) - \lambda_1 \log_2 q(C_{i_1}) + (\sum_{j=1}^{L}\lambda_j)\log_2 P_{i_1}\Big)}_{\varphi_1(C_{i_1},P_{i_1})} + \cdots \\ & + \sum_{i_l=1}^{M_{l,\mathbf{i}_{l-1}}} \Bigg(\underbrace{q(C_{\mathbf{i}_l})\Big(-\phi_l \ \mathrm{sinc}^2\left(\frac{1}{P_{\mathbf{i}_l}\cdot \widetilde{P}_{\mathbf{i}_{l-1}}}\right) x^2(C_{\mathbf{i}_l}) - \lambda_l \log_2 q(C_{\mathbf{i}_l}) + (\sum_{j=l}^{L}\lambda_j)\log_2 P_{\mathbf{i}_l}\Big)}_{\varphi_l(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}},P_{\mathbf{i}_l})} + \cdots \\ & + \sum_{i_L=1}^{M_{L,\mathbf{i}_{L-1}}} \Bigg(\underbrace{q(C_{\mathbf{i}_L})\Big(-\phi_L \ \mathrm{sinc}^2\left(\frac{1}{P_{\mathbf{i}_L}\cdot \widetilde{P}_{\mathbf{i}_{L-1}}}\right) x^2(C_{\mathbf{i}_L}) - \lambda_L \log_2 q(C_{\mathbf{i}_L}) + \lambda_L\log_2 P_{\mathbf{i}_L}\Big)}_{\varphi_L(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}},P_{\mathbf{i}_L})} \quad \underbrace{\Bigg)\cdots\Bigg)}_{L \text{ parentheses}} \quad . \end{aligned} \tag{20}$$

1], we conclude that it has a finite solution. Additionally, we denote

$$\varphi_L^*([c,d),P) = \varphi_L([c,d),P,P_L^*([c,d),P)). \tag{23}$$

Further, we replace $P_{\mathbf{i}_L}$ in $\mathcal{F}'(\mathbf{Q}_L)$ with its optimal value $P_L^*(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}})$, for each $\mathbf{i}_L$. Then the following cost function is obtained,

$$\begin{aligned}&\mathcal{F}'_{L,1}(\mathbf{Q}_L) \triangleq \\ &\frac{1}{2}\sum_{i_1=1}^{M_1}\Bigg(\varphi_1(C_{i_1},P_{i_1}) + \cdots + \sum_{i_l=1}^{M_{l,\mathbf{i}_{l-1}}}\Bigg(\varphi_l(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}},P_{\mathbf{i}_l}) + \\ &\cdots + \sum_{i_{L-1}=1}^{M_{L-1,\mathbf{i}_{L-2}}}\Bigg(\varphi_{L-1}(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-2}},P_{\mathbf{i}_{L-1}}) + \\ &\sum_{i_L=1}^{M_{L,\mathbf{i}_{L-1}}}\Bigg(\varphi_L^*(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}}) \quad \underbrace{\Bigg)\cdots\Bigg)}_{L \text{ parentheses}} \quad .\end{aligned}$$

Since $\mathcal{F}'(\mathbf{Q}_L) \geq \mathcal{F}'_{L,1}(\mathbf{Q}_L)$, it follows that the problem (6) is equivalent to minimizing $\mathcal{F}'_{L,1}(\mathbf{Q}_L)$. Further, if the values $\varphi_L^*(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}})$ are known for each possible pair $(C_{\mathbf{i}_L},\widetilde{P}_{\mathbf{i}_{L-1}})$, then the refined partition of $C_{\mathbf{i}_{L-1}}$ into cells $C_{\mathbf{i}_L}$ can be optimized separately for each pair $(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-1}})$. This optimal partition will be denoted by $\mathbf{r}_L^*(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-1}})$. The optimal partition can be computed for each possible magnitude level $C_{\mathbf{i}_{L-1}}$, i.e., for each interval $[c,d)$, where $c,d \in \bar{\mathcal{A}}$, with $c < d$, and each possible positive integer $P \leq P_{\max}^{(L-1)}$, representing a possible value of $\widetilde{P}_{\mathbf{i}_{L-1}}$. In other words, we obtain

$$\mathbf{r}_L^*([c,d),P) = \arg\min_{M,\mathbf{r}} \sum_{i=1}^{M} \varphi_L^*([r_{i-1},r_i),P), \tag{24}$$

where $\mathbf{r} = (r_0,\cdots,r_M) \in \mathcal{S}_{M+1}(c,d) \cap \bar{\mathcal{A}}^{M+1}$. Let us further denote by $\tau_{L-1}^*([c,d),P)$ the cost obtained at optimality in (24), i.e.,

$$\tau_{L-1}^*([c,d),P) = \sum_{i=1}^{M^*} \varphi_L^*([r_{i-1}^*,r_i^*),P), \tag{25}$$

where $\mathbf{r}_L^*([c,d),P) = (r_0^*,\cdots,r_{M^*}^*)$.

Subsequently, by replacing, for each $\mathbf{i}_{L-1}$, the partition of $C_{\mathbf{i}_{L-1}}$ in $\mathcal{F}'_{L,1}(\mathbf{Q}_L)$ with the optimal partition $\mathbf{r}_L^*(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-1}})$, we obtain the new cost function $\mathcal{F}'_{L,2}(\mathbf{Q}_L)$ presented in equation (26) on next page, which depends only on the EC-SRUPQs levels 1 through $L-1$.

Let us assume now that $L \geq 3$. We will continue the description of the process recursively. Namely, let us fix some $l, l \in [2 : L-1]$, and assume that for each $k, l < k \leq L$, each positive integer $P \leq P_{\max}^{(k-1)}$ and each pair $(c,d) \in \bar{\mathcal{A}}^2$ with $c < d$, we have defined the quantities $P_k^*([c,d),P)$, $\mathbf{r}_k^*([c,d),P)$ and $\tau_{k-1}^*([c,d),P)$. The current cost function is $\mathcal{F}'_{l+1,2}(\mathbf{Q}_L)$ presented in (27) at the top of next page, which depends only on the EC-SRUPQs levels 1 through l. Thus, solving the problem (6) is equivalent to minimizing the cost in (27). By analyzing this cost, we observe that the optimal $P_{\mathbf{i}_l}$ can be computed separately for fixed cell $C_{\mathbf{i}_l}$ and fixed product $\widetilde{P}_{\mathbf{i}_{l-1}}$. We will compute it for each possible cell $C_{\mathbf{i}_l}$, i.e., for each interval $[c,d)$, $c,d \in \bar{\mathcal{A}}$, $c < d$, and for each positive integer $P \leq P_{\max}^{(l-1)}$, representing a possible value $\widetilde{P}_{\mathbf{i}_{l-1}}$, as follows

$$P_l^*([c,d),P) = \arg \min_{\substack{P' \in \mathbb{Z}_+ \\ P' \leq \frac{P_{\max}^{(l)}}{P}}} \Big(\varphi_l([c,d),P,P') + \tau_l^*([c,d),P\cdot P')\Big), \tag{28}$$

where the smallest value is chosen in case of multiple minimizers. Further, we denote

$$\begin{aligned}&\varphi_l^*([c,d),P) = \\ &\quad \varphi_l([c,d),P,P_l^*([c,d),P)) + \tau_l^*([c,d),P\cdot P_l^*([c,d),P)).\end{aligned} \tag{29}$$

By replacing the variable $P_{\mathbf{i}_l}$ in $\mathcal{F}'_{l+1,2}(\mathbf{Q}_L)$ with the optimum $P_l^*(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}})$, for each $\mathbf{i}_l$, we obtain the following cost

$$\begin{aligned}&\mathcal{F}'_{l,1}(\mathbf{Q}_L) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1}\Bigg(\varphi_1(C_{i_1},P_{i_1}) + \cdots + \sum_{i_{l-1}=1}^{M_{l-1,\mathbf{i}_{l-2}}}\Bigg(\\ &\varphi_{l-1}(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-2}},P_{\mathbf{i}_{l-1}}) + \sum_{i_l=1}^{M_{l,\mathbf{i}_{l-1}}}\Bigg(\varphi_l^*(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}}) \quad \underbrace{\Bigg)\cdots\Bigg)}_{l \text{ parentheses}} \quad .\end{aligned}$$

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$$\begin{aligned}\mathcal{F}'_{L,2}(\mathbf{Q}_L) \triangleq \frac{1}{2}\sum_{i_1=1}^{M_1}\Bigg(\varphi_1(C_{i_1},P_{i_1}) + \cdots + \sum_{i_l=1}^{M_{l,\mathbf{i}_{l-1}}}\Bigg(\varphi_l(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}},P_{\mathbf{i}_l}) + \cdots + \sum_{i_{L-2}=1}^{M_{L-2,\mathbf{i}_{L-3}}}\Bigg(\varphi_{L-2}(C_{\mathbf{i}_{L-2}},\widetilde{P}_{\mathbf{i}_{L-3}},P_{\mathbf{i}_{L-2}}) \\ + \sum_{i_{L-1}=1}^{M_{L-1,\mathbf{i}_{L-2}}}\Bigg(\varphi_{L-1}(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-2}},P_{\mathbf{i}_{L-1}}) + \tau^*_{L-1}(C_{\mathbf{i}_{L-1}},\widetilde{P}_{\mathbf{i}_{L-1}}) \quad \underbrace{\Bigg)\cdots\Bigg)}_{L-1 \text{ parentheses}} \quad .\end{aligned} \tag{26}$$

$$\begin{aligned}\mathcal{F}'_{l+1,2}(\mathbf{Q}_L) \triangleq &\frac{1}{2}\sum_{i_1=1}^{M_1}\Bigg(\varphi_1(C_{i_1},P_{i_1}) + \cdots + \sum_{i_{l-1}=1}^{M_{l-1,\mathbf{i}_{l-2}}}\Bigg(\varphi_{l-1}(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-2}},P_{\mathbf{i}_{l-1}}) \\ &+ \sum_{i_l=1}^{M_{l,\mathbf{i}_{l-1}}}\Bigg(\varphi_l(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}},P_{\mathbf{i}_l}) + \tau^*_l(C_{\mathbf{i}_l},P_{\mathbf{i}_l}\cdot\widetilde{P}_{\mathbf{i}_{l-1}}) \quad \underbrace{\Bigg)\cdots\Bigg)}_{l \text{ parentheses}} \quad .\end{aligned} \tag{27}$$

$$\begin{aligned}\mathcal{F}'_{l,2}(\mathbf{Q}_L) \triangleq &\frac{1}{2}\sum_{i_1=1}^{M_1}\Bigg(\varphi_1(C_{i_1},P_{i_1}) + \cdots + \sum_{i_{l-2}=1}^{M_{l-2,\mathbf{i}_{l-3}}}\Bigg(\varphi_{l-2}(C_{\mathbf{i}_{l-2}},\widetilde{P}_{\mathbf{i}_{l-3}},P_{\mathbf{i}_{l-2}}) \\ &+ \sum_{i_{l-1}=1}^{M_{l-1,\mathbf{i}_{l-2}}}\Bigg(\varphi_{l-1}(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-2}},P_{\mathbf{i}_{l-1}}) + \tau^*_{l-1}(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-1}}) \quad \underbrace{\Bigg)\cdots\Bigg)}_{l-1 \text{ parentheses}} \quad .\end{aligned} \tag{32}$$

Based on the fact that $\mathcal{F}'_{l+1,2}(\mathbf{Q}_L) \geq \mathcal{F}'_{l,1}(\mathbf{Q}_L)$, the problem (6) is further reduced to minimizing the cost $\mathcal{F}'_{l,1}(\mathbf{Q}_L)$. It follows that, if the values $\varphi^*_l(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}})$ are already computed for each possible pair $(C_{\mathbf{i}_l},\widetilde{P}_{\mathbf{i}_{l-1}})$, then the partition of $C_{\mathbf{i}_{l-1}}$ into cells $C_{\mathbf{i}_l}$ can be optimized separately for each pair $(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-1}})$. We denote by $\mathbf{r}^*_l(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-1}})$ this optimal refined partition, which will be computed for each possible cell $C_{\mathbf{i}_{l-1}}$, i.e., for each interval $[c,d)$, $c,d\in\bar{\mathcal{A}}$, $c<d$, and for each positive integer $P \leq P^{(l-1)}_{\max}$, representing a possible value $\widetilde{P}_{\mathbf{i}_{l-1}}$. In other words, we obtain

$$\mathbf{r}^*_l([c,d),P) = \arg\min_{M,\mathbf{r}}\sum_{i=1}^{M}\varphi^*_l([r_{i-1},r_i),P), \tag{30}$$

where $\mathbf{r}=(r_0,\cdots,r_M)\in\mathcal{S}_{M+1}(c,d)\cap\bar{\mathcal{A}}^{M+1}$. Subsequently, the following cost is obtained at optimality in (30),

$$\tau^*_{l-1}([c,d),P) = \sum_{i=1}^{M^*}\varphi^*_l([r^*_{i-1},r^*_i),P), \tag{31}$$

where $\mathbf{r}^*_l([c,d),P) = (r^*_0,\cdots,r^*_{M^*})$. By further replacing the partition of each $C_{\mathbf{i}_{l-1}}$ in $\mathcal{F}'_{l,1}(\mathbf{Q}_L)$ with the optimum partition $\mathbf{r}^*_l(C_{\mathbf{i}_{l-1}},\widetilde{P}_{\mathbf{i}_{l-1}})$, we obtain a cost function that depends only on the EC-SRUPQs levels 1 through $l-1$, namely $\mathcal{F}'_{l,2}(\mathbf{Q}_L)$ in (32) at the top of this page. Since $\mathcal{F}'_{l,1}(\mathbf{Q}_L) \geq \mathcal{F}'_{l,2}(\mathbf{Q}_L)$, it follows that the problem (6) is equivalent to minimizing $\mathcal{F}'_{l,2}(\mathbf{Q}_L)$.

The aforementioned procedure will be repeated in decreasing order for $l = L-1, L-2, \cdots, 2$. After doing so, the problem (6) reduces to minimizing the cost $\mathcal{F}'_{2,2}(\mathbf{Q}_L)$ given as follows,

$$\mathcal{F}'_{2,2}(\mathbf{Q}_L) = \frac{1}{2}\sum_{i_1=1}^{M_1}\Big(\varphi_1(C_{i_1},P_{i_1}) + \tau^*_1(C_{i_1},P_{i_1})\Big).$$

As in the previous section, if the values $\tau^*_1(C_{i_1},P_{i_1})$ are known for all possible pairs (C_{i_1},P_{i_1}), then the optimal P_{i_1} can be evaluated independently for each C_{i_1}. Thus, the optimal P_{i_1} can be computed for each possible choice of C_{i_1}, i.e., for each interval $[c,d)$, $c,d\in\bar{\mathcal{A}}$, $c<d$. Let us denote by $P^*_1([c,d))$ this optimal P_{i_1}, i.e.,

$$P^*_1([c,d)) = \arg\min_{\substack{P'\in\mathbb{Z}_+ \\ P'\leq P^{(1)}_{\max}}}\Big(\varphi_1([c,d),P') + \tau^*_1([c,d),P')\Big), \tag{33}$$

where the smallest value is taken in case of multiple minimizers. Additionally, we denote

$$\varphi^*_1([c,d)) = \varphi_1([c,d),P^*_1([c,d))) + \tau^*_1([c,d),P^*_1([c,d))). \tag{34}$$

By replacing the above in the cost function $\mathcal{F}'_{2,2}(\mathbf{Q}_L)$, the problem (6) reduces to

$$\begin{aligned}&\min_{M_1,\mathbf{r}} \quad \frac{1}{2}\sum_{i_1=1}^{M_1}\varphi^*_1([r_{i_1-1},r_{i_1})) \\ &\text{subject to} \quad \mathbf{r}\in\mathcal{S}_{M_1+1}(0,\infty)\cap\bar{\mathcal{A}}^{M_1+1}.\end{aligned} \tag{35}$$

Based on the above discussion, we conclude that the following procedure can be utilized to solve the problem (6).

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Step 1) For each $l, l \in [1 : L-1]$, determine an integer $P_{\max}^{(l)}$ such that there is an optimal EC-SRUPQ satisfying condition (21).

Step 2) Repeat the following steps for all l from L down to 2:

Step 2.A) For each pair $(c, d) \in \bar{\mathcal{A}}^2$, with $c < d$, and each positive integer $P \le P_{\max}^{(l-1)}$, compute $P_l^*([c, d], P)$ according to (22) when $l = L$, respectively according to (28) for $l < L$.

Step 2.B) For each pair $(c, d) \in \bar{\mathcal{A}}^2$, with $c < d$, and each positive integer $P \le P_{\max}^{(l-1)}$, compute the optimal partition $\mathbf{r}_l^*([c, d], P)$ defined in (30) and $\tau_{l-1}^*([c, d], P)$ given in (31)[3].

Step 3) For each pair $(c, d) \in \bar{\mathcal{A}}^2$, with $c < d$, compute $P_1^*([c, d])$ defined in (33).

Step 4) Solve the problem (35).

The detailed solutions for each step are discussed in the following Section.

It is also important to notice that when $L = 3$, the above procedure reduces to the sequence of steps to solve the problem (7), described in the previous section. Likewise, if we replace L by 2, we recover the major steps of the design algorithm for the optimal EC-SRUPQ with two refinement levels, presented in [20, Section III. B].

V. Step-by-step Solution for the EC-SRUPQ Design with General L

A. Solution for Step 1)

In order to determine the values of $P_{\max}^{(l)}$, $l \in [1 : L-1]$, we need to consider problem (16) in [20], which, for completeness, is presented next. Let us denote $f(y) = -\text{sinc}^2(\frac{1}{y})$ and $h(y) = \ln y$, for any $y > 0$. Then for any $P \in \mathbb{Z}_+$ and $\delta > 0$, consider the following minimization problem

$$\min_{P' \in \mathbb{Z}_+} (f(PP') + \delta h(PP')). \tag{36}$$

According to [20, Proposition 1], the above minimum can be achieved with a finite value of P', which will be denoted by $P_{opt}(P, \delta)$. Note that, if there are more than one minimizers, the smallest one is considered.

For each $j, j \in [1 : L]$, denote $\delta_j = \frac{\lambda_j}{\phi_j x([a_K, a_{K+1}])^2 \ln 2}$. Next, for each $l, l \in [1 : L]$, denote

$$P_{0,l} = \max_{l \le j \le L} P_{opt}(1, \delta_j). \tag{37}$$

We will define the values $P_{\max}^{(l)}$ recursively. Namely,

$$P_{\max}^{(1)} = P_{0,1}, \tag{38}$$

and, for $l, l \in [2 : L-1]$,

$$P_{\max}^{(l)} = P_{0,l} + P_{\max}^{(l-1)}. \tag{39}$$

Clearly, one has $P_{\max}^{(l)} \le l P_{\max}^{(1)}$.

The following result validates that the above definition satisfies the requirements imposed on $P_{\max}^{(l)}$. The proof of the result is deferred to the appendix.

Proposition 1: There is an optimal EC-SRUPQ (i.e., a solution to problem (6)) that satisfies condition (21).

According to [17], [20], each value $P_{opt}(1, \delta_j)$ can be determined using a linear search in $O(P_{opt}(1, \delta_j))$ time, $j \in [1 : L]$. To compute all of them takes $O(LP_{\max}^{(1)})$ time, since $P_{\max}^{(1)} \ge P_{opt}(1, \delta_j)$, $j \in [1 : L]$. By accounting for the remaining maximizations in (37), the running time to complete Step 1) amounts to $O(LP_{\max}^{(1)} + L) = O(LP_{\max}^{(1)})$.

B. Solution to Step 2.A)

As we have already pointed out, the problem (22) is identical to problem (10) in [20] up to a change of parameters. More specifically, it is equivalent to solving problem (36) for $\delta = \frac{\lambda_L}{\phi_L x([c,d])^2 \ln 2}$. Therefore, Step 2.A) for $l = L$ can be solved using the same procedure as the solution to Step 1 in [20, Section III. C]. Specifically, for each integer P, all values $P_L^*([c, d], P)$ can be determined using Algorithm 1 in [17]. This requires $O(KP_{opt}(P, \delta_L) + K^2)$ time for fixed P. Performing this for all P, $1 \le P \le P_{\max}^{(L-1)}$, amounts to $O(K \sum_{P=1}^{P_{\max}^{(L-1)}} P_{opt}(P, \delta_L) + K^2 P_{\max}^{(L-1)})$ operations. According to Proposition 4 of [20], the following holds

$$P_{opt}(P, \delta_L) \le \frac{P_{opt}(1, \delta_L)}{P} + 1, \tag{40}$$

which implies that

$$\sum_{P=1}^{P_{\max}^{(L-1)}} P_{opt}(P, \delta_L) \le P_{opt}(1, \delta_L) \sum_{P=1}^{P_{\max}^{(L-1)}} \frac{1}{P} + P_{\max}^{(L-1)} \le P_{0,L}(\ln P_{\max}^{(L-1)} + 1) + P_{\max}^{(L-1)},$$

where we used the well-known upper bound on the partial sum of the Harmonic series. We conclude that the time complexity of the algorithm to solve Step 2.A) when $l = L$ is $O(KP_{0,L} \log P_{\max}^{(L-1)} + K^2 P_{\max}^{(L-1)})$.

Let us analyze now Step 2.A) for $l < L$. Solving the problem (28) for fixed pair (c, d) and fixed P is straightforward using linear search and requires $O(P_{\max}^{(l)}/P)$ time. Doing so for all pairs (c, d) and all values of P, takes $O(K^2 \sum_{P=1}^{P_{\max}^{(l)}} P_{\max}^{(l)}/P) = O(K^2 P_{\max}^{(l)} \log P_{\max}^{(l)})$ operations. Since $P_{\max}^{(l)} \le P_{\max}^{(L-1)} \le (L-1)P_{\max}^{(1)}$, we conclude that solving Step 2.A) for all l, $l \in [2 : L-1]$, requires $O(L^2 K^2 P_{\max}^{(1)}(\log P_{\max}^{(1)} + \log L))$ operations.

C. Solution to Step 2.B)

Let us discuss now the solution for Step 2.B). It is obvious that for each positive integer P, the problem (30) is identical to problem (12) in [20], up to a change in notations. Therefore, the solution for Step 2 in [20, Section III. D] can be used for Step 2.B). According to the justification in [20, Section III. D], solving the problem (30) for fixed P and all pairs (c, d) is equivalent to solving multiple minimum weight path (MWP) problems in the weighted directed acyclic graph (WDAG) $G_{P,l}$ specified next.

For each $l \in [2 : L]$ and each positive integer P, construct the WDAG $G_{P,l} = (V, E, w_{P,l})$ with vertex set $V = \{0, 1, \cdots, K+1\}$ and edge set $E = \{(m, n) \in V^2 | 0 \le$

[3] Note that by letting $l = L$ in (30) and (31), relations (24) and (25) are recovered, respectively.

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$m < n \leq K+1\}$. For each edge $(m,n) \in E$, its weight $w_{P,l}(m,n)$ is

$$w_{P,l}(m,n) \triangleq \varphi_l^*([a_m,a_n),P),$$

where $\varphi_l^*([a_m,a_n),P)$ is given by (29). Then $\mathbf{r}_l^*([a_m,a_n),P)$ is an MWP from node m to node n in $G_{P,l}$. The algorithm to find all these paths will loop through $m \in V$, and for each m, it will solve the single source MWP problem with m being the source, i.e., it will find the MWP from m to any other node in the graph $G_{P,l}$ reachable from m. The time complexity of the solution for the single source MWP problem is $O(|V|+|E|) = O(K^2)$. Since this is done for each $m \in V$ and each $P \leq P_{\max}^{(l-1)}$, the total time complexity to solve Step 2.B) amounts to $O(K^3 P_{\max}^{(l-1)})$. Accounting for all values of l from L down to 2, we obtain the time complexity of $O(LK^3 P_{\max}^{(L-1)}) = O(L^2 K^3 P_{\max}^{(1)})$.

D. Solution to Steps 3) and 4)

Step 3) is straightforward. Namely, for each pair (c,d), a linear search is performed. Thus, the total running time amounts to $O(K^2 P_{\max}^{(1)})$.

Let us discuss now Step 4). Note that the problem (35) is identical to problem (15) in [20], up to a change in notations. Thus, according to the justification in [20], solving Step 4) is equivalent to finding an MWP in the WDAG G defined next. Namely, $G = (V,E,w)$, where V and E are as defined in subsection V-C. The weight of each edge $(m,n) \in E$ is $w(m,n)$ defined as

$$w(m,n) \triangleq \varphi_1^*([a_m,a_n)).$$

If all edge weights are available, which is the case since they were computed at the previous steps, then Step 4) can be solved in $O(|V|+|E|) = O(K^2)$ operations.

E. Time Complexity

Clearly, Step 2) is the most computationally intensive. In conclusion, the time complexity of the proposed solution algorithm to the problem (6) is $O(LK^2 P_{\max}^{(L-1)}(\log P_{\max}^{(L-1)} + K))$ time. If $\log P_{\max}^{(L-1)} \leq K$, which is expected to hold in practical scenarios, then the overall time complexity becomes $O(LK^3 P_{\max}^{(L-1)}) = O(L^2 K^3 P_{\max}^{(1)})$, which is essentially only a factor of $O(L^2)$ higher than the time complexity for $L = 2$.

VI. Experimental Results

This section assesses the practical performance of the proposed EC-SRUPQ design algorithm for $L = 3$ and compares it with the theoretical rate-distortion bounds, with the entropy-coded FR-SRUPQ of [7] and with the single-level optimal ECUPQ of [17]. The experiments are conducted for a two-dimensional random vector (X_1, X_2), where X_1 and X_2 are i.i.d. Gaussian variables with zero-mean and unit-variance, with the following joint pdf in polar coordinates

$$p(r,\theta) = \frac{r}{2\pi}\exp\left(-\frac{r^2}{2}\right),\ 0 \leq r < \infty,\ 0 \leq \theta < 2\pi,$$

where $r = \sqrt{x_1^2 + x_2^2}$, and $\theta = \tan^{-1}(x_2/x_1)$. It then follows that $g(r) = r\exp(-r^2/2)$.

The set of possible thresholds $\mathcal{A}$ is obtained by dividing the range $[0,6]$ into subintervals of size 0.05. In other words, $K = 120$ and $a_i = 0.05i$, for $0 \leq i \leq K$. In this section, the notations D_l and R_l are utilized in place of $D(Q_l)$, respectively $R(Q_l)$, for $l = 1,2,3$. Additionally, $R(D_l)$ denotes the rate-distortion function for the Gaussian source, i.e., $R(D_l) = -0.5\log_2(D_l)$.

We have implemented and run the proposed algorithm using the weight vectors (ϕ_1,ϕ_2,ϕ_3)= $(0.1,0.1,0.8)$, $(0.1,0.8,0.1)$, $(0.8,0.1,0.1)$, $(0.1,0.45,0.45)$, $(0.45,0.1,0.45)$, $(0.45,0.45,0.1)$, $(0.2,0.6,0.2)$, $(0.33,0.33,0.34)$, $(0.9,0.05,0.05)$, $(0.05,0.9,0.05)$ and $(0.05,0.05,0.9)$.

Let us first visualize the output of the proposed algorithm and the structure of the corresponding three-level EC-SRUPQ for an example. Figure 2 shows the MWPs in the WDAGs $G_{P,3}$ (with red solid arcs), $G_{P,2}$ (blue dotted arcs) and G (black dashed arcs), respectively, when $(\phi_1,\phi_2,\phi_3) = (0.33,0.33,0.34)$ and $(\lambda_1,\lambda_2,\lambda_3) = (0.2,0.1,0.059)$. The optimal values $P_3^*([a_m,a_n),P)$, $P_2^*([a_m,a_n),P)$ and $P_1^*([a_m,a_n))$ are also shown along the corresponding edges. Figure 3 plots the partitions of the component UPQs of the corresponding EC-SRUPQ. The triples of rates and distortions (in dB) are $(R_1,R_2,R_3) = (0.461, 1.394, 1.979)$ and $(D_1,D_2,D_3) = (-1.950, -6.767, -10.093)$, respectively. The sequences of thresholds of the magnitude quantizers are $(0, 2.0, \infty)$, $(0, 2.0, 4.4, \infty)$ and $(0, 1.25, 2.0, 3.4, 4.4, 6.0, \infty)$ for Q_1, Q_2 and Q_3, respectively. The numbers of phase regions associated to the magnitude bins are 1 and 6 for Q_1, 4, 12 and 16 for Q_2 and $4, 8, 12, 24, 18$ and 36 for Q_3.

Next we compare the performance of the proposed EC-SRUPQ design with the theoretical rate-distortion bounds for the Gaussian source. It is known that the Gaussian source with squared-error distortion is successively refinable, which implies that any triple of distortions (D_1,D_2,D_3) is achievable by a successively refinable code with rate triple $(R(D_1),R(D_2),R(D_3))$ as the block length approaches infinity [24]. Note that since the proposed scheme uses scalar quantization, the existence of a rate gap to the rate-distortion function $(R(D_1),R(D_2),R(D_3))$ is expected. In particular, the rate gap between the optimal single-level ECUPQ and the rate-distortion limit was proved in [6] to be $\frac{1}{2}\log_2\frac{2\pi e}{12} = 0.2546$ bits/sample as the rate approaches infinity. On the other hand, an asymptotical analysis of performance is not available for EC-SRUPQ. However, the results of [17, Table II] suggest that the optimal ECUPQ is not successively refinable at finite rates. In other words, the optimal ECUPQ for some rate R_2' is not necessarily a refinement of the optimal ECUPQ at another rate R_1', where $R_1' < R_2'$. This implies that the component ECUPQs of an EC-SRUPQ cannot achieve the optimal single-level performance simultaneously. Therefore, it is expected for the rate gap $R_l - R(D_l)$, $l = 1,2,3$, at finite rates in the EC-SRUPQ framework to be larger than the value of 0.2546 bits/sample.

Figures 4(a)–4(c) illustrate the performance of the proposed EC-SRUPQ with $L = 3$, in terms of the rate-gap pair $(R_2 - R(D_2), R_1 - R(D_1))$, $(R_3 - R(D_3), R_2 - R(D_2))$ and

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

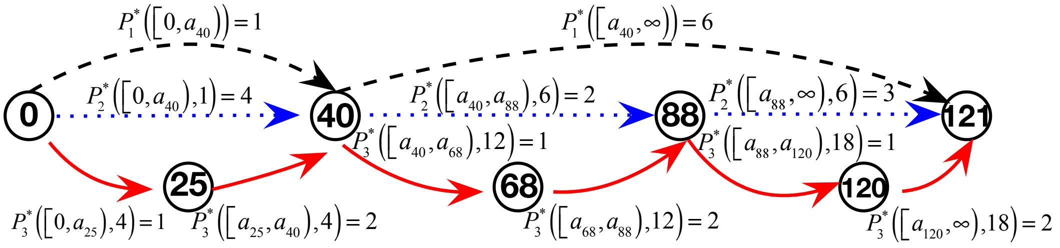
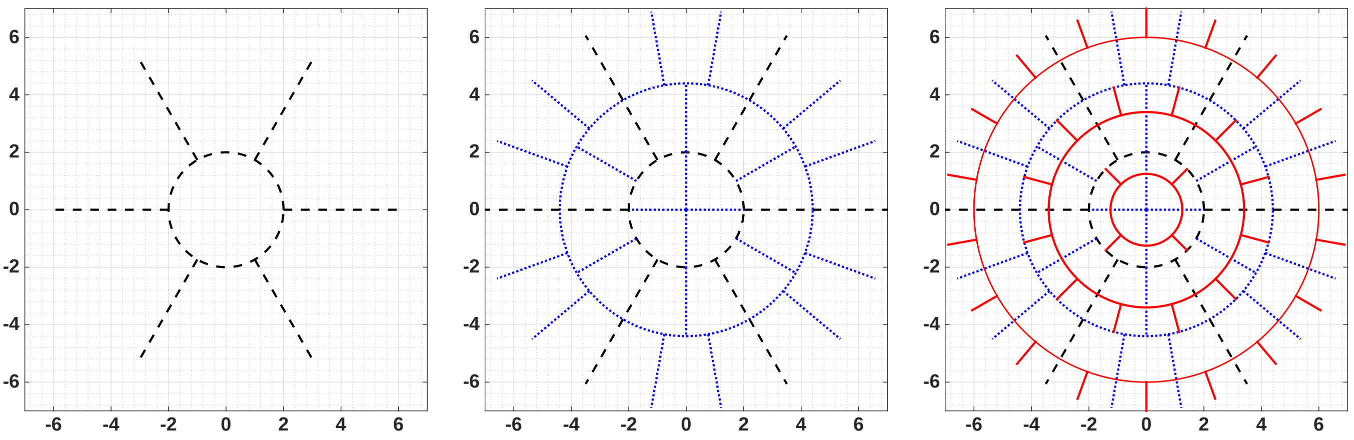

$P_1^*([0,a_{40}))=1$
$P_1^*([a_{40},\infty))=6$
$P_2^*([0,a_{40}),1)=4$
$P_2^*([a_{40},a_{88}),6)=2$
$P_2^*([a_{88},\infty),6)=3$
0
40
88
121
$P_3^*([a_{40},a_{68}),12)=1$
$P_3^*([a_{88},a_{120}),18)=1$
25
68
120
$P_3^*([0,a_{25}),4)=1$
$P_3^*([a_{25},a_{40}),4)=2$
$P_3^*([a_{68},a_{88}),12)=2$
$P_3^*([a_{120},\infty),18)=2$

Fig. 2. Visualization of the MWPs in the WDAGs $G_{P,3}$ (with red solid edges), $G_{P,2}$ (blue dotted edges) and G (black dashed edges), respectively, when $(\phi_1, \phi_2, \phi_3) = (0.33, 0.33, 0.34)$ and $(\lambda_1, \lambda_2, \lambda_3) = (0.2, 0.1, 0.059)$; discussed in Section VI.



(a) The partitions of Q_1

(b) The partition of Q_2

(c) The partition of Q_3

Fig. 3. The partitions of the EC-SRUPQ $\mathbf{Q}_3$ with the parameters in Figure 2. The dashed black lines represent the boundaries of the quantization regions of Q_1, while the dotted blue lines and the solid red lines represent the boundaries corresponding to the refinement at the second and third level, respectively, as described in Section VI.

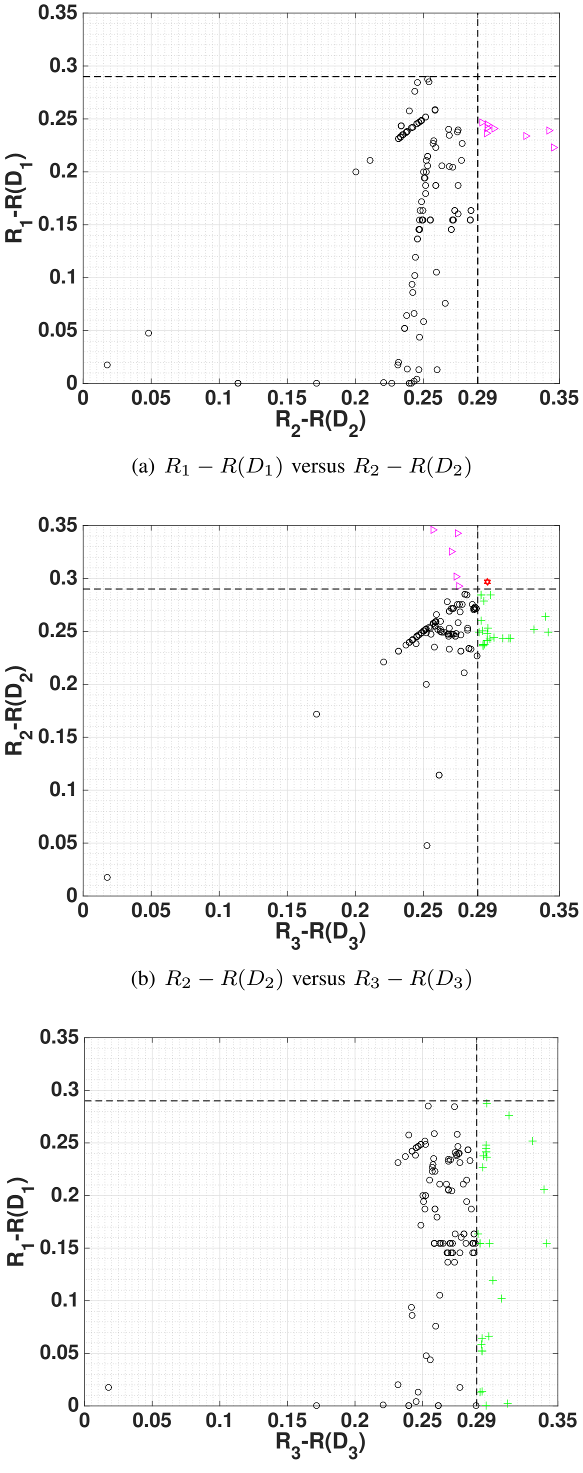
$(R_3 - R(D_3), R_1 - R(D_1))$, respectively. Additionally, the performance of the achieved distortion pair (D_2, D_1), (D_3, D_2) and (D_3, D_1) are plotted in Figures 5(a)–5(c), respectively.

It can be noticed from Figures 4(a)–4(c) that in most cases the gaps $R_l - R(D_l)$, $l = 1, 2, 3$, are within 0.29 bits/sample, which is very close to the value of 0.2546 bits/sample. Moreover, note that there are also cases in which there is some additional loss, but only either in the second or in the third refinement level and very rarely in both of them simultaneously. These cases are represented in Figures 4(a)–4(c) with triangles (extra loss only in the second level), pluses (extra loss only in the third level) and hexagrams (extra loss in both second and third levels), respectively. It should be noted that the cases with extra loss occur mostly when the corresponding distortion is small, i.e., mostly less than 0.1 as shown in Figures 5(a)–5(c). As explained in [20], the existence of this additional loss could be attributed to the additional tension induced in the optimization by competing requirements simultaneously at the three decoders, instead of only one decoder.

Let us now discuss the impact of the weights ϕ_l, $l \in [1:L]$. As mentioned earlier, it is generally not possible for all components $(Q_1, \cdots, Q_L)$ of an EC-SRUPQ to simultaneously achieve their corresponding optimal single-level performance. This motivates the inclusion in the objective function of a weighted sum of the distortions at all levels. The weight ϕ_l assigned to level l can be interpreted as the relative importance of that level. The higher ϕ_l, the higher the emphasis on the minimization of distortion $D(Q_l)$ is. In particular, if for some $l_0 \in [1:L]$, ϕ_{l_0} is very high in comparison with the other values ϕ_l, then the UPQ at level l_0 should be very close to the optimal single-level ECUPQ. To verify the above claim we have implemented the proposed algorithm for $L = 3$ and $(\phi_1, \phi_2, \phi_3) = (0.9, 0.05, 0.05)$, $(0.05, 0.9, 0.05)$, $(0.05, 0.05, 0.9)$. In order to better assess the impact of the weights ϕ_l, we have also considered $(\phi_1, \phi_2, \phi_3) = (0.33, 0.33, 0.34)$. For each of the above triples (ϕ_1, ϕ_2, ϕ_3), several different triples $(\lambda_1, \lambda_2, \lambda_3)$ were used in order to obtain different rates. The results are presented in Table I. The table also illustrates the comparison between the performance achieved at each refinement level with the optimal ECUPQ performance for the corresponding rate. For this we chose to compare against the ECUPQ designed in [17], since, according to the experimental results reported in [17, Tables II and IV], it outperforms all existing practical UPQ schemes for rates up to about 5.9 bits/sample. More specifically, the ECUPQ of [17] is superior to the practical uniform ECUPQ proposed in [6] based on the asymptotic analysis and to the entropy-coded FRUPQ of [3].

In Table I, the triple (R_1, R_2, R_3) is utilized to denote the rates for our scheme, while the corresponding distortion triple in dB is denoted by (D_1, D_2, D_3). Further, for $l = 1, 2, 3$, the

0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

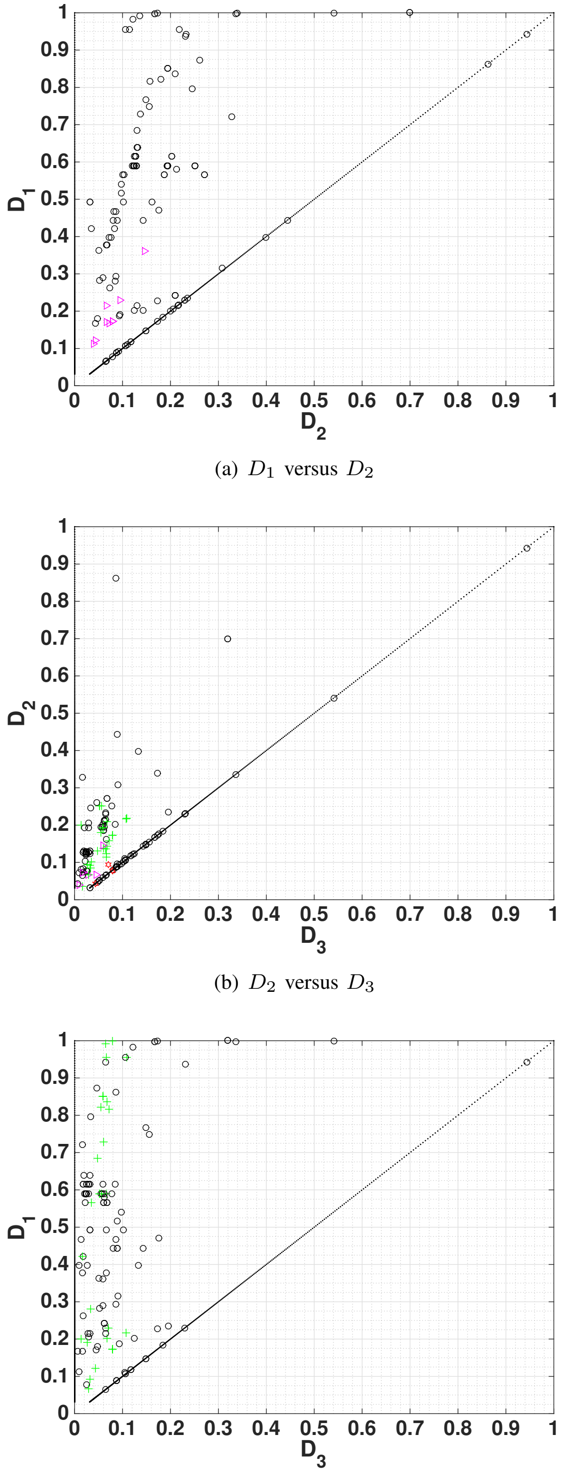


(a) $R_1 - R(D_1)$ versus $R_2 - R(D_2)$

(b) $R_2 - R(D_2)$ versus $R_3 - R(D_3)$

(c) $R_1 - R(D_1)$ versus $R_3 - R(D_3)$

Fig. 4. Gap in rate versus the theoretical lower bounds for the proposed EC-SRUPQ with $L = 3$; discussed in Section VI.



(a) D_1 versus D_2

(b) D_2 versus D_3

(c) D_1 versus D_3

Fig. 5. Distortion performance of the proposed EC-SRUPQ with $L = 3$; discussed in Section VI.

notation $\Delta_l = D_l - D_l^{[17]}$ represents the gap in dB between the performance of our scheme at level l and the optimal ECUPQ designed using the algorithm of [17], for a rate $R_l^{[17]}$ such that $|R_l - R_l^{[17]}| \leq 0.0011$. We observe that when the weight ϕ_l for one level is very high in comparison with the other levels, the performance at that level is very close to the optimum, i.e., Δ_l is very low. Additionally, for fixed triple (ϕ_1, ϕ_2, ϕ_3) and $l = 2, 3$, Δ_l generally increases as $R_l - R_{l-1}$ decreases. Similarly to [20], the above observation may be explained by the fact that the requirement that the partitions of Q_{l-1} and Q_l are embedded becomes more restrictive when $R_l - R_{l-1}$ is small, making it more difficult to design Q_{l-1} and Q_l close

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

TABLE I
PERFORMANCE COMPARISON OF THE PROPOSED EC-SRUPQ WITH THE ECUPQ OF [17].

(ϕ_1, ϕ_2, ϕ_3)	row label	(R_1, R_2, R_3)	(D_1, D_2, D_3)	$R_2 - R_1$	$R_3 - R_2$	$(\Delta_1, \Delta_2, \Delta_3)$
$(0.33, 0.33, 0.34)$	ℓ_1	$(1.297, 1.995, 2.214)$	$(-6.388, -10.226, -11.538)$	0.698	0.219	$(0.029, 0.292, 0.279)$
	ℓ_2	$(1.827, 2.621, 3.614)$	$(-9.514, -14.019, -20.095)$	0.794	0.993	$(0.008, 0.234, 0.114)$
	ℓ_3	$(0.553, 1.393, 2.290)$	$(-2.366, -6.726, -12.109)$	0.840	0.897	$(0.015, 0.245, 0.159)$
	ℓ_4	$(0.497, 1.423, 2.066)$	$(-2.115, -6.939, -10.705)$	0.926	0.643	$(1.127 \times 10^{-10},$ $0.207, 0.234)$
	ℓ_5	$(0.656, 1.936, 2.009)$	$(-2.869, -10.143, -10.524)$	1.280	0.073	$(0.004, 0.025, 0.072)$
$(0.90, 0.05, 0.05)$	ℓ_6	$(0.574, 1.228, 2.222)$	$(-2.474, -5.674, -11.689)$	0.654	0.994	$(0.004, 0.335, 0.177)$
	ℓ_7	$(0.535, 1.459, 3.073)$	$(-2.290, -7.146, -16.862)$	0.924	1.614	$(0.004, 0.207, 0.103)$
	ℓ_8	$(0.742, 2.054, 3.398)$	$(-3.299, -10.856, -18.759)$	1.312	1.344	$(2.227 \times 10^{-5},$ $0.018, 0.159)$
$(0.05, 0.90, 0.05)$	ℓ_9	$(1.393, 1.713, 2.749)$	$(-6.673, -8.834, -14.901)$	0.32	1.036	$(0.298, 0.011, 0.114)$
	ℓ_{10}	$(0.742, 2.045, 3.444)$	$(-3.299, -10.803, -19.047)$	1.303	1.399	$(9.448 \times 10^{-5},$ $0.008, 0.098)$
	ℓ_{11}	$(0.393, 1.717, 2.488)$	$(-1.648, -8.867, -13.159)$	1.324	0.771	$(3.061 \times 10^{-7},$ $0.002, 0.299)$
$(0.05, 0.05, 0.90)$	ℓ_{12}	$(0.959, 1.738, 2.296)$	$(-4.427, -8.382, -12.274)$	0.779	0.558	$(0.027, 0.617, 0.033)$
	ℓ_{13}	$(0.240, 1.280, 2.707)$	$(-0.990, -6.108, -14.737)$	1.040	1.427	$(0.006, 0.203, 0.016)$
	ℓ_{14}	$(0.535, 1.733, 2.954)$	$(-2.290, -8.895, -16.224)$	1.198	1.221	$(0.004, 0.067, 0.025)$

to the corresponding optimal ECUPQs. On the other hand, if $R_l - R_{l-1}$ is too small for the refinement made by Q_l to be remarkable, we expect Δ_l to be close to Δ_{l-1}. This is how the small value of $\Delta_3 = 0.072$ dB seen in row ℓ_5 could be explained, even if $R_3 - R_2$ is also small (0.073), since the structures of Q_2 and Q_3 are very similar and Δ_2 is also very low (0.025 dB).

It can also be observed from Table I that for fixed (ϕ_1, ϕ_2, ϕ_3), Δ_1 tends to decrease as $R_2 - R_1$ increases, but at a faster rate than the decrease of Δ_l in response to the increase of $R_l - R_{l-1}$ when $l = 2, 3$. In particular, $\Delta_1 < 0.03$ when $R_2 - R_1 \geq 0.698$, while for Δ_2 or Δ_3 to become that small, the rate difference $R_2 - R_1$, respectively $R_3 - R_2$, has to be much higher. Finally, another important observation is that Δ_l can achieve very small values even if ϕ_l is not larger than the weights corresponding to the other levels, provided that $R_l - R_{l-1}$ is large enough for $l = 2, 3$, respectively, $R_2 - R_1$ is large enough for $l = 1$. Thus, values of Δ_l smaller than 0.1 dB can be obtained even when ϕ_l is moderate or low as can be seen in rows $\ell_5, \ell_8, \ell_{14}$ for $l = 2$, in rows ℓ_5, ℓ_{10} for $l = 3$, and in all rows except for ℓ_9 for $l = 1$.

Even though Table I demonstrates that the performance of the proposed EC-SRUPQ at each level can be made very close to its single-level counterpart, some small performance degradation still remains. This raises the question of what is the advantage of EC-SRUPQ over the switched ECUPQ, which uses the optimal single-level ECUPQ for each desired rate, in situations where the rate adaptation is needed. The answer is that with EC-SRUPQ, if during the transmission of a bitstream its rate has to be changed at an intermediate network node, this can be done easily simply by dropping a suffix or appending a suffix to the current bitstream. On the other hand, with the switched ECUPQ, a new bitstream has to be generated from scratch, operation which requires more computing resources and hence is more energy consuming than in the SRUPQ case. Using techniques that enable energy savings is of utter importance nowadays and is in accordance with the efforts of reducing the carbon footprint of modern communication systems.

Next, the performance of the proposed EC-SRUPQ is compared with the entropy-coded FR-SRUPQ of [7]. Recall that the UPQ at the l-th level in the latter scheme contains exactly 2^l quantization bins and, since it is an FRUPQ, it has the rate $R^{(l)} = l/2$ bits/sample. The design of [7] is greedy, i.e., $Q^{(1)}$ is the optimal FRUPQ with two cells, and for each $l \in [2 : L]$, $Q^{(l)}$ is the (asymptotically) best one-bit refinement of $Q^{(l-1)}$. This is achieved by refining either the magnitude (i.e., dividing the magnitude cell into two) or the phase (i.e., doubling the number of phase regions) for each magnitude region of $Q^{(l-1)}$. Since our EC-SRUPQ uses entropy coding, we will also apply entropy coding to the FR-SRUPQ of [7] for the purpose of the comparison. Thus, we will compute the rate of $Q^{(l)}$ as $H^{(l)}$, which is half of the entropy of the output of $Q^{(l)}$. Note that the design procedure implies that for each $l \geq 1$, the pair $(H^{(l)}, D^{(l)})$ is fixed. The only way of achieving other rates and distortions at the L levels is by constructing the FR-SRUPQ with a larger number of levels $L' > L$ and then selecting L of its components. However, this method is still very restrictive since the rate-distortion pairs achievable at a refinement level are confined to the set $\mathcal{RD} = \{(H^{(l)}, D^{(l)}) | l \geq 1\}$. Our scheme does not have such a limitation and, as it can be inferred from Figures 4(a)–4(c) and 5(a)–5(c), it can achieve a dense set of rate-distortion pairs at each refinement level.

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED EC-SRUPQ WITH THE ENTROPY-CODED FR-SRUPQ OF [7].

(R_1, R_2, R_3)	(N_1, N_2, N_3)	(D_1, D_2, D_3)	$(N_1, N_2, N_3)^{[7]}$	$(H_1, H_2, H_3)^{[7]}$	$(D_1, D_2, D_3)^{[7]}$	$(\Delta'_1, \Delta'_2, \Delta'_3)$
$(0.461, 1.394, 1.979)$	$(7, 34, 102)$	$(-1.950, -6.767, -10.093)$	$(2, 8, 16)$	$(0.5, 1.482, 1.979)$	$(-1.664, -6.043, -8.882)$	$(0.286, 0.725, 1.211)$
$(0.497, 1.838, 2.926)$	$(7, 98, 412)$	$(-2.115 - 9.577, -15.904)$	$(2, 16, 64)$	$(0.5, 1.979, 2.947)$	$(-1.664, -8.882, -14.783)$	$(0.451, 0.695, 1.121)$
$(0.965, 1.969, 2.810)$	$(20, 100, 431)$	$(-4.493, -10.168, -15.207)$	$(4, 16, 64)$	$(1.0, 1.979, 2.948)$	$(-4.396, -8.882, -14.783)$	$(0.097, 1.286, 0.424)$
$(1.924, 2.398, 3.409)$	$(106, 285, 797)$	$(-10.039, -12.597, -18.850)$	$(16, 32, 128)$	$(1.979, 2.476, 3.435)$	$(-8.882, -11.430, -17.241)$	$(1.157, 1.167, 1.609)$

Table II illustrates the performance comparison of the proposed three-level EC-SRUPQ with the entropy-coded FR-SRUPQ of [7]. We implemented the scheme of [7] for $L = 7$ (a brief description of our implementation of the algorithm of [7] can be found in [20, Section V]) and selected several triples of component UPQs. For each case, the triple of rates is denoted by $(H_1, H_2, H_3)^{[7]}$ and the triple of distortions by $(D_1, D_2, D_3)^{[7]}$. In each case, the proposed EC-SRUPQ used for comparison was obtained by running our algorithm for the weights $(\phi_1, \phi_2, \phi_3) = (0.33, 0.33, 0.34)$ (so that no particular level is favored by the objective function) and various triples $(\lambda_1, \lambda_2, \lambda_3)$ such that R_l is smaller than or equal to $H_l^{[7]}$, for each $l \in [1 : 3]$. Table II also shows the triples of numbers of quantization regions for the two schemes and the difference in distortion $\Delta'_l = D_l^{[7]} - D_l$ in dB at each level $l, l \in [1 : L]$.

It can be noted from Table II that the performance improvement over the entropy-coded FR-SRUPQ of [7] is rather significant at all refinement levels. For instance, our scheme achieves an improvement of $\Delta'_1 = 0.451$ dB at the first refinement level when the rate is 0.5 bits/sample, even if the corresponding UPQ of the scheme of [7] is the optimal two-cell FRUPQ. Notably, the performance improvement over [7] has the tendency to increase with the increase of the refinement level, reaching a peak of 1.609 dB in Table II. The superiority of the proposed scheme comes from the following aspects. First, the entropy-constrained optimization used in the proposed design does not impose a fixed number of quantization regions, thus generating more quantization cells than the scheme of [7] for the same (or even smaller) rates. As Table II shows, the difference in the number of quantization cells between the two schemes is quite substantial. To illustrate graphically this dramatic difference we have depicted in Figure 6 the quantizer partitions of the FR-SRUPQ considered in the first row of Table II, i.e. formed of $(Q^{(1)}, Q^{(3)}, Q^{(4)})$. The partitions of the EC-SRUPQ used for comparison are depicted in Figure 3. Second, due to the greedy design method used in [7], the construction of the FRUPQs at higher refinement levels is severely constrained by the fixed partitions at the lower levels, fact which limits the performance considerably. On the other hand, no such constraints are imposed in our design, leading to more freedom in the choice of the partitions.

In summary, the proposed EC-SRUPQ outperforms substantially the FR-SRUPQ of [7] at each refinement level, even when entropy coding is applied to the latter scheme. It is true that the proposed scheme has higher design complexity than the FR-SRUPQ of [7], but this is not an insurmountable problem since the design can be performed offline for various values of the parameters and the results stored in tables. It pays off to do this since considerable performance improvement can be achieved. A possible design strategy to be used in practice is as follows. For a given source and a desired total rate, an EC-SRUPQ with a large number L of refinement levels can be designed such that to produce a decent rate increment for each level, which is required only once. After doing so, the EC-SRUPQ can be tailored with any number of refinement levels that is smaller than L, for various applications, depending on the bitrate and reconstruction quality requirements. There may be some degradation at individual refinement levels in comparison with the optimal ECUPQ for the corresponding rate, but this small loss is offset by the advantages vested by the successive refinement property, which allows for a fast adaptation of the bitstream rate and a graceful degradation of performance when the channel conditions deteriorate.

VII. CONCLUSION

This paper presents an algorithm for the design of general entropy-constrained successively refinable unrestricted polar quantizer (EC-SRUPQ), with an arbitrary number L of refinement levels. The cost to be minimized is a weighted sum of distortions and rates. We consider the constrained problem where the thresholds of the magnitude quantizers are confined to some predefined finite set and present a globally optimal solution. The proposed algorithm consists of L stages and the solution to each stage is based on solving the minimum-weight path problem for multiple node pairs in multiple weighted directed acyclic graphs. Moreover, for each refinement level of the EC-SRUPQ, we develop a corresponding upper bound on the possible number of phase regions of the phase quantizers. Additionally, the time complexity of the proposed approach is only a factor of $O(L^2)$ higher than that for the case of $L = 2$. Finally, the experimental results performed on a bivariate circularly symmetric Gaussian source in the case of $L = 3$ refinement levels show the excellent performance in practice.

VIII. ACKNOWLEDGEMENT

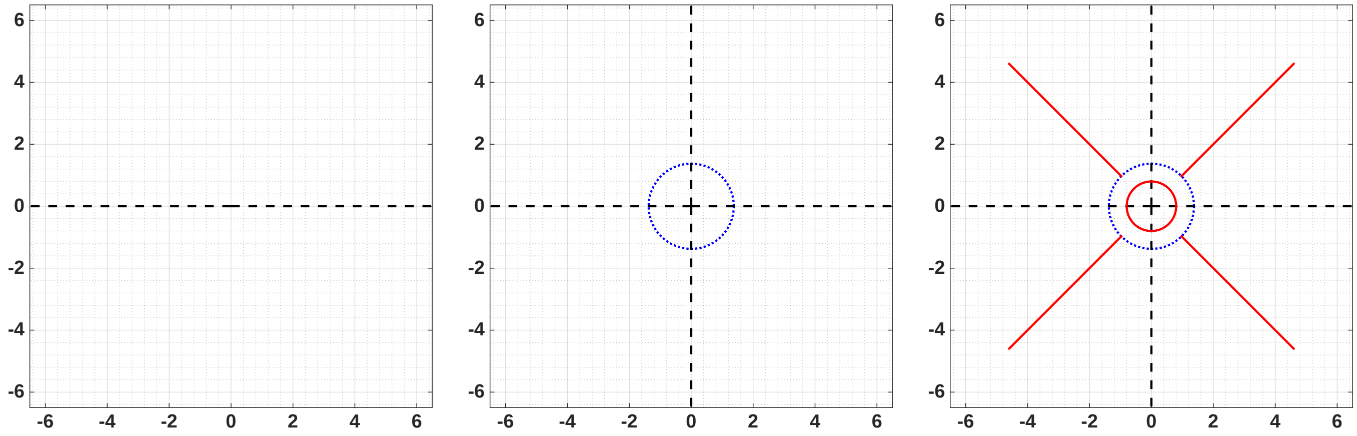
The authors would like to thank the Editor and the anonymous reviewers for their valuable comments and suggestions, which helped improve the quality and presentation of the work.

REFERENCES

[1] N. C. Gallagher, Jr., "Quantizing schemes for the discrete Fourier transform of a random time-series," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 2, pp. 156-163, Mar. 1978.

[2] W. A. Pearlman and R. M. Gray, "Source coding of the discrete Fourier transform", *IEEE Trans. Inform. Theory,* vol. IT-24, no. 6, pp. 683-692, Nov. 1978.

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) The partition for $Q^{(1)}$.

(b) The partition of $Q^{(3)}$.

(c) The partition of $Q^{(4)}$.

Fig. 6. The partitions of the three-level FR-SRUPQ of [7] formed of $(Q^{(1)}, Q^{(3)}, Q^{(4)})$ with $(N_1, N_2, N_3)^{[7]} = (2, 8, 16)$. The dashed black lines represent the boundaries of the quantization regions of $Q^{(1)}$, while the dotted blue lines and the solid red lines represent the boundaries corresponding to the refinement at the second and third level, respectively, as described in Section VI.

[3] S. G. Wilson, “Magnitude/phase quantization of independent Gaussian variates”, *IEEE Trans. Commun.,* vol. COM-28, no. 11, pp. 1924-1929, Nov. 1980.

[4] D. L. Neuhoff, “Polar quantization revisited,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT 1997),* pp. 60, Ulm, Germany, Jun. 1997.

[5] A. M. Bruckstein, R. J. Holt and A. N. Netravali, “Holographic representations of images”, *IEEE Trans. Image Process.,* vol. 7, no. 11, pp. 1583-1597, Nov. 1998.

[6] R. Vafin and W. B. Kleijn, “Entropy-constrained polar quantization and its application to audio coding”, *IEEE Trans. Speech and Audio Process.,* vol. 13, no. 2, pp. 220-232, Mar. 2005.

[7] E. Ravelli and L. Daudet, “Embedded polar quantization”, *IEEE Signal Process. Lett.,* vol. 14, no. 10, pp. 657-660, Oct. 2007.

[8] Z. Peric, M. R. Dincic and M. D. Petkovic, “The general design of asymptotic unrestricted polar quantizers with square cells”, *Digital Signal Process.,* vol. 23, no. 5, pp. 1731-1737, Sep. 2013.

[9] Z. Peric and J. Nikolic, “Design of asymptotically optimal unrestricted polar quantizer for Gaussian source”, *IEEE Signal Process. Lett.,* vol. 20, no. 10, pp. 980-983, Oct. 2013.

[10] P. Nazari, B-K. Chun, F. Tzeng and P. Heydari, “Polar quantizer for wireless receivers: theory, analysis, and CMOS implementation”, *IEEE Trans. Circuits and Systems-I: Regular Papers*, vol. 61, no. 3, pp. 877-887, Mar. 2014.

[11] J. Nikolic, Z. Peric and A. Jovanovic, “Variance mismatch analysis of unrestricted polar quantization for Gaussian source”, *IEEE Signal Process. Lett.,* vol. 21, no. 5, pp. 540-544, May. 2014.

[12] B. Chun, P. Nazari and P. Heydari, “An SQNR improvement technique based on magnitude segmentation for polar quantizers”, *IEEE Trans. Commun,* vol. 62, no. 11, pp. 3835-3841, Nov. 2014.

[13] M. Dincic and Z. Peric, “Multiproduct uniform polar quantizer”, *Radioengineering*, vol. 24, no. 1, pp. 233-239, Apr. 2015.

[14] A. Jovanovic, Z. Peric, J. Nikolic and M. Dincic, “Asymptotic analysis and design of restricted uniform polar quantizer for Gaussian sources”, *Digital Signal Process.,* vol. 49, pp. 24-32, Feb. 2016.

[15] N. Tawa and T. Kaneko, “A 950MHz RF 20MHz bandwidth direct RF sampling bit streamer receiver based on an FPGA”, in *Proc. 2017 IEEE MTT-S International Microwave Symposium (IMS)*, Honolulu, USA, pp. 594-597, Jun. 2017.

[16] Z. Peric, M. Petkovic, J. Nikolic and A. Jovanovic, “Support region estimation of the product polar companded quantizer for Gaussian source”, *Signal Process.*, vol. 143, pp. 140-145, Feb. 2018.

[17] H. Wu and S. Dumitrescu, “Design of optimal entropy-constrained unrestricted polar quantizer for bivariate circularly symmetric sources”, *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2169-2180, May. 2018.

[18] H. Wu and S. Dumitrescu, “Design of optimal fixed-rate unrestricted polar quantizer for bivariate circularly symmetric sources”, *IEEE Signal Process. Lett.,* vol. 25, no. 5, pp. 715-719, May. 2018.

[19] A. Jovanovic, Z. Peric and J. Nikolic, “An efficient iterative algorithm for designing an asymptotically optimal modified unrestricted uniform polar quantization of bivariate Gaussian random variables”, *Digital Signal Process.,* vol. 88, pp. 197-206, May. 2019.

[20] H. Wu and S. Dumitrescu, “Design of successively refinable unrestricted polar quantizer,” *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3525-3539, May. 2019.

[21] I. Bashir, R. B. Staszewski and P. T. Balsara, “Numerical model of an injection-locked wideband frequency modulator for polar transmitters”, *IEEE Trans. Microwave Theory and Techniques*, vol. 65, no. 5, pp. 1914-1920, May. 2017.

[22] Y-H. Chen, T-H. Wang, S-C. Lin, J-H. Chen and Y-J. E. Chen, “A 40-MHz bandwidth pulse-modulated polar transmitter for mobile applications”, in *Proc. 2019 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*, Orlando, USA, pp. 1-3, Jan. 2019.

[23] N. Markulic, P. T. Renukaswamy, E. Martens, B. V. Liempd and P. Wambacq, “A 5.5-GHz background-calibrated subsampling polar transmitter with -41.3-dB EVM at 1024 QAM in 28-nm CMOS”, *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1059-1073, Apr. 2019.

[24] W. Equitz and T. Cover, “Successive refinement of information,” *IEEE Trans. Inform. Theory,* vol. 37, no. 2, pp. 269–275, Mar. 1991.

[25] H. Jafarkhani and V. Tarokh, “Design of successively refinable trellis coded quantizers,” *IEEE Trans. Inform. Theory,* vol. 45, no. 5, pp. 1490-1497, Jul. 1999.

[26] X. Wu and S. Dumitrescu, ”On optimal multi-resolution scalar quantization”, *Proc. IEEE Data Compression Conf.*, pp. 322-331, Snowbird, UT, Apr. 2002.

[27] S. Dumitrescu and X. Wu, “Algorithms for optimal multi-resolution quantization,” *J. Algorithms,* vol. 50, no. 1, pp. 1-22, Jan. 2004.

[28] J. Chen, S. Dumitrescu, Y. Zhang and J. Wang, “Robust multiresolution coding,” *IEEE Trans. Commun,* vol. 58, no. 11, pp. 3186-3195, Nov. 2010.

[29] C-Y. Wang and M. Gastpar, “On distributed successive refinement with lossless recovery,” *IEEE Int. Symp. Inform. Theory (ISIT),* pp. 2669-2673, Honolulu, HI, Jun. 2014.

[30] B. N. Vellambi and R. Timo, “Common reconstructions in the successive refinement problem with receiver side information”, *IEEE Trans. Inform. Theory*, vol. 65, no. 10, pp. 6332-6354, Oct 2019.

[31] V. Kostina and E. Tuncel, “Successive refinement of abstract sources”, *IEEE Trans. Inform. Theory*, vol. 65, no. 10, pp. 6385-6398, Oct 2019.

[32] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 stillimage compression standard,” *IEEE Signal Process. Magazine,* vol. 18, no. 5, pp. 36-58, Sep. 2001.

[33] D. Taubman and M. Marcellin, *JPEG2000 image compression fundamentals, standards and practice,* Springer, 2012.

[34] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard”, *IEEE Trans. Circuits and Systems for Video Technology,* vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[35] P. A. Chou, T. Lookabaugh, and R. M. Gray, “Entropy-constrained vector quantization,” *IEEE Trans. Acoust., Speech, Signal Proc.,* vol. 37, no. 1, pp. 31-42, Jan. 1989.

This article has been accepted for publication in a future issue of *IEEE Transactions on Communications*, but has not been fully edited.
0090-6778 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

[36] D. Muresan and M. Effros, "Quantization as histogram segmentation: Optimal scalar quantizer design in network systems," *IEEE Trans. Info. Theory,* vol. 54, no. 1, pp. 344-366, Jan. 2008.

[37] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operat. Res.,* vol. 11, no. 3, pp. 399-417, Jun. 1963.

[38] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

APPENDIX

Proof of Proposition 1: Let us assume, for the sake of contradiction, that there is no optimal EC-SRUPQ that satisfies condition (21). Let us consider now an optimal EC-SRUPQ and let l_0 be the smallest value for which there is an $\mathbf{i}_{l_0}$ such that

$$\widetilde{P}_{\mathbf{i}_{l_0}} = \Pi_{j=1}^{l_0} P_{\mathbf{i}_j} > P_{\max}^{(l_0)}. \tag{41}$$

Then either $l_0 = 1$, in which case $P_{\mathbf{i}_{l_0}} \geq 2$ since $P_{\max}^{(1)} \geq 1$, or $l_0 > 1$ and

$$\widetilde{P}_{\mathbf{i}_l} = \Pi_{j=1}^{l} P_{\mathbf{i}_j} \leq P_{\max}^{(l)}, \text{ for all } 1 \leq l < l_0. \tag{42}$$

Note that, when $l_0 > 1$, relations (41) and (42) imply that

$$P_{\mathbf{i}_{l_0}} > P_{\max}^{(l_0)} / \widetilde{P}_{\mathbf{i}_{l_0 - 1}} \geq P_{\max}^{(l_0)} / P_{\max}^{(l_0 - 1)} > 1, \tag{43}$$

which further leads to $P_{\mathbf{i}_{l_0}} \geq 2$, since $P_{\mathbf{i}_{l_0}}$ is an integer.

Next, we show that by replacing $P_{\mathbf{i}_{l_0}}$ by $P' = P_{\mathbf{i}_{l_0}} - 1$ in the optimal EC-SRUPQ, the cost does not increase. To this end, let us first introduce the following notation. For each $l, l \in [1:L]$, each $P \in \mathbb{Z}_+$ and each interval $C = [c, d) \subseteq [0, \infty)$, let

$$\alpha_l(C, P) = q(C)\Big(- \phi_l \operatorname{sinc}^2\left(\frac{1}{P}\right) x^2(C) + \lambda_l \log_2 P\Big).$$

Moreover, let us denote by $\beta(P_{\mathbf{i}_l})$ the portion of the cost function (20), which depends on $P_{\mathbf{i}_l}$. It can be easily seen that $\beta(P_{\mathbf{i}_l})$ can be written as follows,

$$\beta(P_{\mathbf{i}_l}) = \frac{1}{2}\alpha_l(C_{\mathbf{i}_l}, \widetilde{P}_{\mathbf{i}_l}) + \frac{1}{2}\sum_{i_{l+1}=1}^{M_{l+1,\mathbf{i}_l}} \Bigg(\alpha_{l+1}(C_{\mathbf{i}_{l+1}}, \widetilde{P}_{\mathbf{i}_{l+1}}) + \cdots + \sum_{i_L=1}^{M_{L,\mathbf{i}_{L-1}}} \Bigg(\alpha_L(C_{\mathbf{i}_L}, \widetilde{P}_{\mathbf{i}_L}) \underbrace{\Bigg) \cdots \Bigg)}_{L-l \text{ parentheses}} .$$

For each $k, l_0 + 1 \leq k \leq L$, let $P'_k = \widetilde{P}_{\mathbf{i}_{l_0 - 1}} \Pi_{j=l_0+1}^{k} P_{\mathbf{i}_j}$, where $\widetilde{P}_{\mathbf{i}_0} = 1$ by convention (to acount for the case when $l_0 = 1$). Then, $\alpha_k(C_{\mathbf{i}_k}, \widetilde{P}_{\mathbf{i}_k}) = \alpha_k(C_{\mathbf{i}_k}, P'_k P_{\mathbf{i}_{l_0}})$. Next we will prove that

$$\alpha_k(C_{\mathbf{i}_k}, P'_k P_{\mathbf{i}_{l_0}}) \geq \alpha_k(C_{\mathbf{i}_k}, P'_k P'), \tag{44}$$

for any $k, l_0 + 1 \leq k \leq L$. Let us fix some arbitrary $k, l_0 + 1 \leq k \leq L$. Notice that

$$\begin{aligned} &\alpha_k(C_{\mathbf{i}_k}, P'_k P_{\mathbf{i}_{l_0}}) = \\ &\phi_k q(C_{\mathbf{i}_k}) x^2(C_{\mathbf{i}_k}) \Big(f(P'_k P_{\mathbf{i}_{l_0}}) + \delta'_k h(P'_k P_{\mathbf{i}_{l_0}}) \Big), \end{aligned} \tag{45}$$

where $\delta'_k = \frac{\lambda_k}{\phi_k x^2(C_{\mathbf{i}_k}) \ln 2}$. Recall that the interval $C_{\mathbf{i}_k}$ has the boundaries in the set $\bar{\mathcal{A}}$. Then its largest possible left bound is a_K and its largest possible right bound is $a_{K+1} = \infty$. Also recall that $\delta'_k = \frac{\lambda_k}{\phi_k x^2([a_K, \infty)) \ln 2}$. By applying [20, Proposition 3], one obtains

$$P_{opt}(1, \delta'_k) \leq P_{opt}(1, \delta_k). \tag{46}$$

Additionally, relations (41), (42) and the definition of $P_{\max}^{(l_0)}$, with the convention that $P_{\max}^{(0)} = 0$, imply that $\widetilde{P}_{\mathbf{i}_{l_0}} > P_{0,l_0} + P_{\max}^{(l_0 - 1)} \geq P_{0,l_0} + \widetilde{P}_{\mathbf{i}_{l_0 - 1}}$, which leads to $\widetilde{P}_{\mathbf{i}_{l_0 - 1}} P' = \widetilde{P}_{\mathbf{i}_{l_0 - 1}} (P_{\mathbf{i}_{l_0}} - 1) \geq P_{0,l_0}$. Since $P'_k P' \geq \widetilde{P}_{\mathbf{i}_{l_0 - 1}} P'$, it further follows that $P'_k P' \geq P_{0,l_0}$. Next, based on the above inequality, on the fact that $P_{0,l_0} \geq P_{opt}(1, \delta_k)$ and on (46), one obtains that $P'_k P_{\mathbf{i}_{l_0}} > P'_k P' \geq P_{opt}(1, \delta'_k)$. Recall that $P_{opt}(1, \delta'_k)$ is the value of P that minimizes $f(P) + \delta'_k h(P)$. Using further Lemma 5 of [20, Appendix A], one obtains that $f(P'_k P_{\mathbf{i}_{l_0}}) + \delta'_k h(P'_k P_{\mathbf{i}_{l_0}}) \geq f(P'_k P') + \delta'_k h(P'_k P')$, which immediately implies (44) and further leads to $\beta(P_{\mathbf{i}_{l_0}}) \geq \beta(P')$. It follows that the cost of the new EC-SRUPQ is no larger than the cost of the initial one. If the new EC-SRUPQ still does not satisfy condition (21), we start all over again a similar substitution process, i.e., by finding the smallest value l_0 such that there is an $\mathbf{i}_{l_0}$ satisfying (41) and then replacing $P_{\mathbf{i}_{l_0}}$ by $P' = P_{\mathbf{i}_{l_0}} - 1$. This way, we obtain another EC-SRUPQ with a cost no larger than the previous one. Eventually, after a finite number of such substitutions, the EC-SRUPQ obtained must satisfy condition (21). Since its cost is no higher than the initial one, which was optimal, the new EC-SRUPQ must also be optimal, which contradicts the assumption made at the beginning of the proof. With this observation, the proof is completed. ■

PLACE
PHOTO
HERE

Huihui Wu received the B.Sc. degree in communication engineering from Southwest University for Nationalities, Chengdu, China, in 2011, and the M.S. degree in communication engineering from Xiamen University, Xiamen, China, in 2014. He received the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, Canada, in 2018. From November 2018 to April 2019, he was a postdoctoral research scientist at Columbia University, New York, USA. He is now a postdoctoral researcher at McGill University, Montreal, Canada. His research interests include channel coding, joint source and channel coding, signal quantization, wireless communications, blockchain and machine learning.

PLACE
PHOTO
HERE

Sorina Dumitrescu (M'05-SM'13) received the B.Sc. and Ph.D. degrees in mathematics from the University of Bucharest, Romania, in 1990 and 1997, respectively. From 2000 to 2002 she was a Postdoctoral Fellow in the Department of Computer Science at the University of Western Ontario, London, Canada. Since 2002 she has been with the Department of Electrical and Computer Engineering at McMaster University, Hamilton, Canada, where she held a Postdoctoral and a Research Associate position, and where she is currently an Associate Professor. Her current research interests include multimedia coding and communications, network-aware data compression, joint source-channel coding, signal quantization. Her earlier research interests were in formal languages and automata theory. She was a recipient of the NSERC University Faculty Award during 2007-2012.