Efficient Algorithms for Optimal Uneven Protection of Single and Multiple Scalable Code Streams against Packet Erasures

Sorina Dumitrescu, Xiaolin Wu and Zhe Wang

Dept. of Electrical and Computer Engineering

McMaster University, Hamilton, ON, Canada L8S 4K1

sorina/xwu@mail.ece.mcmaster.ca; zwang@ece.mcmaster.ca

Abstract.

We study algorithmic approaches for rate-fidelity optimal packetization of a single and multiple scalable source code streams with uneven erasure protection (UEP). A new algorithm is developed to obtain the globally optimal solution for scalable source codes of convex rate-fidelity function and for a wide class of erasure channels, including channels for which the probability of losing n packets is monotonically non-increasing in n, and independent erasure channels with packet erasure rate smaller than 0.5. This is achieved at linear space complexity and near-linear time complexity in the transmission budget, representing significant improvement over the known globally optimal algorithm. When applied to SPIHT compressed images, the results of the proposed algorithm are virtually the same as the global optima.

The above success is also extended to UEP packetization of multiple scalable code streams. We improve the existing algorithms in both speed and performance.

I. INTRODUCTION

In multimedia streaming over the Internet, the quality of service (QoS) hinges on how well the problem of packet losses is dealt with. Optimal packetization of scalable source sequences with uneven erasure protection (UEP) offers a principled solution to the problem.

Scalable compression algorithms, such as SPIHT [12] and EBCOT [15], can reconstruct a coded signal to certain fidelity from any prefix of the code stream. This feature can be exploited by Reed-Solomon (RS) codes to generate a rate-distortion optimized UEP packetization. Specifically, a collection of RS block codes of the same length but decreasing redundancy are used to protect subsequent segments of the scalable source code stream, and the packets are formed across the channel codewords. Any set of received packets can be used to reconstruct the source to some fidelity, and the fidelity increases in the number of received packets. We are interested in the problem of optimal UEP packetization under the criterion of maximizing the expected fidelity at the receiver, constrained by a given transmission budget.

The UEP packetization scheme is illustrated in Figure 1. Let N be the number of packets to be transmitted, and L the number of symbols in each packet (a symbol is a block of a fixed number of bits, usually 8 bits). In general, only a prefix of the scalable source sequence is packetized. This prefix of the source code stream is partitioned into L consecutive segments, and each of these segments is protected by an RS code. Let m_i be the length (in symbols) of the i-th source segment, then the channel code assigned to protect it, is the (N, m_i) RS code. The stream of these m_i source symbols followed by the $f_i = N - m_i$ redundancy symbols constitutes the i-th slice of the joint source-channel code. The packets are formed across the slices, i.e., the n-th packet contains the n-th symbol of each slice. The effect of the (N, m_i) RS code applied to the i-th source segment is that, if at most f_i of N packets are lost, then all the m_i source symbols of the i-th slice can be correctly recovered. However, since the scalable source sequence is only sequentially refinable, the i-th source segment can be decoded only if the previous $i-1$ segments are available. This requires that the number of redundancy symbols assigned to a slice be monotonically non-increasing in the slice index: $f_1 \geq f_2 \geq \cdots \geq f_L$, or equivalently, the number of source symbols allocated to each slice be monotonically non-decreasing in the slice index:

$$m_1 \leq m_2 \leq \cdots \leq m_L, \tag{1}$$

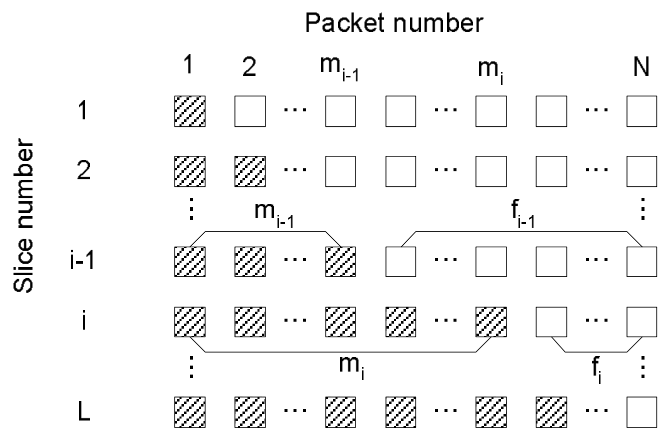

Fig. 1. Uni-group UEP packetization scheme. The slices are positioned horizontally and the packets vertically. The shaded squares represent the source symbols and the white squares represent the redundancy symbols.

Let $\mathbf{m} = (m_1, m_2, \cdots, m_L)$ be the vector whose components are the number of source symbols allocated to the slices. We call $\mathbf{m}$ the L-slice source allocation vector.

Let $\phi(r)$ be the rate-fidelity function of the scalable source sequence, which is a monotonically non-decreasing function in rate $r \in \{0, 1, 2, \cdots, R_{max}\}$, where r denotes the number of symbols in a prefix of the source sequence, and R_{max} is the total number of source symbols. Let $p_N(n)$, for $0 \leq n \leq N$, denote the probability of losing n packets out of N. The efficiency of the packetization scheme is measured by the expected fidelity of the reconstructed sequence at the decoder side, denoted by $\Phi(\mathbf{m})$. This quantity can be expressed as [10]

$$\begin{aligned} \Phi(\mathbf{m}) = P_N(N)\phi(0) + \textstyle\sum_{i=1}^{L} P_N(f_i)(\phi(r_i) - \phi(r_{i-1})) = \\ P_N(N)\phi(0) + \textstyle\sum_{i=1}^{L} P_N(N - m_i)(\phi(r_i) - \phi(r_{i-1})), \end{aligned} \tag{2}$$

where $P_N(k) = \sum_{n=0}^{k} p_N(n)$, $k = 0, 1, \cdots, N$, and $r_i = \sum_{k=1}^{i} m_k$, $1 \leq i \leq L$, $r_0 = 0$.

The problem of optimal UEP packetization under the rate-fidelity criterion can be formulated as

$$\begin{aligned} \underset{\mathbf{m}}{\text{maximize}} \quad & \Phi(\mathbf{m}) \\ \text{subject to} \quad & m_1 \leq m_2 \leq \cdots \leq m_L, \end{aligned} \tag{3}$$

for given N, L, $p_N(n)$, and $\phi(r)$.

This optimization problem has been the subject of intense research [3], [9], [10], [11], [13], [14]. Many researchers assume the convexity of the rate-fidelity function motivated by the near convexity achieved with modern scalable compression algorithms like SPIHT or JPEG 2000.

The algorithms of Puri and Ramchandran [11] and of Mohr *et al.* [9], [10] guarantee the globally optimal solution only if the rate-fidelity function is convex, and if fractional bit allocation is allowed. Under the practical constraint of integer redundancy assignment, however, these two algorithms are still suboptimal even if $\phi(r)$ is strictly convex. The algorithm of Stankovic *et al.* [13] assumes convexity of $\phi(r)$, too, but it does not need the additional assumption of fractional bit allocation. However, it can find only a local optimum. The global optimal solution for the convex setting is given by Stockhammer and Buchner [14] (an $O(N^2L^2)$ time complexity algorithm). In [3] a faster exact solution of $O(NL^2)$ running time is proposed for the case of convex rate-fidelity function and a wide class of erasure channels. Moreover, the same paper also presents the fastest exact algorithm known to date for the most general setting of the problem, i.e., when no assumptions on the rate-fidelity function or on the channel statistics are made. This is a dynamic programming algorithm of $O(N^2L^2)$ running time.

In this work we show that the efficiency of the exact solution can be further improved. We assume the convexity of the rate-fidelity function and the same additional assumptions on the channel statistics as in [3], namely that the probability of losing n packets is monotonically non-increasing in n, or that the channel is an independent erasure channel with packet erasure rate no larger than $\frac{N}{2(N+1)}$. The new algorithm proposed for this setting is based on a Lagrangian formulation of the problem. For each value of the Lagrangian multiplier λ, the algorithm takes $O(NL)$ time. The number of iterations needed to find the optimal λ, and hence to complete the algorithm, is much smaller than L, leading to great savings of computations from the $O(NL^2)$ time algorithm of [3]. The memory usage also drops from $O(NL^2)$ in [3] to $O(NL)$. The saving in memory partially accounts for the increased speed of the new algorithm.

We also investigate the problem of optimal UEP packetization of multiple scalable code streams posed in [6], and improve the existing algorithm in both speed and performance. In the multiple streams variant of the optimal UEP design problem, which has applications in multimedia communication, K code streams representing K contents are first packetized separately, each into N packets of some small length. Then the small packets, one from each object, are concatenated to produce N large packets of size L each. The optimization problem is to find the optimal allocation of slices between objects and of redundancy symbols for each object such that the fidelity is maximized. We maintain the same assumption on the channel statistics and on the rate-fidelity function as above, and propose for this setting two globally optimal algorithms faster than that of [6] for this problem.

The paper is structured as follows. In the next section we prove that the optimal UEP packetization problem can be formulated as a maximum-weight path problem constrained on the number of edges, in a certain weighted directed acyclic graph. This graph has some nice properties which induce the convexity of the optimization problem. This allows us to transform the constrained problem to the unconstrained problem of minimizing the Lagrangian. This is the topic of Section 3. We also show that for each value of the Lagrangian multiplier λ, the unconstrained problem can be efficiently solved in $O(NL)$ time. Specifically, binary search suffices to find the optimal λ (i.e., for which the constraint is satisfied). Section 4 examines the problem of multiple scalable codestream UEP packetization introduced in [6] and presents a globally optimal algorithm of the same time complexity as the suboptimal algorithm in [6]. In section 5 we present a Lagrangian solution for this problem. Experimental results are reported in Section 6 to verify the improved efficiency and good rate-fidelity performance of the new algorithms.

II. GRAPH MODELING OF THE PROBLEM

We assume that the rate-fidelity function $\phi(r)$ is convex[1] and that $p_N(n)$ is non-increasing in n (the other case when the channel is an independent packet erasure channel will be discussed at the end of this section). It was shown in [3] that for convex $\phi(r)$ (without any restriction on the channel) the optimal UEP packetization can be computed by maximizing the expression (2) without imposing the constraint (1) (because the solution will satisfy the constraint anyway). Consequently, this result holds in our case, too. The first step in our development is to show that maximizing (2) is equivalent to solving a maximum-weight path problem constrained on the number of edges.

Consider the weighted directed acyclic graph $G = (V, E)$, whose nodes (or vertices) are identified with nonnegative integer numbers between 0 and M, where $M = \min(R_{max}, NL)$,

[1]All over this paper the term "convex" refers to "upward convex". In other words, $\phi(r)$ is convex if and only if for any r_1 and r_2 the relation $\phi(1/2r_1 + 1/2r_2) \geq 1/2\phi(r_1) + 1/2\phi(r_2)$ holds.

hence the set of vertices is $V = \{0, 1, 2, \cdots, M\}$, and any two nodes u, v such that $0 < v-u \leq N$ are connected by an edge, hence the set of edges is $E = \{(u, v)|0 \leq u < v \leq M, v - u \leq N\}$. The weight of an edge (u, v) is defined to be $w(u, v) = P_N(N - v + u)(\phi(v) - \phi(u))$. Let the source vertex of the graph be 0 and let the set of final vertices coincide with V.

A path in the graph is any sequence of nodes such that any two consecutive nodes are connected by an edge. The weight of the path is the sum of weights of edges connecting the consecutive nodes. Note that any L-slice source allocation vector $\mathbf{m}$, not necessarily satisfying the constraint (1), can be associated to a path of L edges in the graph G, starting at the source node and ending at a final node, namely the path: $r_0, r_1, \cdots, r_L$, where $r_0 = 0$. For each $i \geq 1$, the i^{th} edge on this path (r_{i-1}, r_i) corresponds to the segment of source symbols on the i^{th} slice of the UEP packetization scheme. Moreover, the edge's weight equals the contribution of this segment to the expected fidelity (2). Therefore, the weight of the path equals the value $\Phi(\mathbf{m}) - P_N(N)\phi(0)$. This correspondence between source allocation vectors and paths in G is one to one. Consequently, the problem of optimal UEP packetization is equivalent to the problem of finding the path of maximum weight among all the paths from the source to a final node, which have exactly L edges (the maximum-weight L-edge path problem).

For the convenience of our algorithm development we make the graph G to be complete, i.e., a graph where each ordered pair (u, v) of vertices with $u < v$, forms an edge, by setting to $-\infty$ the weight of pairs $(u, v) \notin E$. The following proposition states a special property of this complete graph which is essential to the complexity reduction of our new algorithm in the next section. In order not to interrupt the flow of the ideas we defer its proof to the Appendix.

Proposition 1. The graph G satisfies the so-called Monge property, i.e.

$$w(u_1, v_1) + w(u_2, v_2) \geq w(u_1, v_2) + w(u_2, v_1), \tag{4}$$

for all u_1, u_2, v_1, v_2 such that $r_0 \leq u_1 < u_2 < v_1 < v_2 \leq M$.

III. Lagrangian Relaxation-based Solution

The maximum-weight L-edge problem, as stated above, is a constrained optimization problem, where the constraint is on the number of edges in the path. It is well known that in the case of convex objective function, the constrained problem can be transformed to an unconstrained one through Lagrangian relaxation. We will show that this is the case for our problem.

Let $\mathcal{P}$ denote the set of all paths from the source node to any terminal node, in the graph G. For any path $P \in \mathcal{P}$ let $W(P)$ denote its weight and $L(P)$ its length (the number of its edges). Consider the set of planar points $\mathcal{U} = \{(L(P), W(P)) | P \in \mathcal{P}\}$.

The problem of maximum-weight L-edge path in G can be formulated as

$$\begin{aligned} &\text{maximize}_{P \in \mathcal{P}} W(P) \\ &\text{subject to } L(P) = L. \end{aligned} \tag{5}$$

The underlying Lagrangian is $J(\lambda, P) = W(P) + \lambda L(P)$, over all paths $P \in \mathcal{P}$ and all real values λ. A path P_λ maximizes the Lagrangian $J(\lambda, P)$ for some λ, i.e., the relation

$$P_\lambda = \max_{P \in \mathcal{P}} J(\lambda, P), \tag{6}$$

holds if and only if the planar point $(L(P_\lambda), W(P_\lambda))$ is on the upper convex hull of $\mathcal{U}$ and the line of slope $-\lambda$ passing through this point is a support line to $\mathcal{U}$ [8], [4]. Thus, if (6) holds then the path P_λ is also the maximum-weight $L(P_\lambda)$-edge path because the upper boundary of $\mathcal{U}$ is not above its upper convex hull. Consequently, if a Lagrangian multiplier λ can be found such that the path P_λ to be of length L, then this path is the solution of the constrained problem (5). Due to the following proposition, whose proof is given in Appendix, such a multiplier λ is guaranteed to exist.

Proposition 2. The inequality

$$2\bar{W}(l) \geq \bar{W}(l-1) + \bar{W}(l+1) \tag{7}$$

holds for all $l, 2 \leq l \leq M-1$, where $\bar{W}(l)$ denotes the weight of the maximum-weight l-edge path from the source to a final node, in the graph G.

The above proposition implies that the point $(L, \bar{W}(L))$ is on the upper convex hull of $\mathcal{U}$. Therefore there is some real value λ_0 such that $L(P_{\lambda_0}) = L$. Then P_{λ_0} is the solution of the constrained problem (5). Moreover, the relation $L(P_{\lambda_0}) = L$ is valid if and only if the following inequalities hold

$$\bar{W}(L) - \bar{W}(L-1) \geq -\lambda_0 \geq \bar{W}(L+1) - \bar{W}(L) \tag{8}$$

meaning that $-\lambda_0$ corresponds to the slope of any support line to the curve $(\cdot, \bar{W}(\cdot))$, passing through the point $(L, \bar{W}(L))$.

Therefore, the L-edge maximum-weight path can be found by solving (6) in conjunction with a search on λ until the number of the edges on the path becomes exactly L. To this end we derive from G a parameterized graph $G(\lambda)$ by adding λ to the weight of each edge of G. In the resulting parameterized graph $G(\lambda)$ the maximization problem of (6) reduces to an unconstrained maximum-weight path problem. This is because $J(\lambda, P)$ equals the weight of the path P in $G(\lambda)$.

The maximum-weight path in a weighted directed acyclic graph can be found by standard algorithms in $O(|V| + |E|)$ time. However, for graphs with the Monge property there is a faster solution of $O(|V|)$ time complexity [16]. By Proposition 1 the graph G satisfies the Monge property, consequently $G(\lambda)$ satisfies the Monge property, too (the inequality in (4) still holds if we add 2λ on each side). Therefore, the maximum-weight paths from the source to each node can be found in $O(NL)$ time and space by using the algorithm proposed in [16]. Further, by computing the maximum of these paths, in no more than $O(NL)$ time, the maximum-weight path of the graph is found. Consequently, for each λ, the maximization of (6) is solved in $O(NL)$ time and space.

Relations (8) imply that the length of P_λ is non-decreasing as the parameter λ increases [8], [1]. Therefore, to find the optimal λ_0, we use bisection search.

Before presenting the search algorithm an observation is due. For some values of λ, there may be several paths to maximize the Lagrangian, some of equal lengths, but also some of different lengths. The latter situation occurs when $-\lambda$ equals the slope of a convex hull edge of the curve $(\cdot, \bar{W}(\cdot))$. Then for each l_0 such that $(l_0, \bar{W}(l_0))$ is on this convex hull edge, there is a path of length l_0 maximizing the Lagrangian. For each λ, our proposed algorithm finds a maximum-weight path in $G(\lambda)$ with the maximal possible number of edges.

In the bisection search, a search interval for λ, $[\lambda_{low}, \lambda_{high}]$ is maintained at any time. Initially, $\lambda_{low} = -\frac{\phi(NL)}{L}$ and $\lambda_{high} = 0$. At the beginning of each iteration, the current value of λ is set to $(\lambda_{low} + \lambda_{high})/2$. If $L(P_\lambda) = L$ the algorithm stops. Otherwise, depending on whether $L(P_\lambda) < L$ or $L(P_\lambda) > L$, the search interval $[\lambda_{low}, \lambda_{high}]$ is updated to $[\lambda, \lambda_{high}]$ or $[\lambda_{low}, \lambda]$ respectively. This technique ensures that the search interval for λ becomes smaller after each iteration. However, since the path lengths $L(P_\lambda)$ take values only in a finite set, it follows that the interval $[L(P_{\lambda_{low}}), L(P_{\lambda_{high}})]$ (which is guaranteed to include L) may remain unchanged after some iterations. The first time it happens (but after λ_{low} and λ_{high} have both changed from their initial values), we switch to another strategy for updating λ, namely $\lambda = (\lambda_{high} - \lambda_{low})/(L(P_{\lambda_{low}}) -$

$L(P_{\lambda_{high}}))$. If after this switch, the interval of lengths $[L(P_{\lambda_{low}}), L(P_{\lambda_{high}})]$ does not change after some iterations, then we stop concluding that the current λ is the optimal one. This situation corresponds to the case when the function $\bar{W}(l)$ is linear for $l \in [L(P_{\lambda_{low}}), L(P_{\lambda_{high}})]$. Further, the desired L-edge path can be constructed from $P_{\lambda_{low}}$ and $P_{\lambda_{high}}$ in $O(L)$ time, as described in the proof of Lemma 2 (with $l_1 = L(P_{\lambda_{low}})$, $l_2 = L(P_{\lambda_{high}})$ and $i = \min\{L(P_{\lambda_{high}}) - L, L - L(P_{\lambda_{low}})\}$).

Optimal UEP packetization can be computed in $O(\tau NL)$ time, where τ is the number of iterations needed to find the optimal λ. For channels with non-increasing $p_N(n)$ and for convex rate-fidelity curves. We have empirically found that on average τ does not depend on N and increases very slowly with L (at a growth rate close to $O(\log L)$). Thus the new algorithm is much faster than the $O(NL^2)$ time algorithm of [3]. Moreover, the space complexity of the new algorithm is linear in the transmission budget, i.e. $O(NL)$ as opposed to $O(NL^2)$ in [3]. This saving is due to the fact that the current path does not have to be stored from one iteration to the next.

Assume now an independent erasure channel with packet erasure rate no larger than $\frac{N}{2(N+1)}$. Let $n_0 = \lfloor \epsilon(N+1) \rfloor$. It was proven in [3] that an optimal L-slice source allocation vector $\mathbf{m}$ exists such that $m_i \leq N - n_0$ for all i. Then the graph G is constructed such that only edges (u, v) with $v - u \leq N - n_0$, to have finite weights (defined as previously), and all the other edges to have the weight $-\infty$. It was also shown in [3] that $p_N(n)$ is nonincreasing for $n \geq n_0$, which is the crucial ingredient to show that the modified graph satisfies the Monge property. Further, the same development applies as in the previous case.

IV. UEP Packetization of Multiple Code Streams

The above advances in algorithmic approach to conventional optimal UEP packetization also bring progress in design algorithms for optimal UEP packetization of multiple scalable codestreams, an important problem for multimedia communications which was first posed by Gan and Ma [6].

The problem proposed by [6] is the following. Assume there are K scalable code streams to be transmitted together using N packets of payload L each. The separate code streams may be obtained from separate encoding of different objects of the same image. Each object is allocated a number of symbols within each packet. Let l_k denote the number of symbols allocated to object k. Clearly, $\sum_{k=1}^{K} l_k = L$. Then the code stream corresponding to object k is packetized

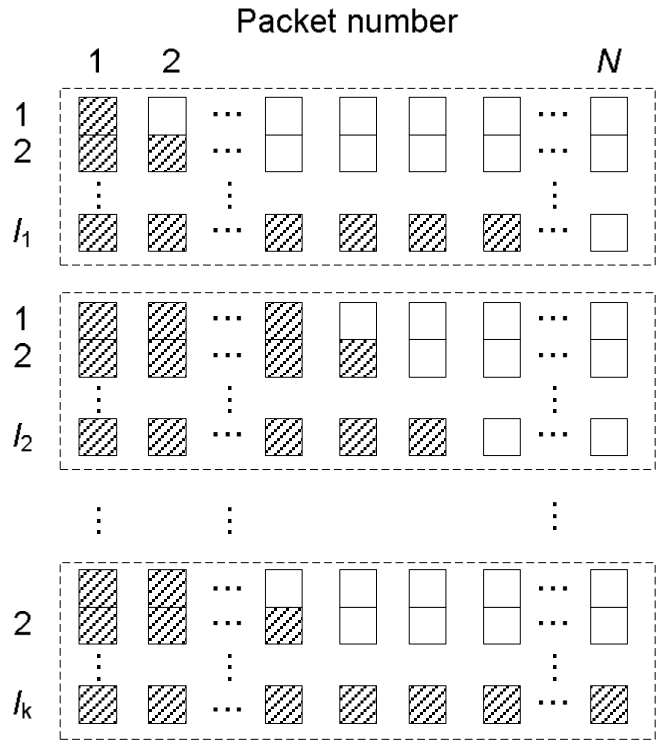

Fig. 2. Multiple stream UEP packetization scheme. The slices are positioned horizontally and the packets vertically. The shaded squares represent the source symbols and the white squares represent the redundancy symbols.

within the UEP framework into N small packets of size l_k each. Finally, the small packets, one from each object, are concatenated to form the large packets of size L.

Figure 2 illustrates the packetization scheme. It still consists of L slices, each of N symbols, the packets being formed across slices. The k-th object is assigned l_k slices out of the total number, more precisely, the slices from the $(l_1 + \cdots + l_{k-1} + 1)$-th to the $(l_1 + \cdots + l_{k-1} + l_k)$-th. A prefix of object k's code stream is divided into non-overlapping consecutive segments of lengths $m_{k,1}, m_{k,2}, \cdots, m_{k,l_k}$, respectively. The i-th segment is protected with an $(N, m_{k,i})$ RS code. The obtained channel codeword forms the i-th slice allocated to object k, i.e., the $(l_1+\cdots+l_{k-1}+i)$-th slice in the global scheme. The l_k-slice source allocation vector $\mathbf{m}_k = (m_{k,1}, m_{k,2}, \cdots, m_{k,l_k})$ must have the components in non-decreasing order, i.e.

$$m_{k,1} \leq m_{k,2} \leq \cdots \leq m_{k,l_k}. \tag{9}$$

Let $r_{k,0}, r_{k,1}, \cdots, r_{k,l_k}$, denote the partition positions of the k-th code stream, i.e., $r_{k,0} = 0$ and $r_{k,i} = \sum_{j=1}^{i} m_{k,j}$. Let $\phi_k(r)$ denote its fidelity function. Then the expected fidelity of the reconstructed object k at the receiver is

$$\Phi_k(l_k, \mathbf{m}_k) = P_N(N)\phi_k(0) + \sum_{i=1}^{l_k} P_N(N - m_{k,i})(\phi_k(r_{k,i}) - \phi_k(r_{k,i-1})). \tag{10}$$

Assuming that the global fidelity is additive in the the separate object's fidelities, the global expected fidelity at the decoder is

$$\sum_{k=1}^{K} \Phi_k(l_k, \mathbf{m}_k). \tag{11}$$

As discussed in [6] the terms in the above sum can be weighted differently according to each object's importance. We assume here that each fidelity function $\phi_k(r)$ is already scaled by the weighting factor assigned to object k.

The problem of optimal UEP packetization of multiple scalable code streams is then

$$\begin{aligned} \underset{l_1,\cdots,l_K,\mathbf{m}_1,\cdots\mathbf{m}_K}{\text{maximize}} \quad & \sum_{k=1}^{K} \Phi_k(l_k, \mathbf{m}_k) \\ \text{subject to} \quad & \sum_{k=1}^{K} l_k = L, \\ & m_{k,1} \leq m_{k,2} \leq \cdots \leq m_{k,l_k}, \text{ for each } k. \end{aligned} \tag{12}$$

As observed in [6], for fixed values $l_1, \cdots, l_K$, the quantity (11) can be maximized over all $\mathbf{m}_1, \cdots \mathbf{m}_K$, by maximizing each $\Phi_k(l_k, \mathbf{m}_k)$ separately. Let $\bar{\Phi}_k(l_k)$ denote the solution of optimal UEP packetization of code stream k into N packets of size l_k. In other words,

$$\bar{\Phi}_k(l_k) = \max_{m_{k,1} \leq \cdots \leq m_{k,l_k}} \Phi_k(l_k, \mathbf{m}_k). \tag{13}$$

Then problem (12) is equivalent to

$$\begin{aligned} \underset{l_1,\cdots,l_K}{\text{maximize}} \quad & \sum_{k=1}^{K} \bar{\Phi}_k(l_k) \\ \text{subject to} \quad & \sum_{k=1}^{K} l_k = L. \end{aligned} \tag{14}$$

If the quantities $\bar{\Phi}_k(l_k)$ are known then problem (14) of optimal slice allocation between objects, is a classical resource allocation problem with integer variables and separable objective function, for which algorithmic solutions are well known [7]. However, to solve the overall problem (12), the algorithm to solve (14) must be aided by an algorithm for optimal UEP packetization of single code stream, in order to compute the necessary values $\bar{\Phi}_k(l_k)$.

The authors of [6] use two alternative methods to solve the slice allocation problem (14), the dynamic programming approach and the greedy approach, combining them with the local search algorithm of [13] for solving single UEP packetization. The two methods to solve the overall problem (12) have time complexities $O(KNL^2)$ and $O(NL^2)$, respectively. As specified in [6], the dynamic programming approach ensures global optimality of the solution for the slice allocation problem, while the greedy approach can guarantee the global optimality only if the functions $\bar{\Phi}_k(l_k)$ are convex in l_k. We add to this observation that, in either case, in order to guarantee the globally optimal solution of the overall problem, the algorithm used to solve the single UEP packetization must be globally optimal, too. However, since the local search algorithm in [6] does not satisfy this requirement, neither of the two methods ensures global optimality. Moreover, although the greedy approach is motivated by the empirical observations on the near convexity of the functions $\bar{\Phi}_k(l_k)$, in l_k, the authors did not clarify under which conditions exact convexity holds.

This open problem is now settled. The functions $\bar{\Phi}_k(l_k)$ are indeed convex under the conditions assumed in this paper on the channel statistics and on the rate-fidelity curves. This follows from Proposition 2 since $\bar{\Phi}_k(l_k)$ is the weight of the maximum weight l_k-edge path in the graph G_k assigned to object k, using the graph model of Section 2. Secondly, we show that the globally optimal solution to the multiple code streams UEP packetization can be obtained in $O(NL^2)$ time, i.e., at the same time complexity as the fastest of the two suboptimal algorithms in [6]. Thirdly, we propose alternative Lagrangian algorithms that are faster than Gan and Ma's suboptimal algorithms when K is small, while still ensuring global optimality.

The first of our algorithms uses the greedy method for the slice allocation problem (14), which can be described as follows.

1. Initialize all $l_k, 1 \leq k \leq K$, to 0.
2. Find $k_0 = \underset{1\leq k\leq K}{\text{argmax}}(\bar{\Phi}_k(l_k+1) - \bar{\Phi}_k(l_k))$.
3. Increment l_{k_0} by 1. If $\sum_{k=1}^{K} l_k < L$ go to Step 2. Else Stop.

Note that Steps 2 and 3 iterate L times. At each iteration, in order to find k_0, the quantity $\bar{\Phi}_k(l_k+1)$ needs to be computed only for a single k (namely, the previous k_0), because for the other k's these values are already known. Apparently, the computation of $\bar{\Phi}_k(l_k+1)$ here requires $O(Nl_k^2)$ time if the globally optimal algorithm with fast matrix search of [3] is used, implying an overall time complexity of $O(NL^3)$. However, the incremental structure of the dynamic programming solution of [3] allows for $\bar{\Phi}_k(l_k+1)$ to be computed in $O(Nl_k)$ time, without running the algorithm of [3] from the beginning, but using quantities already found at previous iterations. Thus, the overall time complexity becomes $O(NL^2)$.

More precisely, the fast algorithm of [3] can be alternatively described using the graph modelling of the problem outlined in Section 2, as follows. To compute $\bar{\Phi}_k(l)$, the algorithm proceeds in l stages. At each stage i, the maximum-weight i-edge path from the source to node a, is computed for each node a. Denote by $\hat{\Phi}_k(i,a)$ the weight of this path. The computation of all $\hat{\Phi}_k(i,a)$ is performed based on the values found at the previous step. Moreover, the fast matrix search technique proposed in [2] is used to perform these computations in $O(Nl)$ time. Finally, at the end of stage l, $\bar{\Phi}_k(l)$ is computed by

$$\bar{\Phi}_k(l) = \max_a \hat{\Phi}_k(l,a). \tag{15}$$

We adapt this algorithm for our purpose by adding at the end of each stage i the evaluation of $\bar{\Phi}_k(i)$ through solving

$$\bar{\Phi}_k(i) = \max_a \hat{\Phi}_k(i,a). \tag{16}$$

Note that the time complexity of stage i remains unchanged. Further, in order to compute $\bar{\Phi}_k(l_k+1)$ at some iteration we only need to run stage l_k+1 of the above outlined algorithm. This proves our complexity claim.

V. Lagrangian-based Solution for Multiple Code Streams UEP Packetization

The second algorithm is based again on the Lagrangian multiplier technique. Since the objective function of the slice allocation problem (14) is convex (as a sum of convex functions), the problem can be solved [4] by maximizing the underlying Lagrangian $J'(\lambda, l_1, \cdots, l_K)$ defined as

$$J'(\lambda, l_1, \cdots, l_K) = \textstyle\sum_{k=1}^{K} \bar{\Phi}_k(l_k) + \lambda \sum_{k=1}^{K} l_k, \tag{17}$$

for various Lagrangian multipliers λ, until the condition

$$\sum_{k=1}^{K} l_k(\lambda) = L, \tag{18}$$

is met, where $l_k(\lambda)$, $1 \leq k \leq K$, denote the values of l_k for which $\max_{l_1,\cdots,l_K} J'(\lambda, l_1, \cdots, l_K)$ is achieved. It is easy to see that the above maximization can be performed by solving separately

$$\max_{l_k}(\bar{\Phi}_k(l_k) + \lambda l_k), \tag{19}$$

for each k. Further, the solution of (19) can be found by solving separately problem (6) for each code stream k, i.e., finding the maximum-weight path in each graph $G_k(\lambda)$, for the same λ. This solution not only gives $l_k(\lambda)$ (i.e. the number of slices allocated to object k), but also the source allocation vector for object k. To find the optimal Lagrangean multiplier λ for which (18) holds, binary search can again be used because $\sum_{k=1}^{K} l_k(\lambda)$ is non-decreasing as λ increases.

To summarize, for given λ, the algorithm proceeds by solving the maximum-weight path in each graph $G_k(\lambda)$. Then λ is updated by the rules of binary search until condition (18) is met. The algorithm runs in $O(\tau KNL)$ time, with τ being the number of iterations until the optimal λ is found. In our experiments, for the case of $K = 2$, the number of iterations τ exhibits the same tendencies as in the case of single sream UEP packetization, i.e., it does not depend on N and increases very slowly with L (at a growth rate close to $O(\log L)$). Therefore, as a practical solution, this algorithm is more efficient than that of the preceding section, hence than those of [6] as well, when the number K of objects is small.

VI. Experimental Results

We have tested the new algorithm for single stream UEP packetization on seven images compressed by SPIHT [12]. The images and their sizes are: barb (576×720), boat (576×720), lena (512×512), zelda (512×512), craft (3072×2048), hat (3072×2048) and motor (3072×2048). The fidelity measure used is the PSNR. In order to have exact convexity of the rate-fidelity curve, we approximated the real PSNR curve by its upper convex hull (the same approximation was also used by other researchers [10], [11], [13]).

In order to test the number of iterations τ, we ran the new algorithm for different values of L (from 50 to 200, in increments of 25) and different values of N (from 50 to 200, in increments of 25). In our experiments we simulated packet erasure channels with exponentially decreasing $p_N(n)$ and different mean packet loss rates: $0.15, 0.2, 0.25, 0.3$. The number of iterations for all our tests ranges between 2 and 14 with an average of 9.61. The extreme values 2 and 14 were statistical outliers.

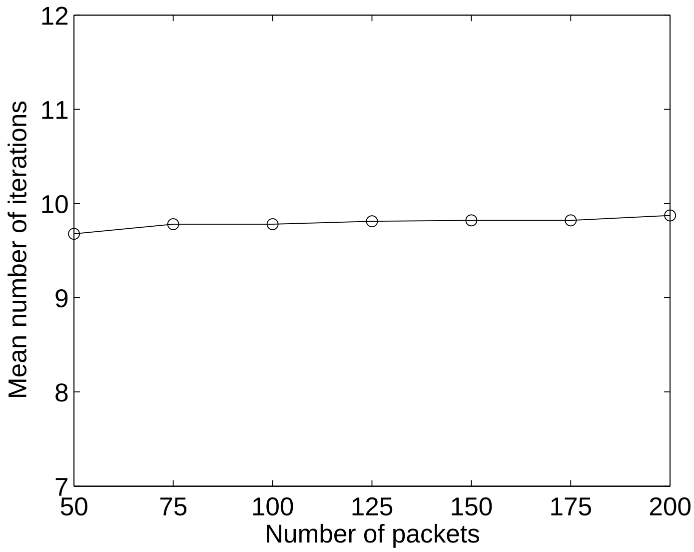

Fig. 3. Average number of iterations τ versus the number of packets N, in the case of single stream UEP packetization. The average is computed over all images, all mean packet loss rates and all symbol lengths.

An interesting observation from our experiments is that on average, the number of iterations does not depend on the number of packets N, and increases very slowly with L (at a rate close to $O(\log L)$). The average values of τ versus N and $\log_{10} L$ are plotted in Figures 3 and 4. These experimental results indicate the average time complexity of $O(NL \log L)$ for our new single stream packetization algorithm.

The proposed algorithm is globally optimal for convex rate-fidelity curves of scalable source code, but it is in general an approximation for practical codes used in multimedia communication. To assess the quality of this approximation we compared the new algorithm against the globally optimal algorithm of [3]. We performed tests on all seven images and all parameter choices as above. The new algorithm achieved solution within 0.01 dB of the optimal one in 78% of the total, and within 0.02 dB of the optimal one in 90% of all cases, the maximum deviation from the optimal being 0.16 dB. Therefore, for all practical purposes one can use the new faster algorithm with confidence that the solutions are very close to the theoretical optimum.

We have also evaluated the speed of the Lagrangian algorithm proposed for the multiple stream packetization. The setting for our experiments is similar to the one in [6]. We considered $K = 2$ and obtained the two objects by manually segmenting the lena image into a foreground

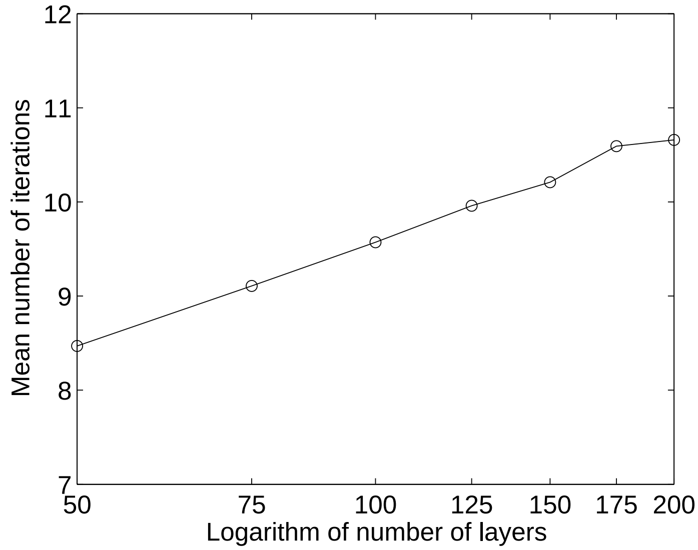

Fig. 4. Average number of iterations τ versus $\log_{10} L$, in the case of single stream UEP packetization. The average is computed over all images, all mean packet loss rates and all number of packets N.

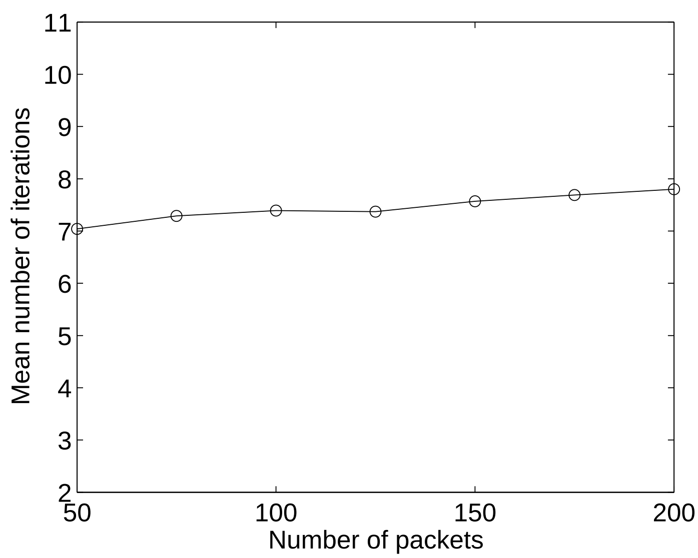

Fig. 5. Average number of iterations τ versus the number of packets N, in the case of multiple streams UEP packetization. The average is computed over all mean packet loss rates and all symbol lengths.

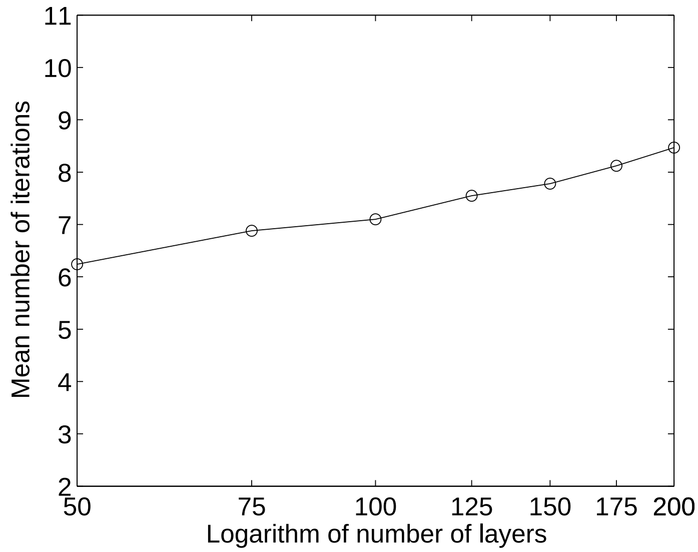

Fig. 6. Average number of iterations τ versus $\log_{10} L$, in the case of multiple streams UEP packetization. The average is computed over all mean packet loss rates and all number of packets N.

object and the background. The shape-adaptive SPIHT algorithm [5] was used to encode the two objects. We ran the Lagrangian algorithm of Section 5 for a similar choice of parameters as in the previous set of experiments. The observations on the number of iterations are consistent with those for the single stream case. Namely, the number of iterations varies roughly in the same range $(2$ to $11)$. It does not depend on N, and it increases very slowly with L (at a rate close to $O(\log L)$). These results are presented in Figures 5 and 6, and they show that the average time complexity of the Lagrangian-based algorithm for optimal multiple stream packetization is $O(KLN \log L)$.

VII. Conclusion

We have proposed a new efficient algorithm for UEP packetization of scalable source code streams. The algorithm finds the globally optimal solution for scalable code streams of convex rate-fidelity function over a large class of erasure channels. The space and time complexities of the new algorithm are linear and near linear in the transmission budget NL, representing a significant improvement over the previous $O(NL^2)$ space and time algorithm. For real SPIHT-compressed images, the new algorithm obtains solutions extremely close to the globally optimum.

Moreover, the above new optimal design approach was extended to UEP packetization of multiple scalable code streams. We found conditions in which the design objective function is convex. A new design algorithm was proposed to obtain the globally optimal solution for multiple scalable code streams in $O(NL^2)$ time, or at the same time complexity as the fastest suboptimal algorithm in [6]. We also developed alternative Lagrangian algorithms that are faster than existing suboptimal algorithms for small K, without compromising global optimality.

Appendix. Proofs of Propositions.

Here we prove Propositions 1 and 2. Before proving each of these results, we will first restate it.

Proposition 1. The graph G satisfies the so-called Monge property, i.e.

$$w(u_1, v_1) + w(u_2, v_2) \geq w(u_1, v_2) + w(u_2, v_1), \tag{20}$$

for all u_1, u_2, v_1, v_2 such that $r_0 \leq u_1 < u_2 < v_1 < v_2 \leq M$.

Proof. It is sufficient to show the following inequality

$$w(u, v) + w(u+1, v+1) \geq w(u, v+1) + w(u+1, v), \tag{21}$$

for all u, v such that $r_0 \leq u, u+1 < v, v+1 \leq M$. Then (4) follows easily by induction on $u_2 - u_1$ and $v_2 - v_1$.

Recall that some of the edges of the graph G have the weight $-\infty$, more precisely, all edges (a, b) such that $b - a > N$. Therefore we consider first the case when some of the weights involved in inequality (21) are $-\infty$. There are three situations leading to this:

1) $v - u - 1 > N$. In this case all weights are $-\infty$, hence (21) holds with equality.
2) $v - u - 1 = N$. In this case all weights except $w(u+1, v)$ are $-\infty$, which again makes the two sides of (21) to be $-\infty$. Thus, (21) holds again with equality.
3) $v - u = N$. Only the term $w(u, v+1)$ is $-\infty$, hence (21) holds again.

Assume now that all the terms in (21) are finite, i.e., $v+1-u \leq N$. Then, using the definition of the edges' weights in the graph G, relation (21) becomes

$$\begin{aligned} P_N(N-v+u)[\phi(v) - \phi(u)] + P_N(N-v-1+u+1)[\phi(v+1) - \phi(u+1)] \geq \\ P_N(N-v-1+u)[\phi(v+1) - \phi(u)] + P_N(N-v+u+1)[\phi(v) - \phi(u+1)]. \end{aligned} \tag{22}$$

The next step is to replace $P_N(N-v-1+u)$ by $P_N(N-v+u)-p_N(N-v+u)$ and $P_N(N-v+u+1)$ by $P_N(N-v+u)+p_N(N-v+u+1)$. After that the terms containing the factor $P_N(N-v+u)$ are grouped together. Since their sum is 0, inequality (22) becomes

$$0 \geq -p_N(N-v+u)[\phi(v+1)-\phi(u)]+p_N(N-v+u+1)[\phi(v)-\phi(u+1)]. \tag{23}$$

The proof of the proposition follows by combining

$$\phi(v+1)-\phi(u) \geq \phi(v)-\phi(u+1) \geq 0, \tag{24}$$

and

$$p_N(N-v+u) \geq p_N(N-v+u+1) \geq 0. \tag{25}$$

Relation (24) is true because the function $\phi(\cdot)$ is non-decreasing, and (25) is true because $p_N(\cdot)$ is non-increasing. $\square$

Before proving Proposition 2 we make the observation that such a convexity result was shown in [1] (Corollary 7 [1]) for a complete graph with Monge property, but only one terminal node. In our graph all nodes are terminal, therefore the result of [1] is not applicable. Moreover, the edge weights in our graph have an additional property (incurred by the convexity of the fidelity function $\phi(\cdot)$), which makes Proposition 2 hold. However, in our proof we will use an intermediary result of [1], which is stated next.

Lemma 1 ([1]). Let a and b be two vertices in G with $a \leq b$. Let further P_a be a path of l_a edges from the source to node a, and P_b be a path of l_b edges from the source to node b, such that $l_a > l_b$. Then for any integer $i, 0 < i \leq l_a - l_b$, there is a path Q_a of $l_a - i$ edges from the source to a, and a path Q_b of $l_b + i$ edges from the source to b, such that

$$W(Q_a)+W(Q_b) \geq W(P_a)+W(P_b). \tag{26}$$

The above result is Lemma 6 of [1]. Since the construction of the two paths Q_a and Q_b is needed in our algorithm we briefly describe it following the idea in [1]. Notice first that the case when $a = b$ and $i = l_a - l_b$ is trivial because we can choose Q_a to be P_b and Q_b to be P_a. We treat next the non-trivial case. Let the path P_a be $a_0, a_1, a_2, \cdots, a_{l_a}$ with $a_0 = 0, a_{l_a} = a$, and let P_b be $b_0, b_1, b_2, \cdots, b_{l_b}$, where $b_0 = 0, b_{l_b} = b$. Consider the largest integer j in the range from 0 to l_b with the property that $b_j \leq a_{j+i}$. Such an integer is guaranteed to exist because $b_0 \leq a_{0+i}$. Moreover, $j \neq l_b$ because $b_{l_b} > a_{l_b+i}$. Hence, $j \leq l_b - 1$. Further, from the definition of j it follows that $b_{j+1} > a_{j+i+1}$. Define now the path Q_a as $b_0, b_1, \cdots, b_{j-1}, b_j, a_{j+i+1}, a_{j+i+2}, \cdots, a_{l_a}$ and the path Q_b as $a_0, a_1, \cdots, a_{j+i-1}, a_{j+i}, b_{j+1}, b_{j+2}, \cdots, b_{l_b}$. Clearly, Q_a has $l_a - i$ edges and Q_b has $l_b + i$ edges. Using further the inequalities $b_j \leq a_{j+i} < a_{j+i+1} < b_{j+1}$, and the Monge property of the graph G (Proposition 1), relation (26) easily follows.

In order to prove Proposition 2 we need the following lemma, too.

Lemma 2. Consider the paths P_1 and P_2 to be the l_1-edge, respectively the l_2-edge, maximum-weight paths in G, for some integers $0 < l_1 < l_2$. Then for any integer $i, 0 < i \leq (l_a - l_b)/2$, there are two paths Q_1 and Q_2, of $l_1 + i$, respectively, $l_2 - i$ edges, both starting from the source of the graph, such that

$$W(Q_1) + W(Q_2) \geq W(P_1) + W(P_2). \tag{27}$$

Proof. The situation when $l_1 = l_2 - 1$ is trivial, therefore we will assume that $l_1 < l_2 - 1$. Let the path P_1 be $r_0, r_1, \cdots, r_{l_1}$, and let the path P_2 be $r'_0, r'_1, \cdots, r'_{l_2}$, where $r_0 = r'_0 = 0$. We need to distinguish further between three cases.

Case 1: $r'_{l_2} \leq r_{l_1}$. The paths Q_1 and Q_2 are constructed as in Lemma 1.

Case 2: $r'_{l_2-i} \leq r_{l_1} < r'_{l_2}$. Let P'_2 denote the $(l_2 - i)$-edge path from 0 to r'_{l_2-1} obtained from P_2 by removing the last i edges. Since $i \leq l_a - (l_b - i)$ we can apply Lemma 1, and hence construct an $(l_1 + i)$-edge path Q_1 from 0 to r_{l_1}, and an $(l_2 - 2i)$-edge path Q'_2 from 0 to r'_{l_2-i} such that

$$W(Q_1) + W(Q'_2) \geq W(P_1) + W(P'_2) \tag{28}$$

holds. Now construct Q_2 by appending to Q'_2 the last i edges of P_2 (i.e., the portion of P_2 between $(r'_{l_2-i}$ and $r'_{l_2})$. Clearly, (27) is satisfied.

Case 3: $r_{l_1} < r'_{l_2-i}$. Let Q_2 be the prefix of the path P_2 up to the node r'_{l_2-i}. Further, Q_1 is obtained by appending to P_1 the path $r_{l_1}, r_{l_1+1}, \cdots, r_{l_1+i}$, where, for $1 \leq j \leq i$, $r_{l_1+j} = r_{l_1+j-1} + r'_{l_2-i+j} - r'_{l_2-i+j-1}$. Then, in order to prove (27) it is enough to show that $w(r_{l_1+j-1}, r_{l_1+j}) \geq w(r'_{l_2-i+j-1}, r'_{l_2-i+j})$, for all $j, 1 \leq j \leq i$. This inequality is equivalent to $\phi(r_{l_1+j}) - \phi(r_{l_1+j-1}) \geq \phi(r'_{l_2-i+j}) - \phi(r'_{l_2-i+j-1})$, which follows immediately using the relations $r_{l_1+j-1} \leq r'_{l_2-i+j-1}$, $r_{l_1+j} - r_{l_1+j-1} = r'_{l_2-i+j} - r'_{l_2-i+j-1}$ and the convexity of $\phi(\cdot)$. □

Remark 1. If the planar points $(l_1, \bar{W}(l_1))$ and $(l_2, \bar{W}(l_2))$, for l_1 and l_2 as in the above lemma, are on a convex hull edge of the curve $(\cdot, \bar{W}(\cdot))$, then the points $(l_1+i, \bar{W}(l_1+i))$ and $(l_2-i, \bar{W}(l_2-i))$

are on this convex hull edge as well, and the following equality easily follows

$$\bar{W}(l_1+i)+\bar{W}(l_2-i)=\bar{W}(l_1)+\bar{W}(l_2). \tag{29}$$

This implies that relation (27) is satisfied with equality and, moreover, the paths Q_1 and Q_2 are the (l_1+i)-edge, respectively, (l_2-i)-edge, maximum-weight paths in G.

We are now ready to prove Proposition 2.

Proposition 2. The inequality

$$2\bar{W}(l)\geq \bar{W}(l-1)+\bar{W}(l+1) \tag{30}$$

holds for all $l, 1\leq l\leq M$.

Proof. Let the paths P_1 and P_2 be the $(l-1)$-edge, respectively $(l+1)$-edge, maximum-weight paths in G. According to Lemma 2, there are two l-edge paths Q_1 and Q_2 such that the following relation to hold:

$$W(Q_1)+W(Q_2)\geq W(P_1)+W(P_2). \tag{31}$$

Since $2\bar{W}(l)\geq W(Q_1)+W(Q_2)$, Proposition 2 immediately follows. □

REFERENCES

[1] A. Aggarwal, B. Schieber, and T. Tokuyama, "Finding a minimum-weight k-link path in graphs with the concave monge property and applications", *Discrete & Computational Geometry* , vol. 12, pp. 263-280, 1994.

[2] A. Aggarwal, M. Klave, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm", *Algorithmica*, 2(1987), pp. 195-208.

[3] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams", *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp.230-239, April 2004.

[4] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources", *Operations Res.*, 11, pp. 399-417, 1963.

[5] J. E. Fowler, "QccPack: an open source softaware library for quantization, compression, and coding" in *Applications of Digital Image Processing XXIII*, vol. 4115, A.G. Tescher, Ed., San Diego, CA, Aug. 2000, pp. 294-301.

[6] T. Gan, and K.-K. Ma, "Weighted unequal error protection for transmitting scalable object-oriented images over packet-erasure networks", *IEEE Trans. Image Proc.*, vol. 14, no. 2, pp. 189-199, feb. 2005.

[7] T. Ibaraki, and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*, Cambridge, MA: MIT Press, 1988.

[8] D. G. Luenberger, *Optimization by vector space methods*, John Wiley & Sons, 1969.

[9] A. E. Mohr, E.A. Riskin, and R. E. Ladner, "Graceful degradation over packet erasure channels through forward error correction", in *Proc. DCC'99* , pp. 92-101, March 1999.

[10] A. E. Mohr, R. E. Ladner, and E.A. Riskin, " Approximately optimal assignment for unequal loss protection", in *Proc. ICIP 2000 International Conf. on Image Processing*, vol. 1, pp. 367-370, Sept. 2000.

[11] R. Puri, and K. Ramchandran, ”Multiple description source coding through forward error correction codes”, *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 342-346, Oct. 1999.

[12] A. Said and W. A. Pearlman, ”A new, fast, and efficient image codec based on set partitioning in hierarchical trees”, *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, pp. 243-250, June 1996.

[13] V. Stankovic, R. Hamzaoui, and Z. Xiong, ”Efficient channel code rate selection algorithms for forward error correction of packetized multimedia bitstreams in varying channels”, *IEEE Trans. Multimedia, Special Issue on Streaming Media*, vol. 6, no.2, pp. 240-248, April 2004.

[14] T. Stockhammer, C. Buchner, ”Progressive texture video streaming for lossy packet networks”, *Proc. 11th International Packet Video Workshop*, Kyongju, May 2001.

[15] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Processing*, vol. 9, pp. 1158-1170, July 2000.

[16] R. Wilber, ”The concave least-weight subsequence problem revisited”, *J. of Algorithms*, 9(1988), pp. 418-425.