Chapter 4

Optimal Bit Allocation for Dependent Quantization [1]

Contents

4.1 Introduction

In this chapter we study the problem of *dependent* bit allocation. All the work addressed in the literature so far has been confined to coding environments where the input signal units (e.g. image blocks or subbands) have been coded *independently*. However, many popular schemes (e.g. DPCM) involve *dependent* coding frameworks, i.e. where the set of available R–D operating points for some coding units (Fig.

[1] Part of this work has been done jointly with Kannan Ramchandran. For related publications see [85, 72, 84, 86]

4-1(b)) depends on the particular choice of R–D point for other coding units (Fig. 4-1(a)). Other dependent coding examples include multiresolution (MR) coders like the Laplacian spatial pyramid [11] and the closed–loop spatio-temporal pyramid video coder [108], as well as the MPEG [58] coder. Refer to Fig. 4-2 for an overview of where dependent coding schemes fit into typical source coding environments.

In this chapter [85, 84, 86], we generalize the bit allocation problem to include *temporally dependent* coding blocks (see [85, 86] for the case when dependent allocation in multiresolution scheme is considered). As in [101], we make no assumptions about the input or quantizer characteristics, and deal with arbitrary input signals and arbitrary discrete admissible quantizer sets. An alternative, model-based, solution of the dependent bit allocation problem has been proposed in [107].

As seen from Fig. 4-1 for the two–frame case, the number of possible operating points for each frame grows exponentially in the dependency tree depth, making the problem very difficult in the general case (e.g. DPCM). However, when the dependency tree is structured, as in MPEG, with several "leaves" or "terminal nodes" (e.g. B-frames of MPEG, as will be seen in Section 4.3), then it is possible to solve the difficult dependent problem elegantly, by describing how to formulate intelligent pruning rules to eliminate suboptimal operating points. Moreover, a number of important applications (like CD-ROM storage) can afford considerable off-line complexity, thus making our proposed approach relevant.

This chapter is organized as follows: in Section 4.2, we provide a Lagrangian–based solution for an arbitrary set of quantizers for each coding block. We point out the complexity of this approach, and introduce a certain monotonicity property of the operational R–D curves of the signal blocks to help reduce the complexity of the search for the optimal solution. We address the temporally dependent coding scheme in detail in Section 4.3 using MPEG as an example, and show how the monotonicity

property introduced earlier can be used to formulate fast heuristic solutions. In Section 4.4, we briefly outline how these techniques could be extended to include buffer constraints as well.

4.2 Dependent coding problem formulation

In this section, we define a general dependent allocation problem, show how this general formulation is applicable to MPEG and multiresolution coders, and give a solution based on Lagrange multipliers. Before we introduce the dependent coding problem, let us review the optimal independent allocation case which has been studied in the literature [50, 101, 113] and which was used in Chapter 2. The formulation is reviewed here for convenience and also to introduce a notation that is more adequate for the specific problems discussed in this chapter.

4.2.1 Optimal independent allocation – the constant slope condition

The classical rate–distortion optimal bit allocation problem consists of minimizing the average distortion D of a collection of signal elements or blocks subject to a total bit rate constraint R_{budget} for all blocks. For the two-block case, where $\{Q_1, R_1(Q_1), D_1(Q_1)\}$ and $\{Q_2, R_2(Q_2), D_2(Q_2)\}$ refer to the quantizer, distortion and bit rate of each block respectively, the independent allocation problem is:

$$\min_{Q_1,Q_2} \left[D_1(Q_1) + D_2(Q_2)\right] \tag{4.1}$$

$$\text{such that } R_1(Q_1) + R_2(Q_2) \leq R_{budget}. \tag{4.2}$$

The "hard" constrained optimization problem of (4.1),(4.2) can be solved by being converted to an "easy" equivalent unconstrained problem. This is done by "merging"

rate and distortion through the Lagrange multiplier $\lambda \geq 0$ [101], and finding the minimum Lagrangian cost $J_i(\lambda) = \min_{Q_i}[D_i(Q_i) + \lambda R_i(Q_i)]$ for $i = 1, 2$. The search for the optimal R–D operating points for each signal block can be done *independently*, for the fixed quality "slope" λ (which trades off distortion for rate) because at R–D optimality, all blocks *must operate at a constant slope point λ on their R–D curves* [101, 87]. The desired optimal constant slope value λ^* is not known *a priori* and depends on the particular target budget or quality constraint, but can be obtained via a fast convex search [87]. See Section 2.4.2.1 for more details.

4.2.2 General formulation

In the more general case, signal units may not be independently coded. Without loss of generality, we first consider a 2-level dependency as in Fig. 4-1. Shown are the R–D characteristics for a given discrete set of quantization choices for the first independent frame $(R_1(Q_1), D_1(Q_1))$ and the second dependent frame $(R_2(Q_1, Q_2), D_2(Q_1, Q_2))$. Our constrained optimization problem is: what quantization choice do we use for each frame such that the total (or average) distortion is minimized subject to a maximum total bit budget constraint? We model the total distortion as a weighted average of the individual distortions D_1 and D_2 in our general formulation. We will show how different choices of the weights lead to different problems of interest. Our problem can be formulated as:

$$\min_{Q_1, Q_2} \; [w_1 D_1(Q_1) + w_2 D_2(Q_1, Q_2)] \tag{4.3}$$

$$\text{such that } R_1(Q_1) + R_2(Q_1, Q_2) \leq R_{budget}. \tag{4.4}$$

Note that although Q_1 and Q_2 here represent frame–level quantization choices, it does not imply that all blocks within the frame have the same quantization scale.

Thus Q_1 could consist of a vector choice of different quantization scales for each block of frame 1. Also note that for arbitrary choices of w_1 and w_2, we have a weighted mean-squared-error (MSE) criterion, which reduces to the conventional (unweighted) MSE measure when $w_1 = w_2 = 1$.

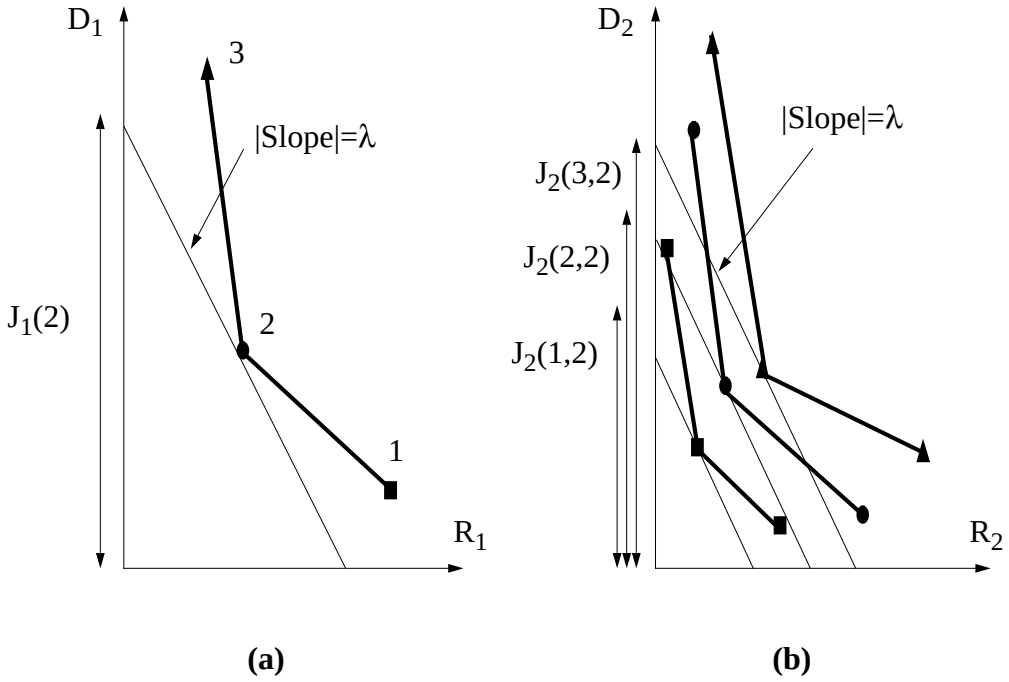

Figure 4-1: Operational R-D characteristics of 2 frames in a dependent coding framework, where frame 2 depends on frame 1. (a) Independent frame's R-D curve. (b) Dependent frame's R-D curves. Note how each quantizer choice for frame 1 leads to a different $R_2 - D_2$ curve. The Lagrangian costs shown are $J = D + \lambda R$ for each frame.

4.2.3 Examples

We now provide examples of problems of interest which follow as special cases of (4.3),(4.4). See Fig. 4-2.

Figure 4-2: Overview of typical source coding environments.

Example 1. Independent coding: The independent case seen in Section 4.2.1 is a special case of (4.3),(4.4), where frame 2 does not depend on frame 1, i.e. $R_2(Q_1, Q_2) = R_2(Q_2)$ and $D_2(Q_1, Q_2) = D_2(Q_2)$. The independent coding case arises for intraframe coding as well as pyramidal coding without quantization feedback [11, 85, 86]. This was also the situation considered throughout Chapter 2.

Example 2. Spatially dependent pyramid coding with quantization feedback: When $w_1 = 0, w_2 = 1$, we have the case of a two-layer closed loop (quantization feedback) pyramid, where the bottom layer (layer 2) depends on the quantization choice of the top layer (layer 1) [85, 86]. Note that (4.4) refers to a total bit rate constraint, and we solve the full–resolution quality optimization problem only. In

addition, for a multiresolution coding environment, it may be necessary to throw in additional constraints on the top layer bit rate [11, 85, 86]:

$$R_1(Q_1) \leq R_1', \tag{4.5}$$

or quality

$$D_1(Q_1) \leq D_1'. \tag{4.6}$$

Example 3. Temporally dependent coding: This is the most general case where (4.3),(4.4) apply without any restrictions. DPCM and MPEG come under this class.

4.2.4 Solution based on Lagrange multipliers

The problem of (4.3),(4.4) can be solved by introducing the Lagrangian cost J corresponding to the Lagrange multiplier $\lambda \geq 0$ as in [101] as follows:

$$J_1(Q_1) = w_1 D_1(Q_1) + \lambda\ R_1(Q_1), \tag{4.7}$$

$$J_2(Q_1, Q_2) = w_2 D_2(Q_1, Q_2) + \lambda\ R_2(Q_1, Q_2), \tag{4.8}$$

and considering the following unconstrained minimization problem:

$$\min_{Q_1, Q_2} [J_1(Q_1) + J_2(Q_1, Q_2)]. \tag{4.9}$$

Then, by a direct extension of Theorem 1 of Shoham and Gersho in [101], the following result follows:

Theorem 4.1 *If (Q_1^*, Q_2^*) solves the unconstrained problem of Eq. (4.9), then it also solves the constrained problem of Eqs. (4.3), (4.4) for the particular case of $R_{budget} = [R_1(Q_1^*) + R_2(Q_1^*, Q_2^*)]$.*

Proof: See Appendix 4.5.

The above result implies that if we solve the unconstrained problem of (4.9) for a fixed value of λ, and if the total bit rate happens to be R_{budget}, then we have also optimally solved the constrained optimization problem of (4.3), (4.4). Further, as λ is swept from 0 to ∞, one traces out the convex hull of the composite R–D curve of the dependent allocation problem. The monotonic relationship between λ and the expended bit rate [101] makes it easy to search for the "correct" value of λ, say λ^*, for a desired R_{budget}.

Note how for the independent case ($J_2(Q_1, Q_2) = J_2(Q_2)$), where there is a single $R_2 - D_2$ curve in Fig. 4-1, (4.9) becomes the familiar result of [101] (See also Chapter 2). Here, each frame is minimized independently, as was shown in Section 4.2.1.

For the general dependent case, the 2-frame problem becomes the search for Q_1^*, Q_2^* that solve:

$$J_1(Q_1^*) + J_2(Q_1^*, Q_2^*) = \min_{Q_1, Q_2} [J_1(Q_1) + J_2(Q_1, Q_2)] \quad (4.10)$$

$$= \min_{Q_1} [J_1(Q_1) + J_2(Q_1, Q_2^*(Q_1))], \quad (4.11)$$

where $J_2(Q_1, Q_2^*(Q_1)) = \min_{Q_2}[w_2 D_2(Q_1, Q_2) + \lambda R_2(Q_1, Q_2)]$ is the minimum Lagrangian cost (for quality condition λ) associated with the dependent frame when the independent frame is quantized with Q_1. See Fig. 4-1. Thus, for the desired operating quality λ, we find the optimal solution by finding, for all choices of Q_1 for the independent layer, the optimal $(Q_2^*(Q_1))$ which "lives" at absolute slope λ on the (dependent) R_2–D_2 curve associated with Q_1.

The important point to note is that here we cannot treat each of the blocks separately as in Section 2.4.2.1 and thus minimizing (4.9) actually requires, if no further assumptions are made, an *exhaustive search* for the minimal lagrangian for

all possible combinations of Q_1, Q_2. Fig. 4-3 shows an example where a greedy choice of quantizer for the first frame, i.e. choosing the quantizer that minimizes $J_1(Q_1)$ would have resulted in overall suboptimality. In general we will have that:

$$\min_{Q_1,Q_2}[J_1(Q_1) + J_2(Q_1, Q_2)] \neq \min_{Q_1}[J_1(Q_1)] + \min_{Q_2}[J_2(Q_1^*, Q_2)]$$

where $Q_1^* = \arg\min_{Q_1}[J_1(Q_1)]$.

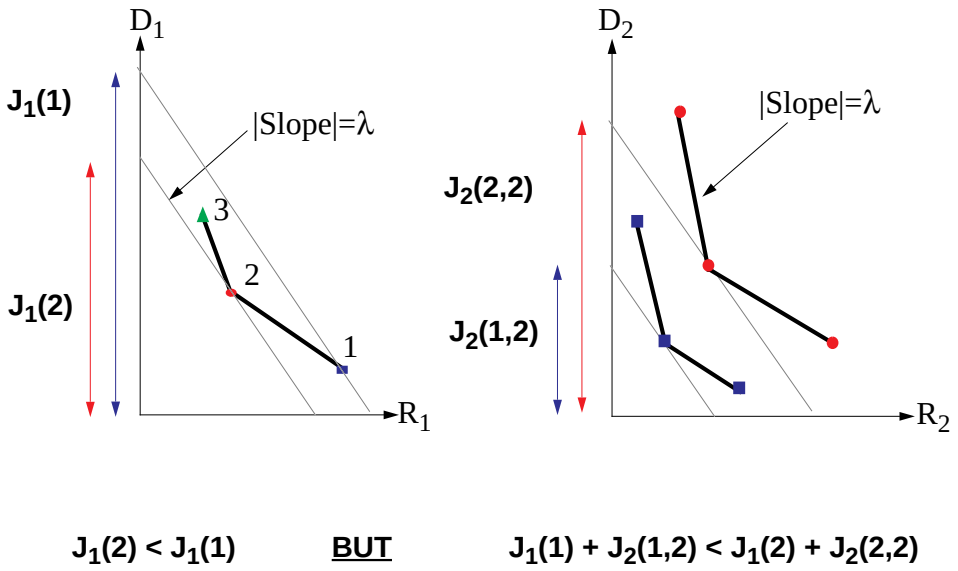

Figure 4-3: Example of search for the minimal λ. Choosing quantizer 2 would be optimal if only the first frame were considered. However choosing quantizer 1 for frame 1 is better overall because the gain for frame 2 compensates the suboptimality for frame 1.

By a simple extension of Theorem 4.1, it follows that the optimal solution to our general N–frame dependency problem consists in introducing $J_i(Q_1, Q_2, \ldots, Q_i) = w_i D_i(Q_1, Q_2, \ldots, Q_i) + \lambda R_i(Q_1, Q_2, \ldots, Q_i)$ for $i = 1, 2, \ldots, N$ and solving the fol-

lowing unconstrained problem for the "correct" value of λ which meets the given R_{budget}:

$$\min_{Q_1,Q_2,\ldots,Q_N} [J_1(Q_1) + J_2(Q_1,Q_2) + \ldots J_N(Q_1,Q_2,\ldots,Q_N)]. \tag{4.12}$$

4.2.5 Complexity

The optimal solution, as shown by (4.12) is obviously exponentially complex in the dependency-tree depth N. Moreover, it has to be pointed out that the computational complexity is dominated by the *data generation* phase, i.e. finding all the $(R_i(Q_1,Q_2,\ldots,Q_i), D_i(Q_1,Q_2,\ldots,Q_i))$ points for the problem is much more complex than finding the optimal solution, given all the possible R–D operating points. The complexity comes from the fact that, contrary to the situation in the independent case, the R–D characteristic for a given block is a function of choices of quantizers for previous blocks and thus an exponentially growing (in the depth of the dependency) number of possible solutions has to be considered.

In order to ease this computational burden, we are therefore interested in methods which will avoid the need to grow all the R–D data, while retaining optimality. We now examine an important property which enables us to do this, and which will be used in Section 4.3 to formulate pruning conditions to eliminate suboptimal choices in the MPEG allocation problem. Note that while our methods rely on generating "real" R–D data, a model-based approach may be used within our framework to reduce the complexity. For instance, one could measure some of the R-D points and the extrapolate the R-D values for the others.

4.2.6 Monotonicity

The key to obtaining a fast solution to the complex dependent allocation problem of (4.3),(4.4) is the monotonicity property of the R–D curves of the dependent components (frames). Consider the example of 2 frames, with the operational R–D curve of the second frame depending on that of the first, as in Fig. 4-1. Assume that the quantizer grades are ordered as monotonically increasing from finest to coarsest. Let us use i and j to denote the quantization choices for the independent and dependent frames, respectively. Thus, using our convention, $i < i'$ denotes that quantizer i is finer than quantizer i'.

Definition 4.1 *A dependent coding system has the monotonicity property if, for any* $\lambda \geq 0$*:*

$$J_2(i,j) \leq J_2(i',j), \qquad \text{for i} \leq \text{i}'. \tag{4.13}$$

For example, for $\lambda = 0$, this means that

$$D_2(i,j) \leq D_2(i',j), \qquad \text{for i} \leq \text{i}'. \tag{4.14}$$

Stated in words, the monotonicity condition simply implies that a "better" (i.e. finer quantized) predictor will lead to more efficient coding, in the rate-distortion sense, of the residue (whose energy decreases as the predictor quality gets better). That is, the dependent frame's family of R-D curves will be monotonic in the fineness of the quantizer choice associated with the parent frame from which they are derived. As can be seen, the finer the quantization for frame 1 (Fig. 4-1(a)), the closer to the origin of the $R_2 - D_2$ graph will be the corresponding curve for the dependent frame 2 (Fig. 4-1(b)). Experimental results involving MPEG *verify this monotonicity property* for all the cases that we studied. Thus, monotonicity appears to be a realistic

property, which has favorable theoretical implications as well, and it can be used to formulate fast pruning conditions for the MPEG allocation problem in Section 4.3.

4.3 Temporal dependency: the MPEG case

We now address the general temporal dependency quantization problem of which MPEG [58] is an example. The MPEG coding format, a CCITT video compression standard, shown in Fig. 4-4, segments the video sequence into groups-of-pictures (GOP's). Each GOP consists of three types of frames using the intraframe (I), prediction (P) and bidirectionally–interpolated (B) modes of operation. The I frames are coded independently, the P frames are predicted from the previous I or P, and the B frames are interpolated from the previous and next I and/or P frames. For the 2-frame dependency illustrated in Fig. 4-1, the problem we are trying to solve (for an MSE criterion) is the problem of (4.3),(4.4) with $w_1 = w_2 = 1$:

$$\min_{Q_1,Q_2} \left[D_1(Q_1) + D_2(Q_1,Q_2)\right] \text{ s.t. } R_1(Q_1) + R_2(Q_1,Q_2) \leq R_{budget}.$$

The solution to this problem was shown in Section 4.2.2 as being exponentially complex in the dependency tree depth. Here, we will show how to reduce this complexity for the MPEG coding case (see Fig. 4-4). Before we tackle the general MPEG problem (with I,P and B frames), we begin with a simpler special case of MPEG that is easier to analyze and which provides the intuition for the more complex general problem.

4.3.1 A particular case of MPEG: I-B-I

We consider a special case of MPEG having only I and B frames (see Fig. 4-5), i.e. the predicted P frames of the more general MPEG format are omitted. The

Figure 4-4: Typical MPEG coding framework. (a) The MPEG frames: the I frames are independently coded, the P frames are predicted from previous I or P frames, and the B frames are interpolated from adjacent I and/or P frame pairs. (b) Temporal dependency in the MPEG framework. Note that the B frames are leaves in the dependency tree, since no prediction is generated using them.

dependency tree is shown in the more compact form of a trellis. The "states" of the trellis represent the quantization choices for the *independently coded* I frames (ordered from top to bottom in the direction of finest to coarsest), while the "branches" denote the quantizer choices associated with the two B frames.

The trellis is populated with Lagrangian costs (for a fixed λ) associated with the quantizers for each frame. Let us focus on the $I_1 - B_1 - B_2 - I_2$ stage of the trellis. The state nodes are populated with the costs of the respective I frame quantizers $J(Q) = (D(Q) + \lambda R(Q))$. Each (i, j) branch connecting quantizer state i of I_1 to quantizer state j of I_2 is populated with the sum of the minimum Lagrangian costs

Figure 4-5: The I-B-I special case of MPEG. Finding an R-D convex hull point corresponding to a λ is equivalent to finding the smallest cost path through the trellis. Each trellis node corresponds to a quantizer choice for the I frames, monotonically ordered from finest to coarsest, and is populated with the associated Lagrangian cost ($J(I) = D(q) + \lambda R(q)$). The branches correspond to the B frame pairs, and are populated with their minimum Lagrangian costs ($J(B) = \min[D(q) + \lambda R(q)]$) for the particular I frame quantizer choices given by each branch's end nodes. For quality slope λ, the optimal total cost path is obtained with the Viterbi algorithm. The "dark line" path joins the smallest cost I frame nodes. Monotonicity implies that all dashed line paths can be pruned out.

of the B_1 and B_2 frames, i.e. with $J(B_1) + J(B_2)$, where:

$$J(B_l) = \min_{Q_{B_l}}[D(Q_{B_l}) + \lambda R(Q_{B_l})] \text{ for } l = 1, 2 \tag{4.15}$$

where the R-D curves for B_1, B_2, are generated from the i, j quantizer choices for I_1, I_2 respectively. From (4.12), it is clear that the optimal path is that which has the minimum total cost across all trellis paths. Since the independent I frames "decouple" the B frame pairs from one another, it is obvious that the popular Viterbi

algorithm (VA) [34] will provide the minimum cost path through the trellis, i.e. we need to keep a single path (the minimum cost one) arriving at each node. More formally, we have the following algorithm for the $I_1 - B_1 - B_2 - I_2$ stage of the trellis:

Algorithm 4.1

Step 1 *: Generate $J(I_1)$ and $J(I_2)$ for I-frames I_1 and I_2 for all quantizers in Q_{I_1}, Q_{I_2} respectively, i.e. populate all the* **nodes** *of the trellis with the Lagrangian costs.*

Step 2 *: For every pair of nodes (i, j), where i and j are the quantizer choices for I_1 and I_2 respectively, assign branch cost $J(B_1) + J(B_2)$ where $J(B_l)$ for $l = 1, 2$ are obtained from (4.15), i.e. populate all the* **branches** *of the trellis with the minimum Lagrangian costs.*

Step 3 *(***VA pruning rule***): At every node of I_2, keep only that branch which minimizes $J(I_1) + J(B_1) + J(B_2)$.*

As noted in Section 4.2.5, the computational complexity is dominated by the *data generation* phase, i.e. in the trellis population phase.

4.3.2 Pruning conditions implied by monotonicity

The monotonicity condition stated earlier in Section 4.2.6 will now be used to formulate pruning conditions to eliminate suboptimal operating points in the temporal dependency coding problem. The first lemma is associated with Fig. 4-6(a). As a reminder, the quantizer states are ranked in a monotonically increasing order from finest to coarsest.

Lemma 4.1 *If*

$$J_1(i) + J_2(i, j) < J_1(i') + J_2(i', j) \quad \textit{for } \textbf{any } i < i', \tag{4.16}$$

Figure 4-6: Pruning conditions obtained from monotonicity. (a) $J_1(i_2) + J_2(i_2, j)$ is the minimum Lagrangian cost of all branches terminating in node j. Therefore (see Lemma 1), the (i_3, j) branch can be pruned. (b) $J_2(i, j_1))$ is the minimum Lagrangian cost of all branches originating from node i. Therefore (see Lemma 2), the (i, j_2) and the (i, j_3) branches can be pruned. (c) Diagram used for the proof of Lemma 1.

then the (i', j) branch cannot be part of the optimal path and can be pruned out.

Proof: We prove the lemma by contradiction. Assume that (i', j) for any $i < i'$ **is** part of the optimal path (see Fig. 4-6(c)) and let the optimal quantizer sequence path be $(i', j, k, \ldots, l)$. But, by monotonicity, we have:

$$J_3(i, j, k) \leq J_3(i', j, k) \tag{4.17}$$

$$\ldots$$

$$J_L(i, j, k, \ldots, l) \leq J_L(i', j, k, \ldots, l) \tag{4.18}$$

Summing up Eqs. (4.16), (4.17), ..., (4.18), we get the contradiction that the total Lagrangian cost of the path $(i, j, k, \ldots, l)$ is smaller than that of the optimal path $(i', j, k, \ldots, l)$. □

The above lemma is associated with pruning branches that merge into a common destination state. A dual result holds for the pruning of branches that originate from a common source state (see Fig. 4-6(b)) leading to the following companion Lemma 2, whose proof is omitted as it is similar to that of Lemma 1:

Lemma 4.2 *If $J_2(i, j) < J_2(i, j')$ for any $j < j'$, then the (i, j') branch cannot be part of the optimal path and can be pruned out.*

Note that a consequence of the above lemma is that if $J_1(i) < J_1(i')$ for $i < i'$, then the state node i' (and all branches from it) can be pruned out. The two pruning conditions of Lemmas 1 and 2 can be used to lower the complexity of the VA-based search. In the special case of MPEG of Section 4.3.1 (refer to Fig. 4-5), Lemmas 1 and 2 eliminate the need to consider the full trellis on which to run the VA, making

Figure 4-7: General MPEG "trellis" diagram extension of Fig. 3. Here, the inclusion of the P frames prevents the decoupling of the B frame pairs, and the entire tree has to be grown. Note that each stage of the trellis is represented by "vector" branches whose dimension grows exponentially with the dependency tree depth.

4.3.3 General MPEG bit allocation

Having established the intuition behind dependent allocation and the power of monotonicity, we now evolve to the more complex (general) MPEG format of Fig. 4-7. The presence of the P frames extends the dependency tree depth, and the decoupling between successive stages of the trellis is lost. We can thus no longer resort to the

Viterbi algorithm, but must instead retain the entire tree, which grows exponentially with the number of dependent levels (although only until a new I frame is transmitted, so that the complexity may still be manageable.) The good news, however, is that the monotonicity conditions still apply, and the pruning conditions of Lemmas 1 and 2 can aid in reducing complexity dramatically. As an example, see Fig. 4-8 where we consider an I-B-P-B-P sequence of MPEG frames (note that for simplicity, we use only one B-frame between I-P pairs) and a choice of 3 quantizer grades for each frame. More formally, the algorithm used is the following (refer to Figs. 4-7 and 4-8):

Algorithm 4.2

Step 1 : *Generate $J(I)$ for all quantizers $q \in Q_I$. See Fig. 4-8(a).*

Step 2 : *(Monotonicity) Prune out all I-nodes lying below minimum cost node $q^* \in Q_I$ in Step 1.*

Step 3 : *Grow $J(I, P_1)$ for all combinations of $q \in Q_{P_1}$ and all remaining $q \in Q_I$ after Step 2. See Fig. 4-8(b).*

Step 4 : *(Monotonicity) Use pruning conditions of Lemmas 1 and 2 to eliminate suboptimal $I - P_1$ combinations. See Fig. 4-8(b).*

Step 5 : *For every surviving $I - P_1$ combination, find the B_1, B_2 quantizer pair that minimizes $J(B_1) + J(B_2)$, i.e. populate the branch costs of the trellis of Fig. 4-7. See Fig. 4-8(c).*

Step 6 : *(Monotonicity) Use pruning conditions of Lemmas 1 and 2 to eliminate suboptimal $I - B_1 - B_2 - P_1$ combinations. See Fig. 4-8(c).*

Step 7 : *For all remaining paths, repeat Steps 3 to 6 for the $(P_1 - B_3 - B_4 - P_2)$ and the $(P_2 - B_5 - B_6 - I)$ sets.*

The smallest cost path after running Algorithm 4.2 is the optimal solution corresponding to the chosen λ for the group-of-pictures considered. While the exhaustive search would have us grow as many as 363 Lagrangian costs, in our example, only 36 costs need to be grown, an order of magnitude reduction in complexity with no loss of optimality if the monotonicity conditions apply (in our example, application of Algorithm 4.2 for $\lambda = 10$ provides an optimal R–D operating point – 40.76 dB at 1 bpp – as was verified through exhaustive search). The complexity reduction due to monotonicity is dependent on the desired quality slope λ, with higher quality targets achieving better reduction. In the limit, as λ goes to 0, the minimum cost path is always the one corresponding to the finest quantizers and thus only a single "highest quality" path has to be grown. Conversely, if λ goes to ∞ the monotonicity property provides no gain. Thus one should not resort to the convex search techniques described in Section 2.4.2.1 and would have to use instead search strategies which start with smaller values for λ and increase it until a solution is reached.

4.3.3.1 Suboptimal heuristics

As pointed out, the amount to which the monotonicity property can be exploited is λ-dependent, and may not suffice for some applications. To this end, it is advisable to come up with fast heuristics, which, used in combination with monotonicity, can approach the optimal performance at a fraction of the complexity. In trying to formulate a fast MPEG heuristic, it is necessary to consider some important points: (i) the "anchor" I-frame is the most important of the group of pictures and must not be compromised, (ii) most signal sequences enjoy a finite memory property, where the influence of a parent frame diminishes with the level of its dependency.

Thus, it may pay to choose (only) the lowest cost nodes for all frames except the I-frame, for which we retain all nodes remaining after applying monotonicity-based

pruning. Thus, a single path is grown from each of these admissible I-frame nodes, whereas in the general case, a whole tree evolves from each such node. Based on this, we propose the following heuristic: (i) retain all paths that *originate from* each of the I-frame quantization states remaining after the monotonicity-based pruning, i.e. it is not prudent to be greedy for the I-frame, as a greedy error affects all dependent frames derived from it; (ii) use a "greedy" pruning condition (in combination with the monotonicity property) to keep only the lowest cost branch thus far at all other stages in the trellis. That is, we follow Algorithm 4.2, except that we add an extra pruning condition in Steps 4 and 6, where we retain only a single (minimum cost) path corresponding to every surviving I-frame node. This heuristic, as shown in Fig. 4-9, leads to near optimal performance at a fraction of the computational cost. For the example of Figs. 4-8 and 4-9, the optimal solution for a particular λ gives 40.76 dB at 1 bpp, while the heuristic achieves 40.45 dB at 0.97 bpp, certainly very close to optimality. Unlike Algorithm 4.2, which relies on monotonicity pruning conditions only and which works best when λ is low (high quality), the fast heuristic retains low complexity even at high values of λ. Note how choosing the greedy solution for the I-frame (bottom-most node) would have given a much worse performance.

4.3.3.2 Discussion

We have shown a method to find the optimal bit allocation strategy for an MPEG coding framework, assuming arbitrary quantizer sets for each MPEG frame. Although our scheme can be computationally complex, it can serve as an optimal benchmark to evaluate more practical allocation strategies. Also, model-based approaches (e.g. measuring one R–D point and using a model-based extrapolation to find other points) can be combined with our techniques to ease the computational burden. Note that since MPEG coders are typically buffered with a buffer size of the order of a group-

Figure 4-8: Tree pruning using the monotonicity property (Lemmas 1,2). The numbers are the cumulative Lagrangian costs for a typical example for $\lambda = 10$. Branches pruned at each stage are shown with dashed lines. In this example, the number of R–D points generated is cut down from 363 (exhaustive) to only 36 with no loss of optimality.

4.4 Buffer constrained allocation for dependent coders

In Chapter 2 we dealt with the problem of allocating bits among different signal blocks when the sequence of blocks had to be transmitted with finite end-to-end delay, i.e. we had an allocation problem with the constraint that the encoder buffer did not overflow. In Chapter 2, however, we had assumed that the blocks to be quantized were coded independently (in an MPEG environment this would mean

Figure 4-9: Tree pruning using monotonicity as well as a "greedy" heuristic for the same conditions as those of Fig. 6. The number of R-D points generated is now 24, at a slight loss of optimality (total Lagrangian cost is 77.91 versus optimal cost of 77.24).

only I frames are used) so that our solution there would not be applicable to the dependent environments discussed in this chapter. Here we outline how the solution of Section 4.2.4 can be modified to account for the buffer constraint.

Given the rate and distortion (R and D, resp.) for each frame and calling $B(i)$ the buffer occupancy after coding frame i, our buffer control problem is that of finding the quantizer choice $(Q_1, \ldots, Q_N)$ to:

$$\min(\sum_{i=1}^{N} D_i(Q_1, \ldots, Q_i)) \quad s.t. \quad (i) \quad \sum_{i=1}^{N} R_i(Q_1, \ldots, Q_i) \leq N \cdot r = R_{budget} \tag{4.19}$$

$$(ii) \quad B(i) \leq B_{max}, \quad \forall i. \tag{4.20}$$

Obviously, without the buffer constraint of (ii), our problem reduces to the dependent bit allocation studied in Section 4.2.4. There, the minimization of the corresponding Lagrangian cost $\sum J_i(Q_1, \ldots, Q_i)$ for each block was formulated as the

search for the minimum Lagrangian cost path in a trellis, where the states represent the possible quantizers for each frame (See Fig. 4-7). The search can be sped up by resorting to heuristics based on the monotonicity property to prune out suboptimal branches. When constraint (ii) is considered, not all solutions obtained using the algorithms of Section 4.2.4 will be feasible due to overflow. However, if we add to the trellis pruning rules a new rule which eliminates paths that overflow, *the resulting algorithm will provide the optimal solution to (4.19) and (4.20)*, within a convex hull approximation. More formally, we grow all the solutions as in Section 4.2.4 where each branch, associated with a given quantizer selection, $(Q_1, \ldots, Q_i)$, will have associated a Lagrangian cost $\sum J_i(Q_1, \ldots, Q_i)$ *and* a buffer state,

$$B(i) = max(B(i-1) + R_i(Q_1, \ldots, Q_i) - r, 0).$$

Thus, on top of the pruning rules described in Section 4.3.2, we will prune out any branch such that $B(i) > (B_{max} - r)$.

The one parameter that remains to be set is the initial buffer state. Typically, since regardless of the coding environment the first frame in the sequence will be coded in *Intra* mode (and thus will very likely require a number of bits above the channel rate), we can assume that $B(0) = 0$ so that the buffer is initially empty. If we focus specifically in an MPEG environment, we can consider a group of pictures (GOP) and compute the allocation so that the initial *and* final buffer states are both zero. In this manner we can decouple the buffer constrained allocation for consecutive GOPs.

The above described method requires, in order to solve the problem, that a λ such that $\sum R_i(Q_1^*, Q_2^*, \ldots, Q_i^*) = N \cdot r$ be found. The pruning rules based on monotonicity are more efficient for small values of λ (i.e. high bit rate) and, similarly, smaller λ's will result in more overflowing paths. Thus it is reasonable to initialize λ to a "small"

value and then, to find the optimal solution, increase λ until the target rate is met. Solutions that were discarded because of overflow for a given λ do not have to be reconsidered for a different λ.

This algorithm leaves also room for fast approximations based on heuristics. For example, assume that $Q_1 = q$ is fixed and consider the resulting R-D characteristic, i.e. the set of all points $(\sum R_i, \sum D_i)$ for which $Q_1 = q$. Then, if the point corresponding to $(q, Q_2, \ldots, Q_N)$ lies on the convex hull of the the R-D characteristic generated with $Q_1 = q$ we have found that typically $(q', Q_2, \ldots, Q_N)$ will *also* lie on the convex hull of the R-D characteristic obtained when setting $Q_1 = q'$. Thus after studying the R-D characteristic for a particular value $Q_1 = q$ one can extrapolate which combinations for $(Q_2, \ldots, Q_N)$ are likely to be suboptimal *for any* Q_1 and therefore reduce the search to those that are likely to be "good".

To summarize, in this section we have described some initial results for a deterministic rate control for MPEG encoders. The optimal solution can be found with slight modifications of the algorithm that solves the unconstrained problem. Further work should focus on exploring more ways of reducing the complexity of the search by exploiting the structure of the R-D data through the use of heuristics.

4.5 Appendix: Proof of Theorem 1

Note that in the proof, in the interest of notation brevity, we omit the explicit dependence of D_1, R_1, J_1 and D_2, R_2, J_2 on Q_1 and Q_2; i.e. we use $J_1 = J_1(Q_1), D_1 = D_1(Q_1), R_1 = R_1(Q_1)$ and $J_2 = J_2(Q_1, Q_2), D_2 = D_2(Q_1, Q_2), R_2 = R_2(Q_1, Q_2)$, and similarly for the optimal quantizers Q_1^* and Q_2^*. Then, for all Q_1, Q_2, we have from the unconstrained minimization:

$$J_1^* + J_2^* \leq J_1 + J_2. \tag{4.21}$$

i.e.,

$$D_1^* + \lambda R_1^* + D_2^* + \lambda R_2^* \leq D_1 + \lambda R_1 + D_2 + \lambda R_2. \tag{4.22}$$

or,

$$[D_1^* + D_2^*] - [D_1 + D_2] \leq \lambda([R_1 + R_2] - [R_1^* + R_2*]). \tag{4.23}$$

Since (4.23) holds for all admissible $\{Q_1, Q_2\}$, it certainly holds for the subset of $\{Q_1, Q_2\}$ for which $[R_1 + R_2] \leq R_{budget}$, where $R_{budget} = [R_1^* + R_2^*]$. Therefore, from (4.23), since $\lambda \geq 0$, we have that:

$$[D_1(Q_1^*) + D_2(Q_1^*, Q_2^*)] - [D_1(Q_1) + D_2(Q_1, Q_2)] \leq 0, \tag{4.24}$$

i.e. over all $\{Q_1, Q_2\}$ which meet the rate budget, (Q_1^*, Q_2^*) gives the minimum distortion. □