DYNAMIC PROGRAMMING ALGORITHM FOR RATE-DISTORTION OPTIMIZED MEDIA STREAMING

Martin Röder[1], *Jean Cardinal*[2], *Raouf Hamzaoui*[1]

[1]Department of Computer and Information Science
University of Konstanz, Germany
[2]Computer Science Department
Université Libre de Bruxelles, Belgium

ABSTRACT

We propose a dynamic programming algorithm for finding optimal transmission policies for a single packet in rate-distortion optimized media streaming. The algorithm relies on an optimality assumption holding in particular when both the forward and round trip times have exponential distributions. In the other cases, we use the assumption as a heuristic principle. Simulations show that for realistic channel models, the algorithm provides optimal solutions and can be significantly faster than the previous fastest exact algorithm. The proposed algorithm can be used as a preprocessing step for streaming mutually dependent packets.

1. INTRODUCTION

Since its introduction in the 1990s, video streaming has become one of the most popular multimedia services over the Internet [1]. In contrast to traditional video transmission systems, a video streaming system allows the receiver to play back the compressed video bitstream continuously, while parts of it are still being received. Chou and Miao [2] introduced a general framework for optimizing the rate-distortion performance of a streaming media system over a packet erasure channel. This rate-distortion based framework has been recognized as one of the most important recent developments in streaming video [3]. In a sender-driven scenario, each member of a group of interdependent packets can be sent to a client at many time opportunities. If a packet is received before a delivery deadline, the client sends an acknowledgment packet to the server, which need not retransmit the packet. Chou and Miao [2] showed that rate-distortion optimal transmission policies can be obtained by minimizing a Lagrangian. They first provided a dynamic programming algorithm to compute an optimal policy when a single packet has to be transmitted. For the transmission of a group of interdependent packets, they proposed a heuristic iterative descent method called sensitivity adaptation, which uses the above dynamic programming algorithm to optimize for one packet at a time, keeping the other packets fixed. Packet dependencies are modeled with a directed acyclic graph, which makes the system suitable to a wide range of source coders, including the new JVT video standard H.264/AVC. The framework of Chou and Miao was adapted for a wireless channel [4] and extended to include multiple arrival deadlines [5]. Chakareski and Girod [6] proposed a modification to the feedback sending mechanism. Instead of sending a separate acknowledgment packet for each received packet, the receiver periodically sends a single feedback packet that informs the server on which packets were received when the acknowledgment packet was sent. Using an additive distortion model as an approximation to the distortion model of [2], Chakareski, Apostolopoulos, and Girod [7] provide a fast method for the computation of a transmission strategy for a group of packets. The simplified model allows a very efficient optimization at the cost of quality loss.

In [8], we studied the computational complexity of rate-distortion optimized media streaming. We showed that computing an optimal transmission policy for a single packet is an NP-hard problem. We also provided a branch and bound algorithm for computing all optimal policies for a single packet. In this paper, we propose a faster dynamic programming algorithm. For realistic transmission conditions, this algorithm can be significantly faster than the branch and bound algorithm without quality loss. Though the algorithm does not provide optimal solutions in general, we prove that the solutions are optimal when both the channel forward and round trip times have exponential distributions.

2. NOTATIONS AND TERMINOLOGY

A media source (e.g., audio, image, video) is encoded and packetized into a finite set of *data units*. The interdependency between the data units is given by a directed acyclic graph (V, E), where the set of vertices V is the set of data units and the set of edges $E \subset V \times V$ is given by $(l', l) \in E$

if l' must be decoded so that l can also be decoded. For $(l', l) \in E$, we write $l' \prec l$ and say that l' is an *ancestor* of l. We also use the notation $l' \preceq l$ if $l' \prec l$ or $l' = l$. We denote by D_0 the distortion if no data unit is decoded, $B_l > 0$ the size in bytes of data unit l, and ΔD_l the amount by which the distortion is decreased if data unit l is decoded compared to the distortion if only the ancestors of l are decoded. A channel packet containing a data unit and sent at time s can be either lost with probability ϵ_F, independent of s, or received at time s', where the *forward trip time* $FTT = s' - s$ is a random variable with probability density function p_F. Similarly, when a packet is sent from the client to the server through the feedback channel, it is either lost with probability ϵ_B or received after a *backward trip time* BTT, which is a random variable with probability density function p_B. We also define the *round trip time* RTT as the sum $FTT + BTT$. For convenience, by setting $FTT = \infty$ when a data unit is lost, one can extend the random variable FTT to make it include packet loss [2]. In this situation, the cumulative distribution function of the extended random variable, $\overline{FTT}$, is

$$P\{\overline{FTT} \leq \tau\} = \int_0^\tau (1 - \epsilon_F) p_F(t) dt.$$

The same extension can be done for BTT, yielding an extended random variable, $\overline{BTT}$. Finally, $\overline{RTT}$ will denote the sum $\overline{FTT} + \overline{BTT}$.

In a sender-driven context, a data unit can be sent to the receiver at transmission opportunities $1, \ldots, n$ with respective times $s_1, \ldots, s_n,\ s_1 < \ldots < s_n < s_D$, where s_D is a delivery deadline. A transmission policy π of length n and size $|\pi|$ is a $|\pi|$-element subset of $\{1, \ldots, n\}$. For $i = 1, \ldots, n$, if $i \notin \pi$, then the data unit should not be sent at opportunity i, whereas $i \in \pi$ means that the data unit should be sent at opportunity i if no acknowledgment packet was received from the feedback channel before time s_i, and that the data unit should not be sent at this opportunity otherwise.

The error $\epsilon(\pi)$ for policy π is defined as the probability that the data unit does not reach its destination before time s_D, that is,

$$\epsilon(\pi) = \prod_{i \in \pi} P\{\overline{FTT} > s_D - s_i\}. \tag{1}$$

The cost $\rho(\pi)$ for policy π is defined as the expected number of data unit transmissions. One can show that

$$\rho(\pi) = \sum_{i \in \pi} \left(\prod_{j < i, j \in \pi} P\{\overline{RTT} > s_i - s_j\} \right). \tag{2}$$

We call a policy π^* *optimal* if there exists no policy π of the same length such that $\epsilon(\pi) \leq \epsilon(\pi^*)$ and $\rho(\pi) < \rho(\pi^*)$. Finding optimal policies is an NP-hard problem [8].

The transmission policies for a group of interdependent data units is described by a policy vector $\vec{\pi} = (\pi_1, \ldots, \pi_L)$, where $\pi_i,\ i = 1, \ldots, L$, is the transmission policy for the ith data unit of the group. The expected rate for $\vec{\pi}$ is $R(\vec{\pi}) = \sum_{l=1}^{L} B_l \rho(\pi_l)$. By assuming independence of the transmission processes, the expected distortion for $\vec{\pi}$ is

$$D(\vec{\pi}) = D_0 - \sum_{l=1}^{L} \Delta D_l \prod_{l' \preceq l} (1 - \epsilon(\pi_{l'})).$$

A policy vector $\vec{\pi}^*$ is *optimal* if there exists no policy vector $\vec{\pi}$ such that $D(\vec{\pi}) \leq D(\vec{\pi}^*)$ and $R(\vec{\pi}) < R(\vec{\pi}^*)$.

3. DYNAMIC PROGRAMMING

To determine all optimal policies, we propose a new dynamic programming algorithm as an alternative to our previous branch and bound algorithm [8]. Dynamic programming [9] requires an optimality property allowing to build solutions to the problem from precomputed partial solutions. Such an optimality property in our problem is not obvious, for if we examine the expression of the cost $\rho(\pi)$, we see that the additional rate due to a "send" action depends not only on the preceding action, but on all the previous ones.

We now state an optimality assumption allowing the computation of all optimal policies by dynamic programming. We define the $(i-1)$*-prefix* of a policy π of length i as the restriction $\pi \cap \{1, \ldots, i-1\}$ of π to the first $i-1$ transmission opportunities. If the assumption is verified, we can build optimal policies of length i recursively using previously computed optimal $(i-1)$-prefixes.

Assumption 1 (Prefix optimality assumption). *Let π be an optimal policy of length n and let π' be the $(n-1)$-prefix of π. Then π' is an optimal policy for the set of policies of length $n-1$ and size $|\pi'|$.*

In general, the prefix optimality assumption does not always hold. However, one can prove it when FTT and RTT have exponential distributions and $\epsilon_F = \epsilon_B = 0$ (see the Appendix), and simulations show it to be valid in many other cases. For this reason, we propose to always make this assumption, at least as a heuristic principle.

Given a number of transmission opportunities n, we want to find the set Π^* of all optimal policies. Let us denote by $\Pi^*(k, m),\ 1 \leq k \leq n,\ 0 \leq m \leq k$, the set of policies that are optimal among all policies of size m and length k. Then Π^* is the subset of optimal policies in $\bigcup_{m=0}^{n} \Pi^*(n, m)$. Thus, letting $\Pi_1 \sqcup \Pi_2 = \bigsqcup_{i=1}^{2} \Pi_i$ denote the optimal subset of $\Pi_1 \cup \Pi_2$ (i.e., $\pi \in \Pi_1 \sqcup \Pi_2$ if $\pi \in \Pi_1 \cup \Pi_2$ and π is optimal in $\Pi_1 \cup \Pi_2$) and $\bigsqcup_{i=0}^{l} \Pi_i = (\bigsqcup_{i=0}^{l-1} \Pi_i) \bigsqcup \Pi_l$ for

$l > 1$, we have $\Pi^* = \bigsqcup_{m=0}^{n} \Pi^*(n, m)$. Note that once the elements of Π_1 and Π_2 are known, the set $\Pi_1 \sqcup \Pi_2$ can be computed in $O(|\Pi_1| + |\Pi_2|)$ time. To compute the sets $\Pi^*(n, m),\ m = 0, \ldots, n,$ we need all sets $\Pi^*(k, m)$, $1 \le k \le n$, $0 \le m \le k$, which can be constructed as follows. Let $\Pi^{(k)} = \{\pi \cup \{k\} | \pi \in \Pi\}$. Then according to Assumption 1, we have

$$\Pi^*(k, m) = \Pi^*(k-1, m-1)^{(k)} \sqcup \Pi^*(k-1, m). \quad (3)$$

That is, the set of all optimal policies of length k and size m is simply given by the optimal policies in the union of two sets: one is the set of optimal policies of length $k-1$ and size $m-1$ in which we add a "send" action at the kth transmission opportunity, and the other one is the set of all optimal policies of length $k-1$ and size m. After initializing $\Pi^*(k, 0) = \{\emptyset\}$ for $k = 1, \ldots, n$ and $\Pi^*(1, 1) = \{\{1\}\}$, we use (3) to compute all sets $\Pi^*(k, m)$, $1 \le k \le n$, $0 \le m \le k$, in lexicographical order.

If we do not count the time for computing the costs and errors of the policies, the time complexity to compute Π^* is $O(A(n))$, where $A(n) = \sum_{k=1}^{n} \sum_{m=0}^{k} |\Pi^*(k, m)|$ is the number of policies checked by the dynamic programming algorithm. The cost and error of a policy π of length k can be derived from those of its $(k-1)$-prefix. This allows us to compute the cost of π in $O(n)$ time and its error in constant time. Thus, the overall time complexity of the algorithm is $O(nA(n))$, which is bounded by $O(n2^n)$. The worst-case bound of $O(n2^n)$ is not reached in practice since $A(n)$ is usually much smaller than 2^n (see the experimental results). Note that $O(n2^n)$ is also the worst-case time complexity of the branch and bound algorithm of [8].

4. EXPERIMENTAL RESULTS

We compared the performance of our dynamic programming algorithm to that of the branch and bound algorithm of [8]. We used the number of prefixes that had to be checked for optimality as the performance measure. For n transmission opportunities, this number is equal to $A(n)$ when the dynamic programming algorithm is used. Note that the number of prefixes checked was also the performance measure used in [8]. Fig. 1 shows the results for shifted gamma distributed random variables FTT and BTT with rightward shifts κ_F and κ_B, and parameters n_F, α_F, n_B, and α_B (see [2, 8]). The same channel parameters as in [8] were used in Fig. 1 (a) and (b). The figure also shows the complexity of a full search algorithm. All three algorithms always found the same (optimal) solutions. When the number of transmission opportunities was small, the dynamic programming algorithm checked slightly more prefixes than the branch and bound method because all policies that turn out to be optimal have to be checked twice during the dynamic programming process. The first check occurs during the construction of $\Pi^*(k, m)$ and the second one during the construction of Π^*. With increasing number of transmission opportunities, our dynamic programming method outperformed the branch and bound method, yielding a speed-up factor of up to 32 for 32 transmission opportunities.

5. CONCLUSION

We proposed a dynamic programming algorithm for computing all optimal policies for a single packet in rate-distortion optimized media streaming. Our algorithm finds optimal policies if the prefix optimality assumption is fulfilled. We proved this assumption for exponentially distributed forward and round trip times. Moreover, when the forward and backward trip times were modeled as shifted gamma distributions in our experiments, our algorithm always provided optimal results and was significantly faster than the best previous exact solution of [8]. Since all single policies of an optimal policy vector must be optimal [8], our dynamic programming algorithm can be used as a preprocessing step for computing optimal policy vectors. Future work may include proving the prefix optimality assumption for more general channel models.

6. APPENDIX

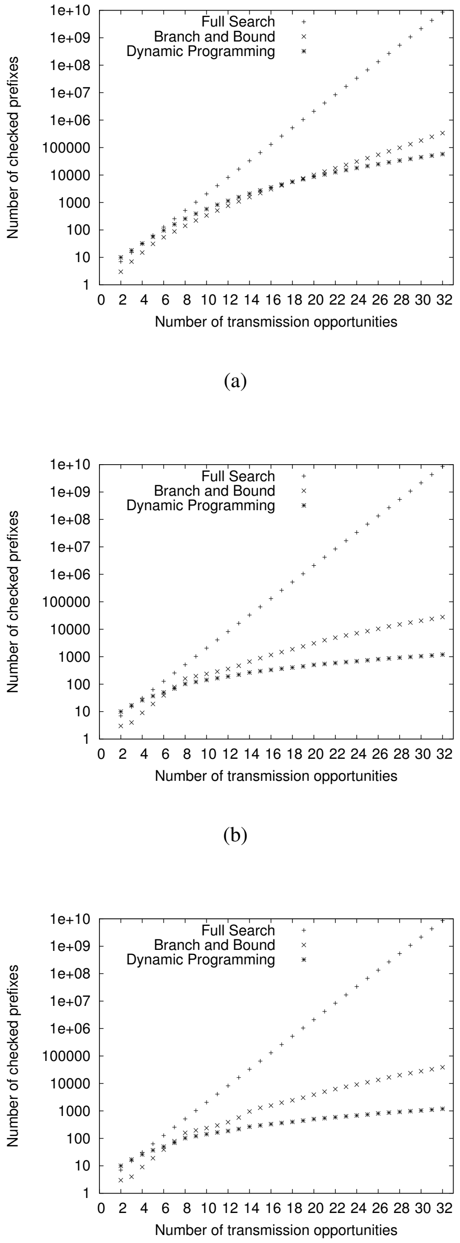
Proposition 1. *If $\epsilon_F = \epsilon_B = 0, P\{FTT > x\} = e^{-\alpha x}$, and $P\{RTT > x\} = e^{-\beta x}$ for some positive real numbers α and β, then the prefix optimality assumption is verified.*

Proof. We prove the proposition by contradiction. Let $\pi \subseteq \{1, \ldots, n\}$ be a transmission policy, and π' its $(n-1)$-prefix. If $n \notin \pi$, then $\pi' = \pi$. Thus, $\epsilon(\pi) = \epsilon(\pi')$ and $\rho(\pi) = \rho(\pi')$. Otherwise,

$$\begin{aligned} \epsilon(\pi) &= \prod_{i \in \pi} P\{FTT > s_D - s_i\} \\ &= \epsilon(\pi') P\{FTT > s_D - s_n\} \\ &= \epsilon(\pi') e^{-\alpha(s_D - s_n)}, \end{aligned} \quad (4)$$

and

$$\begin{aligned} \rho(\pi) &= \sum_{i \in \pi} \left(\prod_{\substack{j<i \\ j \in \pi}} P\{RTT > s_i - s_j\} \right) \\ &= \rho(\pi') + \prod_{j \in \pi'} P\{RTT > s_n - s_j\} \\ &= \rho(\pi') + \prod_{j \in \pi'} e^{-\beta(s_n - s_j)} \\ &= \rho(\pi') + e^{-\beta(s_n - s_D)|\pi'|} \prod_{j \in \pi'} e^{-\beta(s_D - s_j)} \\ &= \rho(\pi') + e^{-\beta(s_n - s_D)|\pi'|} \epsilon(\pi')^{\frac{\beta}{\alpha}}. \end{aligned} \quad (5)$$



(a)

(b)

(c)

Fig. 1. Number of checked prefixes as a function of the number of transmission opportunities for full search, the branch and bound algorithm of [8] and our dynamic programming algorithm. The time interval between two opportunities is 50 ms. The channel conditions are (a) $\epsilon_F = \epsilon_B = 0.2, \kappa_F = \kappa_B = 25$ ms $, n_F = n_B = 2, 1/\alpha_F = 1/\alpha_B = 12.5$ (b) $\epsilon_F = \epsilon_B = 0.01, \kappa_F = \kappa_B = 25$ ms $, n_F = n_B = 8, 1/\alpha_F = 1/\alpha_B = 12.5$, and (c) $\epsilon_F = \epsilon_B = 0.2, \kappa_F = \kappa_B = 25$ ms $, n_F = n_B = 8, 1/\alpha_F = 1/\alpha_B = 12.5$.

If π is optimal, there exists no other transmission policy μ with $\epsilon(\mu) \leq \epsilon(\pi)$ and $\rho(\mu) < \rho(\pi)$. Now suppose that π', the $(n-1)$-prefix of π, be non-optimal among the $(n-1)$-prefixes of the same size $|\pi'|$, i.e., there exists $\mu' \subseteq \{1, 2, \ldots, n-1\}$ with $|\mu'| = |\pi'|$, $\epsilon(\mu') \leq \epsilon(\pi')$ and $\rho(\mu') < \rho(\pi')$. We will show that π cannot be optimal by comparing it to a policy μ constructed from μ'.

If $n \notin \pi$, then we set $\mu = \mu'$. Since $\pi = \pi'$, we have $\epsilon(\mu) \leq \epsilon(\pi)$ and $\rho(\mu) < \rho(\pi)$. Thus π is not optimal.

If $n \in \pi$, then we take $\mu = \mu' \cup \{n\}$. Then from (4) and the fact that $\epsilon(\mu') \leq \epsilon(\pi')$, we have that $\epsilon(\mu) \leq \epsilon(\pi)$. Similarly, from (5) and the fact that $\epsilon(\mu') \leq \epsilon(\pi')$ and $\rho(\mu') < \rho(\pi')$, we obtain $\rho(\mu) < \rho(\pi)$. Again, π is not optimal. □

7. REFERENCES

[1] D. Wu, Y.T. Hou, W. Zhu, Y.-Q. Zhang, and J.M. Peha, “Streaming video over the Internet: Approaches and directions”, *IEEE Trans. Circuits Syst. Video Technol,* vol. 11, pp. 282–300, March 2001.

[2] P.A. Chou and Z. Miao, Rate-distortion optimized streaming of packetized media, Microsoft Research Technical Report MSR-TR-2001-35, Feb. 2001.

[3] B. Girod, M. Kalman, Y. Liang, and R. Zhang, “Advances in channel-adaptive video streaming,” *Wireless Communications and Mobile Computing*, vol. 2, no. 6, pp. 549–552, Sept. 2002.

[4] J. Chakareski, P. A. Chou, and B. Aazhang, “Computing rate-distortion optimized policies for streaming media to wireless clients”, *Proc. DCC’02*, Snowbird, Utah, April 2002.

[5] M. Kalman, P. Ramanathan, and B. Girod, “Rate-distortion optimized video streaming with multiple deadlines”, *Proc. IEEE ICIP-03*, Barcelona, Sept. 2003.

[6] J. Chakareski and B. Girod, “Computing rate-distortion optimized policies for streaming media with rich acknowledgements”, *Proc. DCC’04*, Snowbird, UT, April 2004.

[7] J. Chakareski, J. Apostolopoulos, and B. Girod, “Low-complexity rate-distortion optimized streaming”, *Proc. IEEE ICIP-2004*, Singapore, Oct. 2004.

[8] M. Röder, J. Cardinal, R. Hamzaoui, “On the complexity of rate-distortion optimal streaming of packetized media”, *Proc. DCC’04*, Snowbird, UT, 2004.

[9] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.