Complexity Constrained Representation Selection for Dynamic Adaptive Streaming

Chenglin Li, Laura Toni, Pascal Frossard, Hongkai Xiong, Junni Zou

Abstract—**Dynamic adaptive streaming addresses the user heterogeneity by providing multiple encoded representations for different videos. However, the selection of the optimal source coding parameters of each encoded representation is still challenging for delay sensitive applications, such as live streaming. To address this, we propose a representation selection optimization problem for complexity constrained adaptive video streaming that properly takes into account the different complexity-rate-distortion (C-R-D) characteristics of the videos when implementing rate control for desired representations. Our objective is to maximize the expected video distortion reduction of users, subject not only to encoding rate constraints, but also to complexity constraints. We prove that our optimization problem is a submodular maximization problem with two knapsack constraints. A weighted rate and complexity cost benefit greedy algorithm is then developed to obtain an approximate solution with polynomial time complexity and good approximation performance in simulations.**

Dynamic adaptive video streaming, complexity-rate-distortion, rate control, submodular function maximization.

I. Introduction

The management of video streaming services has become a more complex task due to the ever-increasing heterogeneity of user population in terms of demands for specialized video contents, devices used to display, and access network capacity. Dynamic adaptive streaming over HTTP (DASH) has been recently proposed as an effective method to improve the overall user satisfaction by offering several representations of the same video content to the different clients [1]. Each representation is encoded by the DASH server at a pre-defined bitrate and/or resolution such that users can be served by the most suitable representation in accordance with their requirements and heterogeneous network conditions. While most of the research community focuses on the client-side adaptation schemes for given encoded representations, little work has been done to address the representation selection problem at the server [2]. This representation selection problem becomes more crucial for delay sensitive applications, e.g., real-time video streaming, with strict requirements on the encoding time (delay) and usually a total power budget for all the encoded representations. Constrained by such delay and power requirements, the server cannot encode as many representations as possible to individually serve each user's request. Noting that both the encoding time and power are closely related to the encoding complexity, it is therefore worth investigating the selection of the optimal representations encoded for each video with the corresponding encoder parameters under complexity constraints, i.e., the total complexity used to encode the desired representations should not exceed the maximum load affordable by the server. Meanwhile, the running time of such representation selection procedure should also be short enough to enable delay sensitive applications.

To address the above complexity issue, we formulate a joint representation selection and rate control optimization problem for DASH streaming with proper consideration of the C-R-D properties of representations from different videos, under both the encoding rate and complexity constraints. We further prove that the proposed optimization problem is a submodular maximization problem subject to two knapsack constraints, which is NP-hard. Thus, a weighted rate and complexity cost benefit greedy algorithm is developed in order to obtain an approximate solution with low (i.e., polynomial) time complexity and theoretical approximation guarantees. Simulation results show that the proposed algorithm can seek the tradeoff both between the rate and complexity cost and between the algorithm's performance and computational time.

Existing works address some of these issues partially. In [2] and [3], the optimal representation set selection problem of adaptive streaming under the encoding rate or power constraint is proposed as an integer linear program (ILP), revealing the best coding parameters in terms of the bitrate and resolution for each representation. However, it requires very high (usually exponential) computational complexity to solve this ILP, which is thus not feasible for delay sensitive applications. In addition, the specific source coding parameters (e.g., the quantization parameter, QP) needed to encode each desired representation with the optimal bitrate and resolution is unknown and not provided. As another line of research, the rate control scheme to achieve the minimum encoding distortion for single representation has been investigated in [4] under the consideration of delay, rate and power aspects. However, it is still unclear how to choose the optimal source coding parameters for multiple related representations simultaneously competing for the rate and complexity resource.

The rest of this paper is organized as follows. Sections II and III describe the notations, system models, and optimization formulation. In Section IV, we show that it is a submodular maximization problem and develop an approximate algorithm. Section V presents experimental results, and evaluates the proposed algorithm compared to the performance upper bound. Conclusion is given in Section VI.

II. C-R-D Model for Video Coding

In [4], the models of source coding complexity, rate and distortion have been derived for hybrid video coding. Both the source rate and distortion of an inter-coded frame are derived as functions of the standard deviation σ of the transformed residuals and the quantization step size Q. For a video $f \in \mathcal{F}$, the source rate is approximated by the entropy of the quantized transformed residuals, and the source distortion is mainly incurred by the quantization error:

$$R_f(L,Q) = -P_0 \log_2 P_0 + (1-P_0)\Bigg[\frac{LQ\log_2 e}{1-e^{-LQ}} - \log_2(1-e^{-LQ}) - LQ\gamma\log_2 e + 1\Bigg] \tag{1}$$

$$D_f(L,Q) = \frac{LQe^{\gamma LQ}(2+LQ-2\gamma LQ)+2-2e^{LQ}}{L^2(1-e^{LQ})} \tag{2}$$

where $L = \sqrt{2}/\sigma$ denotes the Laplacian distribution parameter; γQ represents the rounding offset and γ is a parameter between $(0,1)$, such as $1/6$ for H.264/AVC inter frame coding; $P_0 = 1 - e^{-LQ(1-\gamma)}$ is the probability of quantized transform coefficient being zero. Given a video $f \in \mathcal{F}$, σ can be well fitted by a closed form function of the search range λ in motion estimation and the quantization step size Q [4], as:

$$\sigma_f(\lambda,Q) = a_{f,1}\cdot e^{-a_{f,2}\cdot\lambda} + a_{f,3} + a_{f,4}\cdot Q \tag{3}$$

where $a_{f,1}$-$a_{f,4}$ are empirical parameters dependent on the encoding structure and the video content of f. Integrating Eq. (3) into Eqs. (1) and (2), both the source coding rate and distortion of video file f can be then expressed as functions of λ and Q, denoted as $R_f(\lambda,Q)$ and $D_f(\lambda,Q)$, respectively.

On the other hand, since motion estimation (ME) takes up the majority of the total encoding time, the encoding complexity can be approximated by the ME complexity, which is determined by the total number of CPU cycles consumed by the SAD (sum of absolute difference) operations in ME [4]. Thus, for the single reference frame prediction case and given the desired frame encoding time ΔT, the CPU load in clock frequency for encoding a specific video f can also be expressed as a function of λ and Q:

$$C_f(\lambda,Q) = \frac{N(2\lambda+1)^2\cdot\eta(Q)\cdot c_0}{\Delta T} \tag{4}$$

where N is the number of Macroblocks (MBs) in a frame; $(2\lambda+1)^2\cdot\eta(Q)$ is the total number of SAD operations in the two dimensional search area for each MB, and $\eta(Q)$ is an empirical parameter that denotes the ratio of the actual number of SAD operations in the practical video codec to the theoretical total number of SAD operations; c_0 is the number of clock cycles of one SAD operation over a given CPU.

III. Optimization Problem Formulation

The complexity constrained representation selection and rate control problem for DASH streaming can be summarized as: for a given set of source video files, file popularity distribution, and the users' downlink bandwidth, how to decide both the encoded representations for each video and the corresponding source coding parameters for each representation such that the total system utility in terms of the aggregate users' satisfaction is maximized, subject to both the total encoding rate and complexity (in CPU load) constraints of the DASH server.

For video files, let $\mathcal{F} = \{1,2,\ldots,F\}$ denote the set of F video files at the DASH server. Denote $\mathcal{M} = \Lambda \times \mathcal{Q}$ as the set of $M = |\mathcal{M}|$ possible representations. Each element in $\mathcal{M}$ corresponds to a specific source coding parameter pair (λ, Q) with $\lambda \in \Lambda$ and $Q \in \mathcal{Q}$, where Λ is the search range set containing all the possible search range values and $\mathcal{Q}$ denotes the quantization step size set including all the available quantization step sizes. We further sort the representation set $\mathcal{M}$ in an decreasing order of the encoding bitrate, i.e., $R_f(\lambda_{f,i},Q_{f,i}) > R_f(\lambda_{f,j},Q_{f,j}), \forall i,j \in \mathcal{M}$ and $1 \leq i < j \leq M$. The finite ground set of the DASH representation selection and rate control problem is:

$$\mathcal{V} = \{v_{f,m} | \forall f \in \mathcal{F},\ \forall m \in \mathcal{M}\} \tag{5}$$

The ground set $\mathcal{V}$ is defined in Eq. (5) to denote the full set of all representations of all video files that could be encoded by the DASH server, and a specific element $v_{f,m}$ represents that the m-th representation is encoded for video file f.

From the perspective of users, for each user $u \in \mathcal{U}$, let Ω_u denote the set of representations of all video files that can be downloaded by user u according to his/her download link bandwidth B_u, i.e., $\Omega_u = \{v_{f,m} \in \mathcal{V} | R_f(\lambda_{f,m},Q_{f,m}) \leq B_u,\ \forall f \in \mathcal{F},\ \forall m \in \mathcal{M}\}$. Then, based on a specific DASH encoding decision $\mathcal{A} \subseteq \mathcal{V}$ specifying that which representations should be encoded for which video files, the expected average reduction in video distortion for user u is:

$$\bar{D}_u(\mathcal{A}) = \sum_{f=1}^{F}\sum_{m=1}^{M}\Bigg[\prod_{j=1}^{m-1}(1 - \mathbf{1}|_{v_{f,j}\in(\mathcal{A}\cap\Omega_u)})\Bigg] \cdot \mathbf{1}|_{v_{f,m}\in(\mathcal{A}\cap\Omega_u)} \cdot \rho_{u,f} \cdot \Bigg[D_{max} - D_f(\lambda_{f,m},Q_{f,m})\Bigg] \tag{6}$$

In Eq. (6), $\mathbf{1}|_{v\in\mathcal{V}}$ is an indicator function the value of which is 1 if $v \in \mathcal{V}$ and 0 otherwise; the term $[\prod_{j=1}^{m-1}(1 - \mathbf{1}|_{v_{f,j}\in(\mathcal{A}\cap\Omega_u)})] \cdot \mathbf{1}|_{v_{f,m}\in(\mathcal{A}\cap\Omega_u)} = 1$ indicates that the m-th representation of video file f is the best representation that is both encoded at the server and can be downloaded according to user u's bandwidth, and 0 otherwise; $\rho_{u,f}$ is the probability of user u requesting video file f; and D_{max} represents a constant maximal distortion when no video is decoded and $[D_{max} - D_f(\lambda_{f,m},Q_{f,m})]$ denotes the distortion reduction (or quality improvement) after successful decoding the representation with coding parameter pair $(\lambda_{f,m},Q_{f,m})$.

Therefore, the optimization problem can be formulated as:

$$\max_{\mathcal{A}\subseteq\mathcal{V}} \quad D(\mathcal{A}) = \sum_{u\in\mathcal{U}} \bar{D}_u(\mathcal{A}) \tag{7a}$$

$$\text{s.t.} \quad \sum_{f=1}^{F}\sum_{m=1}^{M} \mathbf{1}|_{v_{f,m}\in\mathcal{A}} \cdot R_f(\lambda_{f,m},Q_{f,m}) \leq R_{max} \tag{7b}$$

$$\sum_{f=1}^{F}\sum_{m=1}^{M} \mathbf{1}|_{v_{f,m}\in\mathcal{A}} \cdot C_f(\lambda_{f,m},Q_{f,m}) \leq C_{max} \tag{7c}$$

The objective in Eq. (7a) is to maximize the overall system utility defined as the expected video distortion reduction of all users, and the decision variable is the actual encoded representation set $\mathcal{A} \subseteq \mathcal{V}$. The constraint in Eq. (7b) specifies that the sum of encoding bitrates of all representations does not exceed the bitrate capacity R_{max} which is constrained by either the storage capacity of the server or the bottleneck link of the network. The constraint in Eq. (7c) ensures that the overall complexity consumed to encode all representations is limited by the server's maximum CPU load C_{max}.

IV. Submodularity and Approximation Algorithm

Proposition 1. *The objective function in Eq. (7a) is a monotone submodular function*[1] *over the ground set $\mathcal{V}$ in Eq. (5).*

Proof. Through the definition and property of submodularity, the monotone submodularity of Eq. (7a) can be proved, which is omitted here due to the page limit. □

In Proposition 1, we have justified that Eq. (7a) is a monotone submodular function. Further observing the encoding rate and complexity constraints in Eqs. (7b) and (7c), each element $v_{f,m} \in \mathcal{A}$ has non-uniform rate and complexity cost of $R_f(\lambda_{f,m}, Q_{f,m})$ and $C_f(\lambda_{f,m}, Q_{f,m})$, while the DASH server has the encoding bitrate capacity and CPU load budget of R_{max} and C_{max}, respectively. These two constraints can be viewed as two knapsack constraints on the finite ground set $\mathcal{V}$. Therefore, the rate control problem in Eq. (7) is a submodular maximization problem subject to two knapsack constraints, which is generally NP-hard and requires exponential computational complexity to reach the optimum by either integer linear programming or other optimization methods [5].

To efficiently solve the constrained submodular maximization problem in Eq. (7) with polynomial time complexity and theoretical approximation guarantees, the (ω, k)-weighted cost benefit greedy algorithm is developed as shown in Algorithm 1. The two system parameters, $\omega \in [0, 1]$ and $k = 0, 1, 2, \ldots$, specify the weight between the rate and the complexity cost and the size of the initial set, respectively. Specifically, the proposed (ω, k)-WCB greedy algorithm considers all feasible initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ of cardinality k. Starting from any initial set $\mathcal{A}^0$, at step t, the weighted cost benefit greedy procedure iteratively searches over the remaining set $\mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}$ and inserts into the partial solution $\mathcal{A}^{t-1}$ an element according to Eqs. (8)-(10), until the remaining set reduces to an empty set. In other words, this procedure adds at each iteration an element that maximizes the weighted marginal benefit $D(\mathcal{A}^{t-1} \cup \{v_{f,m}\}) - D(\mathcal{A}^{t-1})$ and cost $R_f(\lambda_{f,m}, Q_{f,m}), C_f(\lambda_{f,m}, Q_{f,m})$ ratio among all elements still affordable with the remaining rate and complexity budget until no more element can be added. The weight parameter ω could adjust the tradeoff between the rate and complexity cost. In some extreme cases, for example, the algorithm reduces to be pure rate cost benefit when $\omega = 1$ and pure complexity cost benefit when $\omega = 0$. The proposed (ω, k)-WCB greedy algorithm then enumerates all initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ of cardinality k, augments each of them following the cost benefit greedy procedure, and selects the initial set achieving the largest value of the objective function $D(\mathcal{A}) = \sum_{u \in \mathcal{U}} \bar{D}_u(\mathcal{A})$ and sets its solution set as the final encoded representation set $\mathcal{A}$.

Algorithm 1 (ω, k)-weighted cost benefit greedy algorithm

```
For all initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ such that $|\mathcal{A}^0| = k$, implement the
following weighted cost benefit greedy procedure.
  Initialization:
    1) Set $\mathcal{V}^0 = \mathcal{V}$ and $t = 1$.
  Greedy Search Iteration: (at step $t = 1, 2, 3, \ldots$)
    1) Given a partial solution $\mathcal{A}^{t-1}$, find
       $v_{f_t,m_t} = \arg \max_{v_{f,m} \in \mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}} \omega \cdot \frac{D(\mathcal{A}^{t-1} \cup \{v_{f,m}\}) - D(\mathcal{A}^{t-1})}{R_f(\lambda_{f,m}, Q_{f,m})} + (1 - \omega) \cdot \frac{D(\mathcal{A}^{t-1} \cup \{v_{f,m}\}) - D(\mathcal{A}^{t-1})}{C_f(\lambda_{f,m}, Q_{f,m})}$   (8)
  Update and Determination:
    1) Set $\mathcal{A}^t = \mathcal{A}^{t-1} \cup \{v_{f_t,m_t}\}$, and $\mathcal{V}^t = \mathcal{V}^{t-1}$, if
       $\sum_{f=1}^{F} \sum_{m=1}^{M} \mathbf{1}|_{v_{f,m} \in (\mathcal{A}^{t-1} \cup \{v_{f_t,m_t}\})} \cdot R_f(\lambda_{f,m}, Q_{f,m}) \leq R_{max},$   (9)
    and
       $\sum_{f=1}^{F} \sum_{m=1}^{M} \mathbf{1}|_{v_{f,m} \in (\mathcal{A}^{t-1} \cup \{v_{f_t,m_t}\})} \cdot C_f(\lambda_{f,m}, Q_{f,m}) \leq C_{max};$   (10)
       otherwise, set $\mathcal{A}^t = \mathcal{A}^{t-1}$, and $\mathcal{V}^t = \mathcal{V}^{t-1} \setminus \{v_{f_t,m_t}\}$.
    2) If $\mathcal{V}^t \setminus \mathcal{A}^t \neq \emptyset$, set $t = t + 1$ and return to the greedy
  search iteration; otherwise, stop the iteration.
The solution is obtained and output as $\mathcal{A}$, which has the largest
value of the objective function $D(\mathcal{A}) = \sum_{u \in \mathcal{U}} \bar{D}_u(\mathcal{A})$ over all
the possible choices of the initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$.
```

In terms of computational complexity, the running time of the proposed algorithm is $O((FM)^{k+2}U)$, indicating a polynomial time complexity. As the value of k increases, the running time of the proposed algorithm becomes larger while the performance improves. As shown in [6], when $k \geq 3$ and in the case of one active knapsack constraint, the theoretical worst-case performance guarantee of the cost benefit algorithm is $1 - 1/e$, i.e., its solution achieves at least the ratio $1 - 1/e$ of the optimal objective value.

V. Experimental Evaluation

We implement the proposed algorithm on a 48-processor server with 252 GB of RAM using Linux 3.1 kernel, where each processor is an Intel Xeon CPU E5-2680 at a clock frequency of 2.50GHz. Suppose that there are $U = 10$ users and their download bandwidth B_u is randomly distributed in the rate range of $[1, 10]$ Mbps as illustrated in Fig. (1a). Three test video sequences ($F = 3$, *Crowd Run*, *Tractor*, and *Sunflower*) with 1080p resolution (1920×1080), available at [7], are selected as the source video files to be encoded at the DASH server. These three test video sequences correspond to different content types, i.e., dense object motion for *Crowd*

[1] Let $\mathcal{V}$ be a finite ground set, and a set function $g : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular iff $g(\mathcal{X} \cup \{v\}) - g(\mathcal{X}) \geq g(\mathcal{Y} \cup \{v\}) - g(\mathcal{Y})$ for any sets $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{V}$ and for any element $v \in (\mathcal{Y} \setminus \mathcal{X})$.

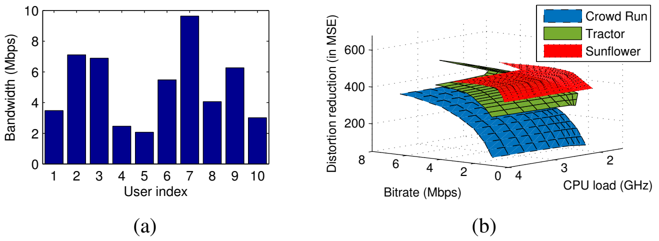

Fig. 1. (a) Download bandwidth of the $U = 10$ users. (b) Distortion reduction vs. encoding bitrate and complexity.

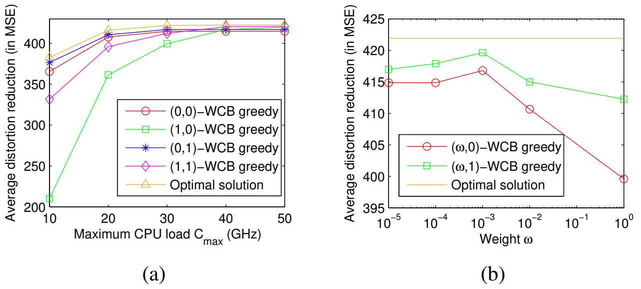

Fig. 2. Given $R_{max} = 30$ Mbps, average distortion reduction per user vs. (a) maximum CPU load C_{max}, and (b) weight ω when $C_{max} = 30$ GHz.

Run sequence, camera movement and medium object motion for *Tractor* sequence, and small object motion for *Sunflower* sequence, respectively. Assume that the encoding time of each video frame is fixed at $\Delta T = 3$ s, and the constant maximal distortion is set as $D_{max} = 500$. At frame rate of 30 fps, we further encode each video sequence f into $M = 63$ representations with the coding parameter pair $(\lambda_{f,m}, Q_{f,m}) \in \Lambda \times \mathcal{Q}$, where $\Lambda = \{2, 6, 10\}$ and the corresponding QP value ranges between 30 and 50. The distortion reduction versus encoding bitrate and complexity curved surfaces of these three sequences are illustrated in Fig. 1(b). Generally, it can be seen that the video content with smaller motion presents a higher curved surface in the three dimensional space than that with larger motion. For video with small object motion (e.g., *Sunflower*), the representation with both small encoding bitrate and low CPU load already introduces a large distortion reduction, while increasing either bitrate or complexity cannot incur significant additional distortion reduction; and vice versa. For the video file popularity, we further assume that these three sequences follow a Zipf distribution with parameter 0.56 [8], i.e., the requesting probabilities of *Crowd Run*, *Tractor*, and *Sunflower* sequences are 0.45, 0.31, and 0.24, respectively.

In Fig. (2a), we set the maximum bitrate capacity at the server to $R_{max} = 30$ Mbps, vary the value of maximum CPU loads C_{max}, and illustrate the average distortion reduction per user under different parameter settings of the proposed (ω, k)-WCB greedy algorithm. The optimal solution obtained by the generic solver IBM ILOG CPLEX [9] using branch and bound method with a very high (i.e., exponential) time complexity $O(2^{FMU})$ is given as a performance upper bound. It confirms that the proposed algorithm achieves a good approximation performance but with a more practical (i.e., polynomial) time complexity $O((FM)^{k+2}U)$. Through comparison, two observations can be made from the curves in Fig. (2a). Given a weight ω, enlarging the number of k incurs higher average distortion reduction per user for all values of C_{max}, but the time complexity would also increase from $O((FM)^2U)$ to $O((FM)^3U)$. On the other hand, when the size of initial set k is fixed, the algorithm performance is affected by the values of C_{max} and ω. It can be seen that when the maximum CPU load is small (e.g., $C_{max} = 10$ GHz), the algorithm with the minimum weight $\omega = 0$ (i.e., complexity cost benefit, 0.984 approximation ratio for $k = 1$) outperforms the weight assignment of $\omega = 1$ (i.e., rate cost benefit, 0.866 approximation ratio for $k = 1$), and vice versa. The reason is as follows. For small C_{max}, complexity becomes a more scarce resource compared to rate, which causes the CPU load constraint to be active while the encoding bitrate constraint remains inactive. In this case, the complexity cost benefit greedy algorithm that adds at each iteration step an element maximizing the marginal benefit and complexity cost ratio would achieve better performance.

When $C_{max} = 30$ GHz, both the CPU load and encoding bitrate constraints become active. In Fig. (2b), the average distortion reduction per user versus weight ω is shown for the cases of $k = 0$ and 1, respectively. Again, for a given value of ω, larger k indicates higher average distortion reduction. What can be further observed is that for both values of k there exists an optimal weight $\omega^* = 0.001$ achieving the peak average distortion reduction (0.988 and 0.995 approximation ratio for $k = 0$ and 1), which indicates the best tradeoff between the complexity and rate cost when both resources are limited.

VI. Conclusion

This paper studied an encoding complexity constrained representation selection and rate control problem for DASH streaming to maximize the expected aggregate video distortion reduction. It was proved to be a submodular maximization problem with an approximate algorithm provided. Experimental results have shown that the proposed algorithm could seek the tradeoff between the rate and complexity cost and between the approximation performance and computational complexity.

References

[1] T. Stockhammer, “Dynamic adaptive streaming over HTTP: standards and design principles,” in *Proc. ACM MMSys*, 2011, pp. 133–144.

[2] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard, “Optimal selection of adaptive streaming representations,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 2s, p. 43, Feb. 2015.

[3] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon, “Transcoding live adaptive video streams at a massive scale in the cloud,” in *Proc. ACM MMSys*, 2015, pp. 49–60.

[4] C. Li, D. Wu, and H. Xiong, “Delay-power-rate-distortion model for wireless video communication under delay and energy constraints,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1170–1183, Jul. 2014.

[5] A. Krause and D. Golovin, “Submodular function maximization,” *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[6] M. Sviridenko, “A note on maximizing a submodular set function subject to a knapsack constraint,” *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.

[7] “Xiph.org video test media.” [Online]. Available: http://media.xiph.org/video/derf/

[8] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network–measurements, models, and implications," *Computer networks*, vol. 53, no. 4, pp. 501–514, 2009.

[9] IBM, "ILOG CPLEX optimization studio." [Online]. Available: http://is.gd/3GGOFp