

CONSTRAINED OPTIMIZATION TECHNIQUES FOR GENERATING ENCODING LADDERS FOR VIDEO STREAMING

CROSS-REFERENCE TO RELATED APPLICATIONS

- 5 **[0001]** This application claims benefit of United States patent application serial number 18/154,680 filed January 13, 2023, which is hereby incorporated herein by reference.

BACKGROUND

Field of the Various Embodiments

- 10 **[0002]** The various embodiments relate generally to computer science and to video streaming technology and, more specifically, to constrained optimization techniques for generating encoding ladders for video streaming.

Description of the Related Art

- 15 **[0003]** A typical video streaming service provides users with access to a library of media titles that can be viewed on a wide range of different client devices. In operation, a given client device connects to the video streaming service under a variety of connection conditions and, therefore, can be susceptible to differing network throughputs. In an effort to ensure that a given media title can be streamed to a client device without playback interruptions, a video streaming service normally generates
- 20 an encoding ladder for the media title. Each rung of the encoding ladder specifies a different encoded video for the media title, the resolution of the encoded video, and the bitrate of the encoded video. Notably, an encoded video having a given bitrate can be streamed to a client device without playback interruptions when the network throughput is greater than that bitrate. As the bitrate of an encoded video for a given
- 25 media title that is streamed to a client device increases, the visual quality of the media title as presented on the client device usually increases as well.

- 30 **[0004]** In practice, the encoded videos specified in the encoding ladders for a library of media titles are normally delivered to client devices via a content delivery network (CDN) that has limited storage resources. Accordingly, generating an encoding ladder for a media title usually involves making tradeoffs between a streaming QoE associated with the encoding ladder and a storage footprint of the encoding ladder. As used herein, a “streaming QoE” associated with an encoding

ladder for a media title refers to an overall QoE for viewers when the encoding ladder is used to stream the media title to client devices. In practice, the streaming QoE usually correlates to the overall visual quality of the media title as streamed to and presented on client devices. A “storage footprint” for an encoding ladder refers to the
5 total size of the encoded videos specified in the encoding ladder.

[0005] In one approach to generating an encoding ladder for a media title, the encoding ladder is incrementally constructed based on heuristics corresponding to a set of ladder constraints. Collectively, the ladder constraints are designed to ensure that requisite streaming QoEs can be achieved when the media title is transmitted to
10 a variety of client devices over networks having a wide range of throughputs. To generate a given encoding ladder for a media title, a relatively large number of different encoded videos representing many different combinations of resolution and bitrate are produced based on a source video of the media title. Starting from an initially empty encoding ladder, the heuristic for each ladder constraint is sequentially
15 executed on the encoding ladder based on the different encoded videos, where the heuristic for a given ladder constraint determines whether the encoding ladder already complies with the ladder constraint. If the encoding ladder already complies with the ladder constraint, then the heuristic does not modify the encoding ladder. Otherwise, the heuristic adds at least one encoded video to the encoding ladder in order to bring
20 the encoding ladder into compliance with that particular ladder constraint.

[0006] One drawback of the above approach is that, because the ladder constraints are enforced one-at-a-time, and no encoded video is ever removed from the encoding ladder, the streaming QoE/storage footprint represented by the encoding ladder can be sub-optimal. More specifically, because ladder constraints
25 are enforced sequentially, opportunities to intentionally select a single encoded video that satisfies multiple ladder constraints in order to improve the streaming QoE/storage footprint tradeoff are missed. In such cases, the streaming QoE of the media title can be unnecessarily low, meaning that the average visual quality of the media title achieved using the encoded videos specified in the encoding ladder is too
30 low given the storage footprint of the encoding ladder. Conversely, the storage footprint of the encoding ladder can be unnecessarily large, meaning that the size of the storage footprint of the encoding ladder is too large given the average visual quality of the media title achieved using the encoded videos specified in the encoding

ladder. When this issue exists, the CDN storage resources could be more efficiently utilized by taking advantage of opportunities to satisfy multiple ladder constraints via a single encoded video to generate an improved encoding ladder for the media title. The improved encoding ladder would have a reduced storage footprint and would
5 provide the same or higher streaming QoE.

[0007] As the foregoing illustrates, what is needed in the art are more effective techniques for generating encoding ladders for video streaming.

SUMMARY

[0008] One embodiment sets forth a computer-implemented method for generating
10 encoding ladders that are used to stream media titles. The method includes generating a first objective function based on a first ladder configuration and a parameterized objective function that approximates a tradeoff between a quality of experience and a cost term associated with a candidate encoding ladder; generating a first set of constraints based on the first ladder configuration and a set of
15 parameterized constraints; and executing a constrained optimization algorithm on the first objective function, the first set of constraints, and encoding point metadata associated with a set of encoded videos to generate a first candidate encoding ladder for a media title.

[0009] At least one technical advantage of the disclosed techniques relative to the
20 prior art is that, with the disclosed techniques, encoding ladders can be generated based on an overall objective of reducing the storage footprint of an encoding ladder while increasing the visual quality levels associated with the encoded videos included in the encoding ladder by concurrently accounting for different ladder constraints when generating the encoding ladder in the first instance. With such an approach,
25 opportunities to use a single encoded video that satisfies multiple different ladder constraints can be identified and exploited when generating an encoding ladder, which improves the tradeoff between the weighted average of the visual quality levels associated with the encoded videos in the encoding ladder and the storage footprint of the encoding ladder. Consequently, the tradeoff between a streaming quality of
30 experience represented by an encoding ladder for a given media title and the storage footprint of the encoding ladder can be substantially improved relative to what can be

achieved using prior art techniques. These technical advantages provide one or more technological improvements over prior art approaches.

BRIEF DESCRIPTION OF THE DRAWINGS

5 [00010] So that the manner in which the above recited features of the various embodiments can be understood in detail, a more particular description of the inventive concepts, briefly summarized above, may be had by reference to various embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of the inventive concepts and are therefore not to be considered limiting of scope in any
10 way, and that there are other equally effective embodiments.

[0010] Figure 1 is a conceptual illustration of a system configured to implement one or more aspects of the various embodiments;

[0011] Figure 2 is a more detailed illustration of the encoding ladder application of Figure 1, according to various embodiments;

15 [0012] Figure 3 is a more detailed illustration of one of the rung assignment engines of Figure 2, according to various embodiments;

[0013] Figure 4 is a more detailed illustration of the simulation evaluation application of Figure 1, according to various embodiments;

20 [0014] Figure 5 is a flow diagram of method steps for generating candidate encoding ladders for use when streaming a media title, according to various embodiments; and

[0015] Figure 6 is a flow diagram of method steps for determining an encoding ladder for a media title based on adaptive streaming simulations of candidate encoding ladders that have been generated for the media title, according to various
25 embodiments.

DETAILED DESCRIPTION

[0016] In the following description, numerous specific details are set forth to provide a more thorough understanding of the various embodiments. However, it will be apparent to one skilled in the art that the inventive concepts may be practiced

without one or more of these specific details. For explanatory purposes, multiple instances of like objects are symbolized with reference numbers identifying the object and parenthetical alphanumeric character(s) identifying the instance where needed.

[0017] In an effort to ensure that a given media title can be streamed to a client device without playback interruptions, a video streaming service normally generates an encoding ladder for the media title. Each rung of the encoding ladder specifies a different encoded video for the media title, the resolution of the encoded video, and the bitrate of the encoded video. Notably, an encoded video having a given bitrate can be streamed to a client device without playback interruptions when the available network bandwidth is greater than that bitrate. As the bitrate of an encoded video for a given media title that is streamed to a client device increases, the visual quality of the media title as presented on the client device usually increases as well.

[0018] In practice, the encoded videos specified in the encoding ladders for a library of media titles are normally partitioned into encoded chunks, and the resulting encoded chunks are delivered to client devices via a CDN. To playback a given media title, a client device executes an endpoint application. Oftentimes, the endpoint application implements an adaptive bitrate algorithm that selects from the different encoded videos specified in the encoding ladder for the media title based on the network throughput and optionally the resolution of an associated screen. The endpoint application transmits a sequence of requests for chunks of the selected encoded video to an edge server device that is included in the CDN and resides relatively close to the client device. As the various encoded chunks are received by the endpoint application, the endpoint application decodes and, when necessary, upscales the encoded chunks to generate reconstructed chunks having the same resolution as the associated screen. The endpoint application plays back the different reconstructed chunks, thereby playing back the media title on the client device.

[0019] Because CDNs have limited storage resources, generating an encoding ladder for a media title usually involves making tradeoffs between a streaming QoE associated with the encoding ladder and a storage footprint of the encoding ladder. As used herein, the “streaming QoE” associated with an encoding ladder refers to an average QoE of viewers of a media title that has been encoded and streamed to client devices. The streaming QoE reflects both visual quality levels associated with the encoded chunks used for streaming and the impact of any re-buffering events or other

events that result in playback interruptions on the overall quality of the viewing experience. A “storage footprint” for an encoding ladder refers to the total size of the encoded videos specified in the encoding ladder.

[0020] In one approach to generating an encoding ladder for a media title, the encoding ladder is incrementally constructed based on heuristics corresponding to a set of ladder constraints. The ladder constraints are designed to ensure that requisite streaming QoEs can be achieved when the media title is transmitted to a wide range of client devices over networks having variable and different throughputs to enable viewing of the media title on screens having different resolutions. To generate an encoding ladder, different encoded videos representing many different combinations of resolution and bitrate are produced based on a source video of the media title. Starting from an initially empty ladder, the heuristic corresponding to each ladder constraint is sequentially applied to the ladder. If the encoding ladder already complies with the ladder constraint, then the heuristic does not modify the encoding ladder. Otherwise, the heuristic adds at least one of the encoded videos to the encoding ladder in order to bring the encoding ladder into compliance with that particular ladder constraint.

[0021] One drawback of the above approach is that, because the ladder constraints are enforced one-at-a-time, and no encoded video is ever removed from the encoding ladder, the streaming QoE/storage footprint represented by the encoding ladder can be sub-optimal. In particular, because ladder constraints are enforced sequentially, opportunities to intentionally select a single encoded video that satisfies multiple ladder constraints in order to improve the streaming QoE/storage footprint tradeoff are missed. In such cases, the size of the storage footprint of the encoding ladder is unnecessarily large given the average visual quality of the media title achieved using the encoded videos specified in the encoding ladder. Consequently, CDN storage resources are squandered.

[0022] With the disclosed techniques, however, an encoding ladder application generates one or more candidate encoding ladders for a media title based on an explicit goal of increasing streaming QoE while decreasing one or more costs (such as the storage footprint) and concurrently satisfying the ladder constraints. In some embodiments, an encoding ladder workflow executes the encoding ladder application to generate multiple candidate optimized ladders. The encoding ladder workflow then

uses a numerical evaluation application and a simulation evaluation application to determine a final encoding ladder for the media title.

[0023] The encoding ladder application formulates the problem of generating an encoding ladder as a parameterized constrained optimization problem of assigning
5 bitrate-quality points to rungs of a candidate encoding ladder. The encoding ladder application determines the constants of the parameterized constraint optimization problem based on bitrate-quality points that are generated based on a source video associated with the media title. Each bitrate-quality point specifies a different
10 encoded video derived from the source video and the corresponding resolution, bitrate, and visual quality score. Because bitrates and visual quality scores associated with encoded videos usually, if not always, have different magnitudes, the encoding ladder application optionally normalizes the bitrates and visual quality scores of the bitrate-quality points to the same range.

[0024] The encoding ladder application defines the objective and constraints of the
15 parameterized constrained optimization problem via a parameterized objective function and parameterized constraints. The parameterized objective function represents a weighted tradeoff between a quality term that approximates a streaming QoE represented by a candidate encoding ladder and a footprint term that is proportional to the storage footprint of the candidate encoding ladder. The quality
20 term is a weighted average of normalized visual quality scores across rungs of the candidate encoding ladder. The footprint term is the sum of normalized bitrates across the rungs. The number of rungs and the weights are parameters of the parameterized objective function. The parameterized constraints include both implicit logic constraints and practical logical constraints. Implicit logic constraints ensure the
25 validity of candidate encoding ladders. Practical logical constraints capture operational restrictions and/or preferences that are associated with capabilities of client devices, network capacity, a CDN, human perception of visual quality, etc.

[0025] The encoding ladder application generates multiple ladder configurations, where each ladder configuration is a different combination of values for the number of
30 rungs, the weights, and various parameters of the parameterized constraints (e.g., a relative bitrate spacing). For each ladder configuration, the encoding ladder generates an objective function and associated constraints based on the parameterized objective function and the parameterized constraints, respectively.

The encoding ladder application uses a constrained optimization algorithm to solve each of the objective functions subject to the associated constraints, thereby generating a different assignment matrix for each ladder configuration. Each assignment matrix specifies a different assignment of bitrate-quality points to each
5 rung of a different candidate encoding ladder.

[0026] The numerical evaluation application performs numerical evaluations of the candidate encoding ladders using statistical data (e.g., throughput distributions, bitrate demand distributions) derived from historical streaming sessions. In some embodiments, the numerical evaluation application filters out any number (including
10 zero) of candidate encoding ladders representing sub-par tradeoffs between streaming QoE and storage footprint.

[0027] In some other embodiments, the numerical evaluation application can estimate any number and/or types of streaming QoE metrics in any technically feasible fashion. Some examples of other streaming QoE metrics are time-weighted
15 visual quality, time-weighted bitrate, percentage of a predefined “excellent” quality, percentage of a predefined “low” quality, and probability of re-buffering. In the same or other embodiments, the numerical evaluation application can filter-out any number (including zero) of candidate encoding ladders based on tradeoffs across multiple dimensions, such as tradeoffs between expected streaming QoE, storage footprint,
20 and network bandwidth consumption.

[0028] The streaming evaluation application performs simulation-based evaluations of the remaining (e.g., unfiltered) candidate encoding ladders. For each of the remaining candidate encoding ladders, the streaming evaluation application generates a different synthetic streaming header based on the encoded videos
25 specified in the candidate encoding ladder. For each encoded video specified in a given candidate encoding ladder, the corresponding synthetic streaming header specifies the encoded video, the resolution of the encoded video, the bitrate of the encoded video, a corresponding sequence of encoded chunks, a bitrate for each of the encoded chunks, and a visual quality score for each of the encoded chunks.

[0029] The streaming evaluation application uses an adaptive streaming simulator to emulate the behavior of an adaptive bitrate algorithm using each of the candidate encoding ladders and the corresponding encoded chunk metadata over multiple

simulated streaming sessions characterized by different streaming session traces. Each streaming session trace specifies network throughput as a function of time for a different historical streaming session. The result of each simulation is a request sequence of encoded chunks of the media title. For each request sequence, the
5 streaming evaluation application computes a different set of values for a set of metrics that are relevant to streaming QoE. The streaming evaluation application performs any number and/or types of comparison operations between the sets of values for the set of metrics to select one of the remaining candidate encoding ladders. The streaming evaluation application generates an encoding ladder for the media title
10 based on the selected candidate encoding ladder.

[0030] At least one technical advantage of the disclosed techniques relative to the prior art is that, with the disclosed techniques, the encoding ladder application concurrently satisfies ladder constraints while explicitly optimizing an approximate streaming QoE/storage footprint tradeoff to generate each candidate encoding ladder.
15 Unlike prior art techniques, the encoding ladder application therefore automatically identifies and exploits opportunities to use a single encoded video to satisfy multiple different ladder constraints and improve the streaming QoE/storage footprint tradeoff. Consequently, the streaming QoE/storage footprint tradeoff represented by a candidate encoding ladder for a given media title can be substantially improved
20 relative to what can be achieved using prior art techniques. Another advantage of the disclosed techniques is that the simulation evaluation application can efficiently compare the performance of a significantly larger number of candidate encoding ladders using less time, processing resources, and network resources than would be required to deploy and evaluate the candidate encoding ladders over actual networks.
25 These technical advantages provide one or more technological improvements over prior art approaches.

[0031] Figure 1 is a conceptual illustration of a system 100 configured to implement one or more aspects of the various embodiments. For explanatory purposes, multiple instances or versions of like objects are denoted with reference
30 numbers identifying the object and parenthetical alphanumeric character(s) identifying the instance or version where needed. As shown, in some embodiments, the system 100 includes, without limitation, a compute instance 110(1), a compute instance

110(2), a historical streaming session database 104, and an adaptive streaming simulator 106.

[0032] In some other embodiments, the system 100 can omit the compute instance 110(1), the compute instance 110(2), the historical streaming session database 104, the adaptive streaming simulator 106, or any combination thereof. In the same or other embodiments, the system 100 can include, without limitation, one or more other compute instances, one or more other historical streaming session databases, or any combination thereof. The components of the system 100 can be distributed across any number of shared geographic locations and/or any number of different geographic locations and/or implemented in one or more cloud computing environments (i.e., encapsulated shared resources, software, data, etc.) in any combination.

[0033] As shown, the compute instance 110(1) includes, without limitation, a processor 112(1) and a memory 116(1), and the compute instance 110(2) includes, without limitation, a processor 112(2) and a memory 116(2). The compute instance 110(1) and the compute instance 110(2) are also referred to herein individually as “the compute instance 110” and collectively as “the compute instances 110.” The processor 112(1) and the processor 112(2) are also referred to herein individually as “the processor 112” and collectively as “the processors 112.” The memory 116(1) and the memory 116(2) are also referred to herein individually as “the memory 116” and collectively as “the memories 116.” Each compute instance (including the compute instances 110) can be implemented in a cloud computing environment, implemented as part of any other distributed computing environment, or implemented in a stand-alone fashion.

[0034] The processor 112 can be any instruction execution system, apparatus, or device capable of executing instructions. For example, the processor 112 could comprise a central processing unit, a graphics processing unit, a controller, a micro-controller, a state machine, or any combination thereof. The memory 116 of the compute instance 110 stores content, such as software applications and data, for use by the processor 112 of the compute instance 110. The memory 116 can be one or more of a readily available memory, such as random-access memory, read only memory, floppy disk, hard disk, or any other form of digital storage, local or remote.

[0035] In some other embodiments, any number of compute instances can include any number of processors and any number of memories in any combination. In particular, the compute instance 110(1), the compute instance 110(2), any number of other compute instances, or any combination thereof can provide a multiprocessing
5 environment in any technically feasible fashion.

[0036] In some embodiments, a storage (not shown) may supplement or replace the memory 116 of the compute instance 110. The storage may include any number and type of external memories that are accessible to the processor 112 of the compute instance 110. For example, and without limitation, the storage can include a
10 Secure Digital Card, an external Flash memory, a portable compact disc read-only memory, an optical storage device, a magnetic storage device, or any suitable combination of the foregoing

[0037] In general, each compute instance (including the compute instances 110) is configured to implement one or more software applications. For explanatory
15 purposes only, each software application is described as residing in the memory 116 of a single compute instance and executing on the processor 112 of the same compute instance. However, in some embodiments, the functionality of each software application can be distributed across any number of other software applications that reside in the memories of any number of compute instances and
20 execute on the processors of any number of compute instances in any combination. Further, the functionality of any number of software applications can be consolidated into a single software application.

[0038] In particular, in some embodiments, a production encoding pipeline 120 resides in the memory 116(1) of the compute instance 110(1) and executes on the
25 processor 112(1) of the compute instance 110(1). As shown, the production encoding pipeline 120 includes, without limitation, a shot-based encoding application 132 and a ladder deployment application 190.

[0039] The shot-based encoding application 132 partitions a source video 102 into shots (not shown). The source video 102 includes, without limitation, any amount
30 and/or types of video content. Some examples of video content include, without limitation, any portion (including all) of feature length films, episodes of television programs, and music videos, to name a few. Each shot includes a sequence of

frames that usually have similar spatial-temporal properties and run for an uninterrupted period of time. In some embodiments, each shot is captured continuously from a single camera or virtual representation of a camera (e.g., in the case of computer animated videos). Together, the shots span the length of the
5 source video 102 in a contiguous, non-overlapping fashion.

[0040] The shot-based encoding application 132 downscales each of the shots to multiple different resolutions to generate lower-resolution shots. The shot-based encoding application 132 encodes each of the shots and each of the lower-resolution shots across different sets of one or more values for a set of one or more encoding
10 parameters to generate encoded shots having different combinations of resolutions and bitrates. The shot-based encoding application 132 computes the bitrate encoded shot and a quality score for each encoded shot.

[0041] As used herein, the bitrate of an encoded sequence of frames (*e.g.*, an encoded shot or an encoded video) refers to an average bitrate across the encoded
15 sequence of frames. The quality score of an encoded sequence of frames refers to a quality score of a reconstructed sequence of frames derived from the encoded sequence of frames. And the quality score of a reconstructed sequence of frames refers to an average estimated visual quality level across the reconstructed sequence of frames.

[0042] A quality score can be a value for any type of metric that correlates to visual quality in any technically feasible fashion. In some embodiments, each quality score is a value for a visual quality metric. Some examples of visual quality metrics include, without limitation, a peak signal-to-noise-ratio (PSNR), and a video multimethod assessment fusion (VMAF) metric. The VMAF metric estimates human-perceived
20 video quality of reconstructed video content (e.g., the reconstructed shots, reconstructed videos, etc.).

[0043] For each resolution, the shot-based encoding application 132 generates a convex hull (not shown) of bitrate-quality points based on the encoded shots having that resolution. Subsequently, the shot-based encoding application 132 sets convex
30 hull metadata 136 equal to a union of the bitrate-quality points included in the convex hulls for the different resolutions. The convex hull metadata 136 facilitates the generation of an encoding ladder for the media title.

[0044] The shot-based encoding application 132 partitions the source video 102 into any number of source chunks, where each source chunk includes a sequence of one or more shots. Each of the source chunks defines a portion of the media title that is to be independently requested by and transmitted to client devices during
5 streaming of the media title. Together, the chunks span the length of the source video 102 in a contiguous, non-overlapping fashion.

[0045] The shot-based encoding application 132 determines the encoded chunks of each encoded video that is specified (via a corresponding encoded video ID) in at least one of the bitrate-quality points included in the convex hull metadata 136 based
10 on the source chunks and the encoded shots. The shot-based encoding application 132 computes the bitrate and quality score for each of the encoded chunks of each encoded video specified in the convex hull metadata 136 to generate encoded chunk metadata 138.

[0046] The ladder deployment application 190 transmits the encoded chunks
15 specified in the encoding ladder for the media title to a CDN (not shown) for subsequent delivery from any number of server devices in the CDN to any number of client devices (not shown). The ladder deployment application 190 also transmits the encoding ladder for the media title to a playback server (not shown) that subsequently enables client devices to request encoded chunks from a proximate server in the
20 CDN based on the encoding ladder and an available network throughput to affect streaming of the media title.

[0047] As described previously herein, in one conventional approach to generating an encoding ladder for a media title, the encoding ladder is incrementally constructed based on heuristics corresponding to a set of ladder constraints and encoded videos
25 representing many different combinations of resolution and bitrate. The heuristic for each ladder constraint is sequentially executed on an initially empty encoding ladder based on the different encoded videos, where the heuristic for a given ladder constraint determines whether the encoding ladder already complies with the ladder constraint. If the encoding ladder already complies with the ladder constraint, then
30 the heuristic does not modify the encoding ladder. Otherwise, the heuristic adds at least one encoded video to the encoding ladder in order to bring the encoding ladder into compliance with that particular ladder constraint.

[0048] One drawback of the above approach is that, because the ladder constraints are enforced one-at-a-time, and no encoded video is ever removed from the encoding ladder, opportunities to intentionally select a single encoded video that satisfies multiple ladder constraints in order to improve the streaming QoE/storage footprint tradeoff represented by the encoding ladder can be missed. When such opportunities are missed, CDN storage resources could be more efficiently utilized by using an improved encoding ladder that would have a reduced storage footprint and would provide the same or higher streaming QoE.

Generating Encoding Ladders Using Constrained Optimization Techniques

10 [0049] To address the above problem, the system 100 includes, without limitation, an encoding ladder workflow 140 that uses constrained optimization techniques to generate an encoding ladder 178 for the media title associated with the source video 102. As shown, in some embodiments, the encoding ladder 178 includes, without limitation, a rung 180(1) - a rung 180(L), where L can be any positive integer. The
15 rung 180(1) specifies an encoded video ID 182(1), a resolution 184(1), and a bitrate 186(1). The encoded video ID 182(1) identifies an encoded video for the source video 102. The resolution 184(1) and the bitrate 186(1) specify the resolution and the bitrate of the encoded video corresponding to the encoded video ID 182(1).

[0050] The encoding ladder workflow 140 includes, without limitation, an encoding
20 ladder application 150, a numerical evaluation application 160, and a simulation evaluation application 170. As shown, in some embodiments, the encoding ladder workflow 140, the encoding ladder application 150, the numerical evaluation application 160, and the simulation evaluation application 170 reside in the memory 116(2) of the compute instance 110(2) and execute on the processor 112(2) of the
25 compute instance 110(2).

[0051] As shown, the encoding ladder application 150 generates a candidate encoding ladder set 158 for the media title corresponding to the source video 102 based on the convex hull metadata 136. The candidate encoding ladder set 158 includes one or more candidate encoding ladders (not shown in Figure 1) for the
30 media title. As used herein, a "candidate encoding ladder" for a media title refers to any encoding ladder that can be used as an encoding ladder for the media title. The encoding ladder application 150 formulates the problem of encoding ladder

generation as one or more different constrained optimization problems based on the convex hull metadata 136.

[0052] Each constrained optimization problem represents an overall objective of reducing the storage footprint of an encoding ladder while increasing the streaming QoE associated with the encoding ladder subject to an associated set of constraints. The importance of reducing the storage footprint of a candidate encoding ladder relative to increasing QoE associated with the candidate encoding ladder and/or the set of constraints vary across the constrained optimization problems. The encoding ladder application 150 independently solves each constrained optimization problem to generate a different candidate encoding ladder in the candidate encoding ladder set 158. The encoding ladder application 150 is described in greater detail below in conjunction with Figure 2.

[0053] The numerical evaluation application 160 performs any number and/or types of numerical evaluations on the candidate encoding ladder set 158 based on the historical streaming session database 104. The historical streaming session database 104 includes, recorded data associated with any number of past streaming sessions and any amount (including none) of data derived from the recorded data. In particular, the numerical evaluation application 160 uses a throughput distribution 122 and/or a bitrate demand distribution 124 derived from recorded data associated with any number of past streaming sessions represented by the historical streaming session database 104 to perform numerical evaluations on each of the candidate encoding ladders included in the candidate encoding ladder set 158. Based on the results of the numerical evaluations, the numerical evaluation application 160 selects any number (including none) of the candidate encoding ladders from the candidate encoding ladder set 158 that represent sub-par tradeoffs between expected streaming QoE and storage footprint. The numerical evaluation application 160 then filters out (e.g., removes) any selected candidate encoding ladders from the candidate encoding ladder set 158 to generate a filtered candidate encoding ladder set 168.

[0054] As shown, in some embodiments, the simulation evaluation application 170 performs any number and/or types of simulation-based evaluations on the filtered candidate encoding ladder set 168 based on the encoded chunks metadata 138 and the historical streaming session database 104. For each of the candidate encoding ladders in the filtered candidate encoding ladder set 168, the simulation evaluation

application 170 performs T different simulations using the candidate encoding ladder and a streaming session trace 126(1) - a streaming session trace 126(T), where T can be any positive integer.

5 [0055] For explanatory purposes, the streaming session trace 126(1) - a streaming session trace 126(T) are also referred to herein individually as a “streaming session trace 126” and collectively as “streaming session traces 126.” Each of the streaming session traces 126 includes recorded measurements of one or more characteristics of a network over a period of time or synthesized measurements of one or more network characteristics over a period of time.

10 [0056] In some embodiments, each of the streaming session traces 126 is network throughput trace that indicates a network throughput as a function of time over the duration of the trace. In some embodiments, including the embodiment depicted in Figure 1, each of the streaming session traces 126 is a recorded network throughput trace that specifies recorded measurements of the throughput of an actual network.

15 In some alternative embodiments, each of the streaming session traces 126 is a synthesized network throughput trace that is generated by a software application (e.g., the simulation evaluation application 170) in any technically feasible fashion.

[0057] In some embodiments, to perform a simulation for a candidate encoding ladder using the streaming session trace 126(t), where t can be any integer from 1
20 through T, the simulation evaluation application 170 transmits the candidate encoding ladder and streaming session trace 126(t) to the adaptive streaming simulator 106. In response, the adaptive streaming simulator 106 executes an adaptive bitrate (ABR) algorithm based on the candidate encoding ladder over a simulated streaming session that is characterized by the streaming session trace 126(t). Over the
25 simulated streaming session, the ABR algorithm attempts to select a sequence of encoded chunks having the highest bitrates possible without exceeding the available network throughput.

[0058] For each of the simulations, the simulation evaluation application 170 computes a set of values for a set of metrics referred to herein as a “streaming
30 evaluation metric set.” The simulation evaluation application 170 performs any number and/or types of evaluations and/or comparisons between the sets of values to

select any number of the associated candidate encoding ladders for further evaluation and/or deployment via the production encoding pipeline 120.

5 [0059] In some embodiments, including the embodiment depicted in Figure 1, the simulation evaluation application 170 selects a single candidate encoding ladder for deployment via the production encoding pipeline 120. The simulation evaluation application 170 generates the encoding ladder 178 based on the selected candidate encoding ladder and then transmits the encoding ladder 178 to the ladder deployment application 190 included in the production encoding pipeline 120. The simulation evaluation application 170 is described in greater detail below in conjunction with
10 Figure 4.

[0060] Advantageously, because the encoding ladder application 150 formulates the encoding ladder generation problem as a constrained optimization problem, the encoding ladder application 150 concurrently accounts for different ladder constraints when generating the candidate encoding ladders included in the candidate encoding
15 ladder set 158. As a result, the encoding ladder application 150 automatically identifies and exploits opportunities to use a single encoded video that satisfies multiple different ladder constraints when generating each candidate encoding ladder. Relative to an encoding ladder for a media title generated using prior-art approaches, the encoding ladder application 150 can therefore generate a candidate encoding
20 ladder for the media title that is associated with a reduced storage footprint and the same or better streaming QoE. And, unlike prior-art techniques, each objective function explicitly represents a streaming QoE/storage footprint tradeoff. Relative to an encoding ladder for a media title generated using prior-art approaches, the encoding ladder application 150 can therefore generate a candidate encoding ladder
25 for a media title that is associated with the same or smaller storage footprint and a better streaming QoE.

[0061] As persons skilled in the art will recognize, typical prior-art approaches to comparing streaming QoEs achieved using different encoding ladders for a media title over different network conditions involve A/B testing. Usually, because A/B testing is
30 time-consuming and consumes significant amounts of processing and network resources, only a relatively small number of encoding ladders for a media title are compared using A/B testing. Advantageously, the simulation evaluation application 170 can efficiently compare the performance of a significantly larger number of

candidate encoding ladders and/or encoding ladders over a significantly larger number of network throughput traces using less time, processing resources, and network resources. Relative to prior-art techniques, because the simulation evaluation application 170 can more efficiently and widely evaluate a space of possible encoding ladders, the simulation evaluation application 170 can generate an encoding ladder representing a better streaming QoE/footprint tradeoff.

[0062] Note that the techniques described herein are illustrative rather than restrictive and may be altered without departing from the broader spirit and scope of the invention. Many modifications and variations on the functionality provided by the encoding ladder application 150, the numerical evaluation application 160, the simulation evaluation application 170, the encoding ladder workflow 140, the shot-based encoding application 132, the ladder deployment application 190, the production encoding pipeline 120, and the adaptive streaming simulator 106 will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. In some embodiments, the inventive concepts described herein in the context of the encoding ladder application 150 can be practiced without any of the other inventive concepts described herein. In some embodiments, the inventive concepts described herein in the context of the simulation evaluation application 170 can be practiced without any of the other inventive concepts described herein. In the same or other embodiments, the simulation evaluation application 170 can use one or more ABR algorithms to evaluate any number of candidate encoding ladders to use when streaming the media title associated with the source video 102.

[0063] In some embodiments, the encoding ladder application 150 can incorporate any number and/or types of objective functions, where each objective function attempts to maximize streaming QoE while minimizing one or more cost terms. As used herein, a “cost term” is a portion of an objective function that is to be reduced when solving a constrained optimization problem and is associated with (e.g., computed based on) any number and/or types of costs. In some embodiments, the footprint term 324 is a cost term of the objective function 340. A “cost” can be any characteristic, metric, etc., associated with transmitting encoded videos to a client device over any number and/or types of network connections. Two examples of costs are a storage footprint of an encoding ladder and network bandwidth consumption.

[0064] In some alternate embodiments, any amount (including none or all) of the convex hull metadata 136 and/or the encoded chunk metadata 138 can be derived from encoded videos and any remaining amount (including none or all) of the convex hull metadata 136 and/or the encoded chunk metadata 138 can be estimated for
5 “virtual” encoded videos, and the techniques described herein are modified accordingly. Metadata estimated for a “virtual encoded video” refers herein to metadata that is estimated for an encoded video that could potentially be generated based on the source video 102. Any amount and/or types of metadata can be estimated for virtual encoded videos in any technically feasible fashion. For instance,
10 in some embodiments, the production encoding pipeline 120 and/or the encoding ladder application 150 estimates metadata for a virtual encoded video based on a curve. In the same or other embodiments, the production encoding pipeline 120 and/or the encoding ladder application 150 performs any number and/or types of extrapolation operations and/or interpolation operations on metadata associated with
15 one or more encoded versions of the source video 102 to estimate metadata for a virtual encoded video that could potentially be generated from the source video 102. For explanatory purposes, as used herein, metadata of an encoded video can refer to either metadata derived from an encoded version of a video or metadata estimated for a virtual encoded video. For example, a bitrate and a quality score of an encoded
20 video can refer to a bitrate and a quality score that is estimated for a virtual encoded video based on the source video 102 and/or zero or more encoded versions of the source video 102.

[0065] Many modifications and variations on the organization, amount, and/or types of data described herein will be apparent to those of ordinary skill in the art
25 without departing from the scope and spirit of the described embodiments. For instance, in some embodiments, each rung of the encoding ladder 178 can specify a quality score in addition to the bitrate and the resolution of the encoded video specified via the encoded video ID.

[0066] It will be appreciated that the system 100 shown herein is illustrative and
30 that variations and modifications are possible. For instance, the connection topology between the various components in Figure 1 may be modified as desired. For example, in some embodiments, one or more of the production encoding pipeline 120, the numerical evaluation application 160, the simulation evaluation application

170, or the encoding ladder workflow 140 are omitted from the system 100, and the encoding ladder application 150 generates and/or obtains the encoded chunk metadata 138 in any technically feasible fashion.

[0067] Figure 2 is a more detailed illustration of the encoding ladder application 5 150 of Figure 1, according to various embodiments. As described previously herein in conjunction with Figure 1, the encoding ladder application 150 generates a candidate encoding ladder set 158 based on the convex hull metadata 136. The candidate encoding ladder set 158 includes a candidate encoding ladder 280(1) - a candidate encoding ladder 280(C), where C can be any positive integer. Each of the candidate 10 encoding ladder 280(1) - the candidate encoding ladder 280(C) can be used as an encoding ladder for the media title associated with the convex hull metadata 136.

[0068] As shown, the convex hull metadata 136 includes, without limitation, a 15 bitrate-quality point 210(1) - a bitrate-quality point 210(M), where M can be any positive integer. As described previously herein in conjunction with Figure 1, each of the bitrate-quality point 210(1) - the bitrate-quality point 210(M) represents a different encoded video that is generated based on the shots included in the source video 102. For explanatory purposes, the bitrate-quality point 210(1) - a bitrate-quality point 210(M) are also referred to herein individually as a "bitrate-quality point 210" and collectively as "bitrate-quality points 210."

20 [0069] As shown, the bitrate-quality point 210(1) includes, without limitation, an encoded video ID 212(1), a resolution 214(1), a bitrate 216(1), and a quality score 218(1). The encoded video ID 212(1) identifies an encoded video that is generated or can be generated based on the shots included in the source video 102. If the encoded video ID 212(1) identifies an encoded video that is already generated, then 25 the resolution 214(1), the bitrate 216(1), and the quality score 218(1) specify the resolution, the bitrate, and the quality score (e.g., a value for a VMAF metric) of the encoded video. Otherwise, the resolution, the bitrate 216(1), and the quality score 218(1) specify the resolution, an estimated average bitrate, and an estimated quality score for an encoded video that can be generated as per the encoded video ID 30 212(1). As depicted in *italics*, the resolution 214(1), the bitrate 216(1), and the quality score 218(1) are also symbolized herein as R_1 , B_1 , and Q_1 , respectively.

[0070] Although not shown, for an integer x from 2 through M , the bitrate-quality point 210(x) includes, without limitation, an encoded video ID 212(x), a resolution 214(x), a bitrate 216(x), and a quality score 218(x). The encoded video ID 212(x) identifies an encoded video that is generated or can be generated based on the shots
5 included in the source video 102. If the encoded video ID 212(x) identifies an encoded video that is already generated, then the resolution 214(x), the bitrate 216(x), and the quality score 218(x) specify the resolution, the average bitrate, and a visual quality score for the encoded video. Otherwise, the resolution 214(x), the bitrate 216(x), and the quality score 218(x), specify the resolution, an estimated
10 average bitrate, and an estimated visual quality score for an encoded video that can be generated as per the encoded video ID 212(x). For explanatory purposes, the resolution 214(x), the bitrate 216(x), and the quality score 218(x) are symbolized herein as R_x , B_x , and Q_x , respectively.

[0071] The encoding ladder application 150 formulates the problem of generating
15 the candidate encoding ladder 280(1) - the candidate encoding ladder 280(C) as an parameterized constrained optimization problem. As used herein a "parameterized" constrained optimization problem is a generalized version of a constrained optimization problem that is expressed in terms of at least one parameter. Each parameter associated with a constrained optimization problem is represented with a
20 symbol. Different combinations of value(s) of the parameter(s) that are associated with the parameterized constrained optimization problem can be used to formulate different constrained optimization problems. While solving a constrained optimization problem, the value(s) of the parameter(s) do not change. A value for a parameter is also referred to herein as a "parameter value." Constrained optimization, constrained
25 optimization problems, and techniques for solving constrained optimization problems are well-known in the art. Please see https://en.wikipedia.org/wiki/Constrained_optimization for an overview.

[0072] As shown, the encoding ladder application 150 includes, without limitation, a normalization engine 220, encoding point metadata 230, a parameterized objective
30 function 240, a parameterized constraint set 250, a ladder configuration 260(1) - a ladder configuration 260(C), a rung assignment engine 270(1) - a rung assignment engine 270(C), the candidate encoding ladder 280(1) - the candidate encoding ladder 280(C), and the candidate encoding ladder set 158.

[0073] As persons skilled in the art will recognize, bitrates and visual quality scores associated with encoded videos usually, if not always, have different magnitudes. In some embodiments, to facilitate formulating encoding ladder generation as a parameterized constraint optimization problem, the normalization engine 220
 5 normalizes the bitrate 216(1) - the bitrate 216(M) and the quality score 218(1) - the quality score 218(M) to a common range.

[0074] More precisely, in some embodiments, the normalization engine 220 log transforms the bitrate 216(1) - the bitrate 216(M) to generate log transformed bitrates (not shown). The normalization engine 220 applies normalization to the log
 10 transformed bitrates and the quality score 218(1) - the quality score 218(M) to generate the normalized bitrate array 236 and the normalized quality score array 238, respectively. The normalized bitrates in the normalized bitrate array 236 are derived from $B_1 - B_M$ and are symbolized herein as $B'_1 - B'_M$, respectively. The normalized quality scores in the normalized quality score array 238 correspond to $Q_1 - Q_M$ and are
 15 symbolized herein as $Q'_1 - Q'_M$, respectively.

[0075] In some alternate embodiments, the normalization engine 220 and/or the encoding ladder application 150 can perform any number (including none) and/or types of normalization operations on the bitrate 216(1) - the bitrate 216(M) and/or the quality score 218(1) - the quality score 218(M) in any technically feasible fashion, and
 20 the techniques described herein are modified accordingly. For example, in some alternate embodiments, the normalization engine 220 normalizes the bitrate 216(1) - the bitrate 216(M) but not the quality score 218(1) - the quality score 218(M) or the quality score 218(1) - the quality score 218(M) but not the bitrate 216(1) - the bitrate 216(M). In some embodiments, the encoding ladder application 150 omits the
 25 normalization engine 220.

[0076] The encoding point metadata 230 includes arrays of constants that are specified in or derived from the convex hull metadata 136 and are used to formulate the parameterized constrained optimization problem. As shown, the encoding point metadata 230 includes, without limitation, an ID array 222, a resolution array 224, a
 30 bitrate array 226, a quality score array 228, the normalized bitrate array 236, and the normalized quality score array 238. The encoding ladder application 150 re-organizes the convex hull metadata 136 to generate the ID array 222, the resolution array 224, the bitrate array 226, and the quality score array 228, The ID array 222

includes the encoded video ID 212(1) - the encoded video ID 212(M). The resolution array 224 includes the resolution 214(1) - the resolution 214(M) that are symbolized as $R_1 - R_M$, respectively. The bitrate array 226 includes the bitrate 216(1) - the bitrate 216(M) that are symbolized as $B_1 - B_M$, respectively. The quality score array 228
5 includes the quality score 218(1) - quality score 218(M) that are symbolized as $Q_1 - Q_M$, respectively. Note that the total number of bitrate-quality points or "bitrate-quality point count" that is symbolized herein as M is a constant in the context of the parameterized constraint optimization problem.

[0077] The encoding ladder application 150 defines the parameterized objective
10 function 240 based on an overall objective of determining N rung assignments for a candidate encoding ladder that collectively optimize a tradeoff between increasing the streaming QoE of the candidate encoding ladder and reducing the storage footprint of the candidate encoding ladder. As used herein, N symbolizes a rung count that is a parameter specifying the total number of rungs of the candidate encoding ladder. A
15 "rung count" is also referred to herein as a "number of rungs." Each rung assignment indicates that a different one of the bitrate-quality points 210 is assigned to a different rung of the candidate encoding ladder.

[0078] As described in greater detail below in conjunction with the parameterized constraint set 250, the rung assignments are constrained such that bitrates specified
20 via the rung 1 through the rung N increase monotonically. For explanatory purposes, rung 1 and rung N of a candidate encoding ladder are also referred to herein as a "lowest rung" and a "highest rung" of the candidate encoding ladder.

[0079] In general, the "streaming QoE" of an encoding ladder quantifies an average QoE of viewers of a media title that has been encoded and streamed to client
25 devices. A typical QoE metric reflects both visual quality levels associated with the encoded chunks used for streaming and the impact of any re-buffering events on the overall quality of the viewing experience. As persons skilled in the art will recognize, the streaming QoE associated with an encoding ladder cannot be accurately measured until the encoding ladder is deployed. Accordingly, the encoding ladder
30 application 150 defines a quality term (not shown in Figure 2) that correlates with the streaming QoE.

[0080] The quality term is a weighted average of the normalized quality scores specified in the N bitrate-quality points that are assigned to the N rungs of a candidate encoding ladder. The quality term is associated with a different “rung quality” weight for each of the N rungs, where the N rung quality weights are parameters of the parameterized constrained optimization problem. A rung quality weight associated with the rung j is symbolized herein as w_j . The values of the rung quality weights can be determined in any technically feasible fashion. In some embodiments, the values of the rung quality weights are predetermined. In some other embodiments, the values of the rung quality weights are derived from empirical statistics, such as network throughput and/or bitrate demand distributions.

[0081] As persons skilled in the art will recognize, the sum of the bitrates of the encoded videos included in an encoding ladder is proportional to the storage footprint of the encoding ladder. Accordingly, the encoding ladder application 150 defines a footprint term (not shown in Figure 2) as the sum of the normalized bitrates of the N encoded videos that are assigned to the N rungs of a candidate encoding ladder.

[0082] An overall objective of maximizing the streaming QoE of a candidate encoding ladder while minimizing the storage footprint of the candidate encoding ladder inherently represents a tradeoff between the streaming QoE and the storage footprint. To explicitly capture the tradeoff between the streaming QoE and the storage footprint in a flexible fashion, the encoding ladder application 150 weights the quality term and the footprint term by a quality term weight and a footprint term weight, respectively. The parameterized objective function 240 therefore represents a weighted tradeoff between a weighted average of a subset of a set of normalized quality scores associated with a set of encoded videos and a sum of a subset of a set of normalized bitrates associated with the set of encoded videos. More precisely, in some embodiments, the parameterized objective function 240 represents a weighted tradeoff between a weighted average of a subset of the normalized quality score array 238 corresponding to a subset of the ID array 222 and a sum of a subset of the normalized bitrate array 236 corresponding to the same subset of the ID array 222.

[0083] The quality term weight and the footprint term weight are parameters of the parameterized constrained optimization problem. Values of the quality term weight and the footprint term weight reflect the relative importance of maximizing the

streaming QoE of a candidate encoding ladder and the relative importance of minimizing the storage footprint of the candidate encoding ladder, respectively.

[0084] As described in greater detail below in conjunction with Figure 3, to facilitate expressing rung assignments via the parameterized objective function 240, the encoding ladder application 150 defines an assignment matrix. The assignment matrix specifies multiple assignments between the set of encoded videos corresponding to the ID array 222 and a set of rungs that are to be included in a candidate encoding ladder. More specifically, the assignment matrix is an MxN Boolean matrix that is symbolized herein as X . The rows 1 - M of X correspond to the bitrate-quality point 210(1) - the bitrate-quality point 210(M), respectively. The columns of X correspond to a rung 1 - a rung N of the candidate encoding ladder 280(1).

[0085] The symbol $X_{i,j}$ denotes the element in the i^{th} row and the j^{th} column of the assignment matrix 370(1). If $X_{i,j}$ is 1, then the bitrate-quality point 210(i) is assigned to the rung j of a candidate encoding ladder. If $X_{i,j}$ is 0, then the bitrate-quality point 210(i) is not assigned to the rung j of the candidate encoding ladder. As referred to herein, if the bitrate-quality point 210(i) is assigned to a rung j of a candidate encoding ladder, then the encoded video corresponding to the encoded video ID 212(i), the resolution 214(i), the bitrate 216(i), and the quality score 218(i) are also assigned to the rung j . Further, if the bitrate-quality point 210(i) is assigned to a rung j of a candidate encoding ladder, then the rung j is referred to herein as “specifying” the encoded video corresponding to the encoded video ID 212(i), the encoded video ID 212(i), the resolution 214(i), the bitrate 216(i), and the quality score 218(i).

[0086] Although not shown, in some embodiments, the encoding ladder application 150 implements the parameterized objective function 240 as follows:

$$\max \left(q \sum_{i=1}^M \sum_{j=1}^N X_{i,j} {}'_i w_j - b \sum_{i=1}^M \sum_{j=1}^N X_{i,j} B'_i \right) \quad (1)$$

In equation (1), the first term is the quality term, the second term is the footprint term, and the symbols q and b denote the quality term weight and the footprint term weight, respectively.

[0087] The encoding ladder application 150 defines values for the rung quality weights $w_1 - w_N$ using the following equations:

$$\sum_{j=1}^N w_j = 1 \quad (2a)$$

where

$$w_j = \alpha \sum_{k=1}^{j-1} w_k, \forall j \in \{2..N\} \quad (2b)$$

In equation (2b), the symbol α denotes a weight generation parameter that is associated with the overall optimization problem.

[0088] Assigning higher values to rung quality weights associated with higher rungs as per equation (2b) emulates historical streaming statistics indicating that, for a given encoding ladder, the frequency with which each of the rungs is actually selected for streaming by client devices typically increases as the corresponding quality score increases. Consequently, the quality term in equation (1) approximates an average visual quality level of the different instances of encoded videos specified in a candidate encoding ladder that are predicted to be streamed to client devices.

[0089] In some other embodiments, the values of the rung quality weights are set to 1 and the quality term in equation (1) represents an average visual quality level of the encoded videos specified in the candidate encoding ladder. More generally, as part of generating an objective function, a rung assignment engine can compute any number of values for any number of weights that are referenced by the objective function and are associated with a set of rungs based on a ladder configuration.

[0090] As shown, the encoding ladder application 150 generates the parameterized constraint set 250. The parameterized constraint set 250 includes any number and/or types of constraints, where each constraint restricts one or more rung assignments and can be associated with zero or more parameters. In some embodiments, the parameterized constraint set 250 includes implicit logical constraints and practical logical constraints. The implicit logical constraints ensure that the set of rung assignments specified via a final “optimized” assignment matrix corresponds to a valid encoding ladder. The practical logical constraints capture

restrictions and/or preferences that are associated with any operational aspects of streaming videos. Some examples of operational aspects of streaming videos include the capabilities of client devices, network capacity, a CDN, human perception of visual quality, and the like.

- 5 **[0091]** In some embodiments, the implicit logical constraints include, without limitation, a parameterized rung assignment constraint, a parameterized point assignment constraint, and a parameterized monotonically increasing bitrate constraint. The practical logical constraints include, without limitation, a
 10 parameterized footprint upper bound constraint, a parameterized required resolution constraint, a parameterized low bitrate point constraint, a parameterized high quality point constraint, a parameterized minimum quality spacing constraint, a parameterized maximum quality spacing constraint, and a parameterized bitrate spacing constraint.

- [0092]** The parameterized rung assignment constraint is that exactly one of the
 15 bitrate-quality points 210 is assigned to each rung of a candidate encoding ladder. The parameterized rung assignment constraint can be expressed as follows:

$$\sum_{i=1}^M X_{i,j} = 1, \forall j \in \{1..N\} \quad (3)$$

- [0093]** The parameterized point assignment constraint is that each of the bitrate-quality points 210 is assigned to at most one rung of a candidate encoding ladder. The parameterized point assignment constraint can be expressed as follows:

$$\sum_{j=1}^N X_{i,j} \leq 1, \forall i \in \{1..M\} \quad (4)$$

- 20 **[0094]** The parameterized monotonically increasing bitrate constraint is that the bitrate increases monotonically between rungs. The parameterized monotonically increasing bitrate constraint can be expressed as follows:

$$\sum_{i=1}^M X_{i,j} B_i \leq \sum_{i=1}^M X_{i,j+1} B_i, \forall j \in \{1..N-1\} \quad (5)$$

In accordance with equation (5), rung 1 and rung N of a candidate encoding ladder correspond to the lowest bitrate and the highest bitrate, respectively of the encoding ladder.

- [0095]** Because the storage resources of CDNs are limited, the encoding ladder application 150 implements the parameterized footprint upper bound constraint. The parameterized footprint upper bound constraint is that the sum of the bitrates of the encoded videos assigned to the rungs of a candidate encoding ladder is less than or equal to a footprint upper bound. The parameterized footprint upper bound constraint can be expressed as follows:

$$\sum_{i=1}^M \sum_{j=1}^N X_{i,j} B_i \leq \text{footprint upper bound} \quad (6)$$

- [0096]** Because the screens of different client devices can have different resolutions, the encoding ladder application 150 implements one or more constraints associated with at least one resolution that the candidate encoding ladder is required to represent. Although not shown in Figure 2, the encoding ladder application 150 implements a different required resolution constraint for each of zero or more required resolutions based on the parameterized required resolution constraint. The parameterized required resolution constraint is that at least one of the bitrate-quality points 210 assigned to the rungs of a candidate encoding ladder specifies a required resolution. The parameterized required resolution constraint can be expressed as follows:

$$\sum_{k \in A} \sum_{j=1}^N X_{k,j} \geq 1 \quad (7)$$

- In equation (7), A denotes the subset of $i \in \{1..M\}$ for which R_i is equal to a “required resolution.”

- [0097]** In an effort to ensure uninterrupted playback of a media title under challenging network conditions, the encoding ladder application 150 implements the parameterized low bitrate point constraint. The parameterized low bitrate point is that at least one of the bitrate-quality points 210 assigned to the rungs of a candidate

encoding ladder specifies a bitrate that is less than or equal to a “low bitrate.” The parameterized low bitrate point constraint can be expressed as follows:

$$\sum_{k \in D} \sum_{j=1}^N X_{k,j} \geq 1 \quad (8)$$

In equation (8), D denotes the subset of $i \in \{1..M\}$ for which B_i is less than or equal to a “low bitrate.”

- 5 **[0098]** In an effort to ensure that a sufficiently high visual quality is perceived by viewers when streaming a media title over a connection having a relatively high network capacity, the encoding ladder application 150 implements the parameterized high quality point constraint. The parameterized high quality point constraint is that at least one of the bitrate-quality points 210 assigned to the rungs of a candidate
- 10 encoding ladder specifies a quality score that is greater than or equal to a “high quality score.” The parameterized high quality point constraint can be expressed as follows:

$$\sum_{k \in E} \sum_{j=1}^N X_{k,j} \geq 1 \quad (9)$$

In equation (9), E denotes the subset of $i \in \{1..M\}$ for which Q_i is greater than or equal to a “high quality score.”

- 15 **[0099]** A parameterized minimum quality spacing constraint is that consecutive rungs of a candidate encoding ladder specify quality scores that are separated by at least a minimum quality spacing. The parameterized minimum quality spacing constraint can be expressed as follows:

$$\sum_{i=1}^M X_{i,j+1} Q_i - \sum_{i=1}^M X_{i,j} Q_i \geq \Delta Q_{\min}, \forall j \in \{1..N-1\} \quad (10)$$

In equation (10), $\Delta Q_{\min}$ symbolizes a minimum quality spacing.

- 20 **[0100]** A parameterized maximum quality spacing constraint is that consecutive rungs of a candidate encoding ladder specify quality scores that are separated by no more than a maximum quality spacing. The parameterized maximum quality spacing constraint can be expressed as follows:

$$\sum_{i=1}^M X_{i,j+1} Q_i - \sum_{i=1}^M X_{i,j} Q_i \leq \Delta Q_{\max}, \forall j \in \{1..N-1\} \quad (11)$$

In equation (11), $\Delta Q_{\max}$ symbolizes a maximum quality spacing

[0101] The parameterized bitrate spacing constraint is that consecutive rungs of a candidate encoding ladder specify bitrates that are separated by no more than a relative bitrate spacing. The parameterized bitrate spacing constraint can be

5 expressed as follows:

$$\sum_{i=1}^M X_{i,j+1} B_i \leq \lambda \sum_{i=1}^M X_{i,j} B_i, \forall j \in \{1..N-1\} \quad (12)$$

In equation (12), λ symbolizes a relative bitrate spacing.

[0102] Each of the ladder configuration 260(1) - the ladder configuration 260(C) specifies, without limitation, a different set of values for the set of parameters associated with the parameterized constraint optimization problem. The encoding ladder application 150 can generate the ladder configuration 260(1) - the ladder configuration 260(C) in any technically feasible fashion. More generally, the encoding ladder application 150 can determine different sets of parameter values for a union of a set of parameters included in the parameterized objective function 240 and a set of parameters included in the parameterized constraint set 250 to generate the ladder the ladder configuration 260(1) - the ladder configuration 260(C),

[0103] As shown, the ladder configuration 260(1) includes, without limitation, a rung count 262(1), a rung quality weight set 264(1), an objective parameter set 266(1), and a constraint parameter set 268(1). Although not shown, for an integer x from 2 through C, the ladder configuration 260(x) includes, without limitation, a rung count 262(x), a rung quality weight set 264(x), an objective parameter set 266(x), and a constraint parameter set 268(x).

[0104] Each of the rung count 262(1) - the rung count 262(C) specifies a value for the rung count. Each of the rung quality weight set 264(1) - the rung quality weight set 264(C) specifies a different value for each of the rung quality weights. In some embodiments, the encoding ladder application 150 computes the rung quality weight

set 264(1) - the rung quality weight set 264(C) based on the equations (2a) and (2b) and the rung count 262(1) - the rung count 262(C), respectively.

[0105] Each of the objective parameter set 266(1) - the objective parameter set 266(C) specifies values for the quality term weight and the bitrate term weight. Each
 5 of the constraint parameter set 268(1) - the constraint parameter set 268(C) specifies zero or more required resolutions and values for the footprint upper bound, the high quality score, the low bitrate, the minimum quality spacing, the maximum quality spacing, and the relative bitrate spacing.

[0106] The rung assignment engine 270(1) - the rung assignment engine 270(C)
 10 are different instances of a single software application referred to herein as the "rung assignment engine." As shown, the encoding ladder application 150 executes the rung assignment engine 270(1) on the ladder configuration 260(1), the encoding point metadata 230, the parameterized objective function 240, and the parameterized constraint set 250 to generate the candidate encoding ladder 280(1). As also shown,
 15 the encoding ladder application 150 executes the rung assignment engine 270(C) on the ladder configuration 260(C), the encoding point metadata 230, the parameterized objective function 240, and the parameterized constraint set 250 to generate the candidate encoding ladder 280(C). Although not shown, for an integer x from 2 through $(C-1)$, the encoding ladder application 150 executes the rung assignment
 20 engine 270(x) on the ladder configuration 260(x), the encoding point metadata 230, the parameterized objective function 240, and the parameterized constraint set 250 to generate the candidate encoding ladder 280(x).

[0107] In general, the rung assignment engine uses the parameter values specified in a ladder configuration to derive an objective function from the
 25 parameterized objective function 240 and constraints from the parameterized constraint set 250. The rung assignment engine implements any number and/or types of constraint optimization techniques in an attempt to determine values for the elements of the assignment matrix that optimize the objective function subject to the constraints. Attempting to determine values for the assignment matrix that optimize
 30 the objective function subject to the constraints is also referred to herein as "solving" the constrained optimization problem defined by the objective function and the constraints. The rung assignment engine 270(1) is described in greater detail below in conjunction with Figure 3.

[0108] Many modifications and variations on the functionality of the encoding ladder application 150, the parameterized objective function 240, the parameterized constraint set 250, the ladder configuration 260(1) - the ladder configuration 260(C), and the rung assignment engine as described herein will be apparent to those of
5 ordinary skill in the art without departing from the scope and spirit of the described embodiments. In some embodiments, the encoding ladder application 150 can implement any number and/or types of objective functions and/or parameterized objective functions instead of or in addition to the parameterized objective function 240. In the same or other embodiments, the encoding ladder application 150 can
10 generate any number of candidate encoding ladders for each of any number of objective functions and/or parameterized objective functions. In some embodiments, the number and/or types of parameters associated with the parameterized objective function 240 and/or the parameterized constraint set 250 can vary. In the same or other embodiments, the number and/or types of parameterized constraints included in
15 the parameterized constraint set 250 can vary.

[0109] Figure 3 is a more detailed illustration of one of the rung assignment engines of Figure 2, according to various embodiments. More specifically, Figure 3 depicts the rung assignment engine 270(1) that generates the candidate encoding ladder 280(1) based on the ladder configuration 260(1), the encoding point metadata
20 230, the parameterized objective function 240, and the parameterized constraint set 250.

[0110] As described previously herein in conjunction with Figure 2, the ladder configuration 260(1) includes, without limitation, the rung count 262(1), the rung quality weight set 264(1), the objective parameter set 266(1), and the constraint
25 parameter set 268(1). In the embodiment depicted in Figure 3, the rung count 262(1) is six, and therefore the rung quality weight set 264(1) specifies values for six rung quality weights that are symbolized as $w_1 - w_6$.

[0111] For explanatory purposes, the objective parameter set 266(1) specifies values of 0.4 and 0.6 for the quality term weight and the footprint term weight,
30 respectively. And the constraint parameter set 268(1) specifies two required resolutions (e.g., 1280x720 and 1920x1080), values for the footprint upper bound, the high quality score, the low bitrate, the minimum quality spacing, and the maximum quality spacing, and a value of 1.7 for the relative bitrate spacing.

[0112] As described previously herein in conjunction with Figure 2, the encoding point metadata 230 are arrays of constants that are specified in or derived from the convex hull metadata 136 and are used to formulate the parameterized constrained optimization problem. More specifically, the encoding point metadata 230 includes, without limitation, the resolution array 224, the bitrate array 226, the quality score array 228, the normalized bitrate array 236, and the normalized quality score array 238. As noted previously herein, the resolution array 224, the bitrate array 226, the quality score array 228, the normalized bitrate array 236, and the normalized quality score array 238 include $R_1 - R_M$, $B_1 - B_M$, $Q_1 - Q_M$, $B'_1 - B'_M$, and $Q'_1 - Q'_M$, respectively.

[0113] As shown, the rung assignment engine 270(1) includes, without limitation, an objective function 340, a constraint 350(1) - a constraint 350(11), a constrained optimization solver 360, and an assignment matrix 370. The rung assignment engine 270(1) generates the objective function 340 based on the rung count 262(1) and the objective parameter set 266(1). The rung assignment engine 270(1) generates the constraint 350(1) - the constraint 350(11) based on the ladder configuration 260(1) and the parameterized constraint set 250. The constrained optimization solver 360 implements any number and/or types of constrained optimization techniques in an attempt to determine values for the elements of the assignment matrix 370 that optimize the objective function 340 subject to the constraint 350(1) - the constraint 350(11) and for the ladder configuration 260(1) and the encoding point metadata 230.

[0114] As shown, the assignment matrix 370(1) is an $M \times 6$ Boolean matrix symbolized as X . The rows 1 - M of X correspond to the bitrate-quality point 210(1) - the bitrate-quality point 210(M), respectively. The columns of X correspond to a rung 1 - a rung 6 of the candidate encoding ladder 280(1). After the constrained optimization solver 360 determines final values for the candidate encoding ladder 280(1), the rung assignment engine 270(1) generates the rungs of the candidate encoding ladder 280(1) based on the entries of 1 in the assignment matrix 370(1). If the entry $X_{i,j}$ is 1, then the rung assignment engine 270(1) generates the rung j of the candidate encoding ladder 280(1) based on the bitrate-quality point 210(i).

[0115] As described previously herein in conjunction with Figure 2, the parameterized objective function 240 can be expressed as equation (1). The rung assignment engine 270(1) replaces the parameters represented by the symbols N , q , and b in equation (1) with the rung count 262(1) of 6, the quality term weight of 0.4,

and the footprint term weight of 0.6 to generate the objective function 340 that can be expressed as follows:

$$\max \left(0.4 \sum_{i=1}^M \sum_{j=1}^6 X_{i,j} w_j - 0.6 \sum_{i=1}^M \sum_{j=1}^6 X_{i,j} B'_i \right) \quad (13)$$

In equation (13), the values for $w_1 - w_6$ are specified in the rung quality weight set 264(1).

- 5 **[0116]** As depicted, the following portion of equation (13) expresses a quality term 322 as a rung-weighted average of the normalized quality scores associated with a subset of the bitrate-quality points 210 that are assigned to the rungs of the candidate encoding ladder 280(1):

$$\sum_{i=1}^M \sum_{j=1}^6 X_{i,j} w_j \quad (13a)$$

- 10 **[0117]** And the following portion of equation (13) expresses a footprint term 324 as a sum of the normalized bitrates associated with the subset of the bitrate-quality points 210 that are assigned to the rungs of the candidate encoding ladder 280(1):

$$\sum_{i=1}^M \sum_{j=1}^6 X_{i,j} B'_i \quad (13b)$$

- [0118]** The parameter values of 0.4 and 0.6 for the parameters of the quality term weight and the footprint term weight, respectively, indicate a relative importance of the QoE and the storage footprint associated with the objective function 340 and
 15 therefore the candidate encoding ladder 280(1). More specifically, the tradeoff between QoE and storage footprint represented by the objective function 340 favors decreasing the storage footprint of the candidate encoding ladder 280(1) at the expense of increasing the QoE of the candidate encoding ladder 280(1).

- [0119]** As described previously herein in conjunction with Figure 2, the
 20 parameterized constraint set 250 includes the parameterized footprint upper bound constraint, the parameterized required resolution constraint, the parameterized low bitrate point constraint, the parameterized high quality point constraint. the

parameterized minimum quality spacing constraint, the parameterized maximum quality spacing constraint, and the parameterized bitrate spacing constraint.

[0120] The rung assignment engine 270(1) generates the constraint 350(1) - a constraint 350(11), based on the rung count 262(1), the constraint parameter set 268(1), and the parameterized constraint set 250. The constraint 350(1) - the constraint 350(11) are a rung assignment constraint, a point assignment constraint, a monotonically increasing bitrate constraint, a footprint upper bound constraint, two required resolution constraints, a low bitrate point constraint, a high quality point constraint, a minimum quality spacing constraint, a maximum quality spacing constraint, and a bitrate spacing constraint, respectively. For explanatory purposes, Figure 3 depicts the constraint 350(1) and the constraint 350(11) in more detail.

[0121] The constraint 350(1) is a rung assignment constraint that the rung assignment engine 270(1) derives from the parameterized rung assignment constraint described previously herein in conjunction with Figure 2. The rung assignment engine 270(1) replaces N in equation (3) with the rung count 262(1) of 6 to generate the constraint 350(1). The constraint 350(1) is that exactly one of the bitrate-quality points 210 is assigned to each of the six rungs of the candidate encoding ladder 280(1). As shown, the constraint 350(1) can be expressed as follows:

$$\sum_{i=1}^M X_{i,j} = 1, \forall j \in \{1..6\} \quad (14)$$

[0122] The constraint 350(11) is a bitrate spacing constraint that the rung assignment engine 270(1) derives from the parameterized bitrate spacing constraint described previously herein in conjunction with Figure 2. The rung assignment engine 270(1) replaces (N - 1) and λ in equation (12), with 5 (one less than the rung count 262(1) of 6) and 1.7, respectively, to generate the constraint 350(11). The constraint 350(11) is that consecutive rungs of the candidate encoding ladder 280(1) specify bitrates that are separated by no more than twice the bitrate of the lower rung. As shown, the constraint 350(11) can be expressed as follows:

$$\sum_{i=1}^M X_{i,j+1} B_i \leq 1.7 \sum_{i=1}^M X_{i,j} B_i, \forall j \in \{1..5\} \quad (15)$$

[0123] As shown, the rung assignment engine 270(1) causes the constrained optimization solver 360 to solve the objective function 340 subject to the constraint 350(1) - the constraint 350(11) based on the ladder configuration 260(1) and the encoding point metadata 230 to generate a final version of the assignment matrix

5 370. As shown, the M rows of X lie along a point axis 372 and the 6 columns of X lie along a rung axis 374.

[0124] As per the point axis 372, the top row of the assignment matrix 370 corresponds to the point index of 1 and therefore the bitrate-quality point 210(1), the encoded video ID 212(1), R_1 , B_1 , Q_1 , B'_1 , and Q'_1 . The bottom row of the assignment matrix 370 corresponds to the point index of M and therefore the bitrate-quality point 210(M), the encoded video ID 212(M), R_M , B_M , Q_M , B'_M , and Q'_M . As per the rung axis 374, the leftmost column and the rightmost column of the assignment matrix 370 correspond to a rung 1 and a rung 6, respectively, that are the lowest rung and the highest rung, respectively, of the candidate encoding ladder 280(1).

15 **[0125]** In some embodiments, the constrained optimization solver 360 executes a constrained optimization algorithm on the objective function 340, the constraint 350(1) - the constraint 350(11), the rung quality weight set 264(1), and the encoding point metadata 230 to generate the candidate encoding ladder 280(1). More generally, the constrained optimization solver 360 can execute any number and/or types of
20 constrained optimization algorithms to solve the objective function 340 subject to the constraint 350(1) - the constraint 350(11) and using the values specified in the rung quality weight set 264(1) and the encoding point metadata 230. As used herein, a constrained optimization algorithm is any algorithm that implements any number and/or types of constrained optimization techniques as known in the art.

25 **[0126]** For example, the constrained optimization solver 360 can execute a genetic algorithm that implements search-based optimization techniques. In another example, the constrained optimization solver 360 can execute a surrogate optimization algorithm that implements a surrogate model to approximate the objective function 340. Some examples of other constrained optimization techniques
30 that can be used to generate candidate encoding ladders include branch and bound techniques, cutting planes techniques, and surrogate model techniques. Constrained optimization solvers are well-known in the art. Please see <https://github.com/google/or-tools#readme> for an overview of a software suite known

as “Google Optimization Tools” that includes several different constrained optimization solvers.

[0127] As described previously herein, if $X_{i,j}$ is 1, then the bitrate-quality point 210(i) is assigned to the rung j . Otherwise, the bitrate-quality point 210(i) is not
 5 assigned to the rung j . Notably, as per constraint 350(3) (*i.e.*, the monotonically increasing bitrate constraint), for an integer x from 1 through $(N - 1)$, the bitrate specified in the bitrate-quality point 210 that is assigned to the rung x is less than the bitrate specified in the bitrate-quality point 210 that is assigned to the rung $(x + 1)$.

[0128] After the constrained optimization solver 380 has finished executing, the
 10 rung assignment engine 270(1) generates the candidate encoding ladder 280(1) based on the entries of 1 in the assignment matrix 370(1) and ignores the entries of 0 in the assignment matrix 370(1). More specifically, the rung assignment engine 270(1) generates the candidate encoding ladder 280(1) based on the six entries of 1 in the assignment matrix 370.

[0129] In some embodiments, if the entry $X_{i,j}$ is 1, then the rung assignment engine
 15 270(1) generates a rung j of the candidate encoding ladder 280(1) that specifies, without limitation, the encoded video ID 212(i), R_i , B_i , and optionally Q_i . Referring back to Figure 2, R_i is equal to the resolution 214(i), B_i is equal to the bitrate 216(i), and Q_i is equal to the quality score 218(i) associated with the encoded video
 20 corresponding to the encoded video ID 212(i).

[0130] For explanatory purposes, Figure 3 depicts the entries in the top five rows and the bottom five rows of the assignment matrix 370 at a point-in-time after the constrained optimization solver 380 has finished executing. As shown, $X_{4,1}$ and $X_{M-2,6}$ are both equal to 1. Because $X_{4,1}$ is equal to 1, the rung assignment engine 270(1)
 25 generates a rung 1 of the candidate encoding ladder 280(1) that specifies the encoded video ID 212(4), R_4 , B_4 , and optionally Q_4 . Because $X_{M-2,6}$ is equal to 1, the rung assignment engine 270(1) generates a rung 6 of the candidate encoding ladder 280(1) that specifies the encoded video ID 212($M-2$), R_{M-2} , B_{M-2} , and optionally Q_{M-2} . Although not shown, the rung assignment engine 270(1) also generates a rung 2 - a
 30 rung 5 of the candidate encoding ladder 280(1) based on the four other entries in the assignment matrix 370 that are equal to 1.

Simulation-Based Evaluation and Selection of Candidate Encoding Ladders

[0131] Figure 4 is a more detailed illustration of the simulation evaluation application 170 of Figure 1, according to various embodiments. The simulation evaluation application 170 generates the encoding ladder 178 based on the filtered candidate encoding ladder set 168, the encoded chunk metadata 138, and the streaming session traces 126. As shown, the simulation evaluation application 170 includes, without limitation, a streaming header synthesis engine 410, a synthetic streaming header 430(1) - a synthetic streaming header 430(C), and a ladder evaluation and selection engine 480.

10 [0132] The streaming header synthesis engine 410 generates each of the synthetic streaming header 430(1) - the synthetic streaming header 430(F) based on a different candidate encoding ladder in the filtered candidate encoding ladder set 168 and the encoded chunk metadata 138. As described previously herein in conjunction with Figure 1, the filtered candidate encoding ladder set 168 includes F of the candidate encoding ladder 280(1) - the candidate encoding ladder 280(C), where C can be any positive integer and F can be any positive integer that is less than or equal to C. Notably, each of the F candidate encoding ladders included in the filtered candidate encoding ladder set 168 includes a different set of rungs, where the number of rungs in each set of rungs can vary across the filtered candidate encoding ladder set 168. Furthermore, the encoded videos and the resolutions specified in the different sets of rungs can vary across the filtered candidate encoding ladder set 168. The encoded chunk metadata 138 specifies, without limitation, the encoded chunks of each encoded video specified (via corresponding encoded video IDs) in the convex hull metadata 136, the bitrates of the encoded chunks, and the quality scores of the encoded chunks.

[0133] Each of the synthetic streaming header 430(1) - the synthetic streaming header 430(F) includes a different streaming metadata set for each rung of the corresponding candidate encoding ladder. For explanatory purposes, Figure 4 depicts the synthetic streaming header 430(1) in detail. The streaming header synthesis engine 410 generates the synthetic streaming header 430(1) based on the candidate encoding ladder 280(1) (described previously herein in conjunction with Figure 3) and the encoded chunk metadata 138. More specifically, the streaming header synthesis engine 410 generates the synthetic streaming header 430(1) based

on a set of rungs included in the candidate encoding ladder 280(1) and the encoded chunk metadata 138. Referring back to the ladder configuration 260(1) depicted in Figure 3, the candidate encoding ladder 280(1) has six rungs where each rung specifies a different encoded video ID, the resolution of the encoded video
5 corresponding to the encoded video ID, and the bitrate of the encoded video corresponding to the encoded video ID.

[0134] As shown, the synthetic streaming header 430(1) includes a streaming metadata set 432(1) - a streaming metadata set 432(6) that describe per-chunk video rate and quality information corresponding to the rung 1 - the rung 6, respectively, of
10 the candidate encoding ladder 280(1). The streaming metadata set 432(1) includes, without limitation, an encoded video ID 434(1), encoded chunk IDs 436(1), a resolution 442(1), and a bitrate 444(1), encoded chunk bitrates 446(1), and encoded chunk quality scores 448(1). The encoded video ID 434(1) identifies an encoded video corresponding to the rung 1 of the candidate encoding ladder 280(1). The
15 resolution 442(1) and the bitrate 444(1) specify the resolution and the average bitrate, respectively, of the encoded video corresponding to the encoded video ID 434(1). The encoded chunk IDs 436(1) identify encoded chunks of the encoded video corresponding to the encoded video ID 434(1). The encoded chunk bitrates 446(1) and the encoded chunk quality scores 448(1) specify the bitrates and the quality
20 scores, respectively, of the encoded chunks corresponding to the encoded chunk IDs 436(1). Note that only metadata information is used to generate synthetic streaming headers. Accordingly, in some embodiments, the encoding ladder workflow 140 can generate synthetic streaming headers without requiring the production encoding pipeline 120 to generate corresponding encoded videos.

25 **[0135]** The ladder evaluation and selection engine 480 generates the encoding ladder 178 based on the synthetic streaming header 430(1) - synthetic streaming header 430(F) and the streaming session trace 126(1) - the streaming session trace 126(T). Each of the streaming session trace 126(1) - the streaming session trace 126(T) is a network throughput trace having a duration that is greater than or equal to
30 the playback time of the media title. As described previously herein in conjunction with Figure 1, a network throughput trace indicates a network throughput as a function of time over the duration of the trace.

[0136] In some embodiments, each of the streaming session trace 126(1) - the streaming session trace 126(T) is either a recorded network throughput trace or a synthetic network throughput trace. A recorded network throughput trace specifies recorded measurements of the throughput of an actual network. A synthetic network throughput trace is synthesized by a software application (e.g., the simulation evaluation application 170) in any technically feasible fashion.

[0137] As shown, the ladder evaluation and selection engine 480 executes the adaptive streaming simulator 106 on a simulation configuration 450(1,1) - a simulation configuration 450(F,T) to generate a request sequence 460(1,1) - a request sequence 460(F,T), respectively. The ladder evaluation and selection engine 480 therefore executes the adaptive streaming simulator 106 a total of (F*T) different times. The adaptive streaming simulator 106 emulates some of the behavior of an endpoint application executing on a client device during an adaptive streaming session. In particular, the adaptive streaming simulator 106 implements an ABR algorithm 406 that attempts to optimize the visual quality experienced during playback of a streamed media title while avoiding playback interruptions due to re-buffering events. In other words, the ABR algorithm 406 attempts to select a sequence of encoded chunks having the highest bitrates possible without exceeding the available network throughput. In some other embodiments, the adaptive streaming simulator 106 can implement any number and/or types of ABR algorithms that attempt to optimize the visual quality experienced during playback based on any number and/or types of criteria in any technically feasible fashion.

[0138] For explanatory purposes, an index f can be any integer from 1 through F and an index t can be any integer from 1 through T. For the simulation configuration 450(f,t), the adaptive streaming simulator 106 executes the ABR algorithm 406 on the synthetic streaming header 430(f) based on the streaming session trace 126(t) to generate the request sequence 460(f,t). More precisely, the adaptive streaming simulator 106 configures the ABR algorithm 406 to incrementally generate the request sequence 460(f,t) based on the synthetic streaming header 430(f) over a simulated streaming session that is characterized by the streaming session trace 126(t). Accordingly, the request sequence 460(f,t) is a sequence of requests for encoded chunks that the ABR algorithm 406 generates based on the synthetic streaming header 430(f) and a sequence of network throughputs included in the streaming

session trace 126(t). For each source chunk of the source video 102, the request sequence 460(f, t) therefore specifies a corresponding encoded chunk of one of the encoded videos specified in the rungs of the candidate encoding ladder corresponding to the synthetic streaming header 430(f).

- 5 **[0139]** The ladder evaluation and selection engine 480 generates a metric value set 470(1,1) - a metric value set 470(F,T) based on the request sequence 460(1,1) - a request sequence 460(F,T) and the synthetic streaming header 430(1) - synthetic streaming header 430(F). More precisely, the ladder evaluation and selection engine 480 generates the metric value set 470(f, t) based on the request sequence 460(f, t) and the synthetic streaming header 430(f). Each of the metric value set 470(1,1) - the metric value set 470(F,T) specifies a different set of values for a streaming evaluation metric set.

- 15 **[0140]** The streaming evaluation metric set can include, without limitation, any number and/or types of metrics that are relevant to streaming QoE and/or any number and/or types of associated costs (e.g., storage footprint, network bandwidth consumption). In some embodiments, the streaming evaluation metric set includes any QoE-related metrics typically measured during production A/B testing of encoding ladders. In the same or other embodiments, the metric value set 470(f, t) specifies a time-weighted quality score, an average playback bitrate, a total number of re-
20 buffering events, a total re-buffering time, a total number of rung switches, a frequency of rung switching, a weighted aggregation of any number of the previous metrics representing a streaming QoE, or any combination thereof associated with streaming the media title using the f^{th} candidate encoding ladder in the filtered candidate encoding ladder set 168. Some examples of quality scores are average
25 PSNR values, time-weighted values for the VMAF metric or "time-weighted VMAF scores," and average values for the VMAF metric or "average VMAF scores." In the same or other embodiments, the metric value set 470(f, t) specifies at least a metric value that approximates a tradeoff between streaming quality of experience and a storage footprint associated with the f^{th} candidate encoding ladder in the filtered
30 candidate encoding ladder set 168. In yet other embodiments, the metric value set 470(f, t) specifies at least a metric value that approximates a tradeoff between streaming quality of experience and a cost term associated with expected network bandwidth consumption for the f^{th} candidate encoding ladder in the filtered candidate

encoding ladder set 168 during an adaptive streaming session characterized by the streaming session trace 126(*t*).

5 [0141] The ladder evaluation and selection engine 480 can compute values for each metric included in the streaming evaluation set for any number of periods of time (e.g., an entire simulated streaming session) and/or repeatedly at any granularity (e.g., every minute).

[0142] For explanatory purposes, the metric value set 470(*f*, 1) - the metric value set 470(*f*, *t*) are also referred to herein as an *f*th "metric value set group." The *f*th metric value set group is associated with the synthetic streaming header 430(*f*) and therefore
10 the *f*th candidate encoding ladder in the filtered candidate encoding ladder set 168.

[0143] The ladder evaluation and selection engine 480 performs any number and/or types of evaluations and/or comparisons between the metric value set groups corresponding to any number of the candidate encoding ladders in the filtered candidate encoding ladder set 168 and optionally any other relevant data. Based, at
15 least in part, on the results of the evaluations and/or comparisons, the ladder evaluation and selection engine 480 can select any number of the associated candidate encoding ladders for further evaluation and/or deployment via the production encoding pipeline 120.

[0144] In some embodiments, the ladder evaluation and selection engine 480
20 computes an average streaming QoE for each candidate encoding ladder in the filtered candidate encoding ladder set 168 based, at least in part, on the corresponding metric value set group. The ladder evaluation and selection engine 480 then selects the candidate encoding ladder representing the highest average streaming QoE to storage footprint tradeoff for further evaluation and/or deployment
25 via the production encoding pipeline 120 (not shown in Figure 4).

[0145] In some other embodiments, the ladder evaluation and selection engine 480 performs one-on-one comparisons between pairs of candidate encoding ladders in the filtered candidate encoding ladder set 168 based, at least in part, on the corresponding metric value set groups. The ladder evaluation and selection engine
30 480 then selects any number of the candidate encoding ladders in the filtered candidate encoding ladder set 168 for further evaluation and/or deployment via the

production encoding pipeline 120 based, at least in part, on the results of the one-on-one comparisons.

[0146] For example, the ladder evaluation and selection engine 480 could compare two different metric values for the same metric that are specified in the metric value set 470(1,1) and the metric value set 470(2,1). Based on the result of the comparison, the ladder evaluation and selection engine 480 could determine that a first candidate encoding ladder associated with the metric value set 470(1,1) instead of a second candidate encoding ladder associated with the metric value set 470(2,1) should be used to stream the media title. The ladder evaluation and selection engine 480 could therefore select the first candidate encoding ladder for deployment via the production encoding pipeline 120.

[0147] The ladder evaluation and selection engine 480 generates the encoding ladder 178 based on the selected candidate encoding ladder. In some embodiments, the encoding ladder 178 is a copy of the selected candidate encoding ladder. In some other embodiments, the ladder evaluation and selection engine 480 performs one or more additional operations on the selected candidate encoding ladder to generate the encoding ladder 178. For instance, in some embodiments, the ladder evaluation and selection engine 480 configures the shot-based encoding application 132 to re-generate encoded videos for each rung of the selected candidate encoding ladder using encoding techniques that are computationally more intensive than the encoding techniques associated with the encoded chunk metadata 138. More specifically, relative to a given rung of the selected candidate encoding ladder, the corresponding re-generated encoded video usually has the same resolution, approximately the same bitrate, and a higher quality score.

[0148] The ladder evaluation and selection engine 480 stores the encoding ladder 178 in any type of memory that is accessible to the production encoding pipeline 120 and/or the encoding ladder workflow 140. In some embodiments, the ladder evaluation and selection engine 480 transmits the encoding ladder 178 to the production encoding pipeline 120, the encoding ladder workflow 140, any number of other software applications, or any combination thereof. In some embodiments, the encoding ladder 178 is a final encoding ladder that is used to stream the media title to one or more client devices over a network.

[0149] In some embodiments, the ladder evaluation and selection engine 480 can perform any number and/or types of evaluations and/or comparisons between the metric value set groups corresponding to any number of the candidate encoding ladders in the filtered candidate encoding ladder set 168 and optionally any other
5 relevant data. Based, at least in part, on the results of the evaluations and/or comparisons, the ladder evaluation and selection engine 480 can select any number of the associated candidate encoding ladders for further evaluation and/or deployment via the production encoding pipeline 120.

[0150] In some embodiments, the ladder evaluation and selection engine 480
10 computes an average streaming QoE for each candidate encoding ladder in the filtered candidate encoding ladder set 168 based, at least in part, on the corresponding metric value set group. The ladder evaluation and selection engine 480 then selects the candidate encoding ladder representing the highest average streaming QoE to storage footprint tradeoff for further evaluation and/or deployment
15 via the production encoding pipeline 120 (not shown in Figure 4).

[0151] Again, note that the techniques described herein are illustrative rather than restrictive and may be altered without departing from the broader spirit and scope of the invention. For instance, in some embodiments the functionality of the simulation evaluation application 170, the adaptive streaming simulator 106, and the ABR
20 algorithm 406 as described herein can be consolidated into a single software application or distributed across any number of software applications in any technically feasible fashion. In the same or other embodiments, the simulation evaluation application 170 can perform simulation evaluations of any number of candidate encoding ladders and/or any number of encoding ladders (e.g., a current
25 production encoding ladder) using any number and/or types of ABR algorithms and any number and/or types of network throughput traces in any technically feasible fashion.

[0152] Figure 5 is a flow diagram of method steps for generating candidate encoding ladders for use when streaming a media title, according to various
30 embodiments. Although the method steps are described with reference to the systems of Figures 1 - 3, persons skilled in the art will understand that any system configured to implement the method steps, in any order, falls within the scope of the embodiments.

[0153] As shown, a method 500 begins at step 502, where the shot-based encoding application 132 partitions a source video 102 corresponding to a media title into shots. At step 504, the shot-based encoding application 132 generates encoded shots based on the shots, any number of resolutions, and any number of sets of values for a set of encoding parameters. At step 506, for each resolution, the shot-based encoding application 132 generates a convex hull of bitrate-quality points based on the encoded shots having the resolution.

[0154] At step 508, the encoding ladder application 150 optionally normalizes the bitrates and the quality scores specified in the bitrate-quality points across all convex hulls to the same range. At step 510, the encoding ladder application 150 defines the parameterized objective function 240 representing a weighted tradeoff between a weighted average of the quality scores (optionally normalized) across rungs of a candidate encoding ladder and the sum of the bitrates (optionally normalized) across the rungs. At step 512, the encoding ladder application 150 defines parameterized constraints for candidate encoding ladders.

[0155] At step 514, the encoding ladder application 150 generates one or more different ladder configurations that each specify a different combination of values for a rung count, rung quality weights, objective parameters, and constraint parameters. At step 516, for each ladder configuration, the encoding ladder application 150 generates an objective function and associated constraints based on the parameterized objective function 240 and the parameterized constraints.

[0156] At step 518, for each ladder configuration, the encoding ladder application 150 uses a constrained optimization solver to generate a different candidate encoding ladder that solves the objective function subject to the associated constraints. At step 520, the encoding ladder application 150 stores and/or transmits the candidate encoding ladders to any number and/or types of software applications for evaluation and selection of candidate encoding ladder(s) as encoding ladder(s) for the media title. The method 500 then terminates.

[0157] Figure 6 is a flow diagram of method steps for determining an encoding ladder for a media title based on adaptive streaming simulations of candidate encoding ladders that have been generated for the media title. Although the method steps are described with reference to the systems of Figures 1 - 4, persons skilled in

the art will understand that any system configured to implement the method steps, in any order, falls within the scope of the embodiments.

[0158] As shown, a method 600 begins at step 602, where the simulation evaluation application 170 selects a first candidate encoding ladder for a media title.

5 At step 604, the simulation evaluation application 170 generates a synthetic streaming header based on the selected candidate encoding ladder and the bitrates and quality scores for the encoded chunks of the associated encoded videos and then selects a first streaming session trace. At step 606, the simulation evaluation application 170 executes the adaptive streaming simulator 106 on the synthetic
10 streaming header and the selected streaming session trace to determine a corresponding request sequence and a corresponding metric value set.

[0159] At step 608, the simulation evaluation application 170 determines whether the selected streaming session trace is the last streaming session trace. If, at step 608, the simulation evaluation application 170 determines that the selected streaming
15 trace is not the last streaming session trace, then the method 600 proceeds to step 610. At step 610, the simulation evaluation application 170 selects the next streaming session trace. The method 600 then returns to step 606, where the simulation evaluation application 170 executes the adaptive streaming simulator 106 on the synthetic streaming header and the selected streaming session trace to determine a
20 corresponding request sequence and a corresponding metric value set.

[0160] If, however, at step 608, the simulation evaluation application 170 determines that the selected streaming session trace is the last streaming session trace, then the method 600 proceeds directly to step 612. At step 612, the simulation evaluation application 170 determines whether the selected candidate encoding
25 ladder is the last candidate encoding ladder for the media title. If, at step 612, the simulation evaluation application 170 determines that the selected candidate encoding ladder is not the last candidate encoding ladder for the media title, then the method 600 proceeds to step 614. At step 614, the simulation evaluation application 170 selects the next candidate encoding ladder for the media title. The method 600
30 then returns to step 604, where the simulation evaluation application 170 generates a synthetic streaming header based on the selected candidate encoding ladder and then selects the first streaming session trace.

[0161] If, however, at step 612, the simulation evaluation application 170 determines that the selected streaming session trace is the last streaming session trace, then the method 600 proceeds directly to step 616. At step 616, the simulation evaluation application 170 selects one or more of the candidate encoding ladders for further evaluation and/or deployment via the production encoding pipeline 120 based on the corresponding metric value sets. At step 618, the simulation evaluation application 170 generates an encoding ladder for the media title based on the selected candidate encoding ladder. At step 620, the simulation evaluation application 170 stores and/or transmits the encoding ladder for the media title to any number of software applications for further evaluation and/or deployment via the production encoding pipeline 120. The method 600 then terminates.

[0162] In sum, the disclosed techniques can be used to generate an encoding ladder for a media title that inherently represents an objective of increasing streaming QoE associated with the encoding ladder while decreasing the storage footprint of the encoding ladder and concurrently satisfying multiple constraints. In some embodiments, a shot-based encoding application partitions a source video corresponding to the media title into different shots. The shot-based encoding application encodes each shot across a set of resolutions and multiple different encoding parameter sets to generate encoded shots. For each resolution, the shot-based encoding application generates a convex hull of bitrate-quality points based on the encoded shots corresponding to the resolution. Each convex hull optimizes tradeoffs between bitrate and visual quality level for the resolution. Each of the bitrate-quality points specifies a different encoded video and the corresponding resolution, bitrate, and quality score. Notably, the resolution across each of the encoded videos is constant, but the bitrate and quality score can vary.

[0163] An encoding ladder application formulates the problem of generating an encoding ladder as a parameterized constrained optimization problem of assigning bitrate-quality points to rungs of a candidate encoding ladder. The encoding ladder application determines the constants of the parameterized constraint optimization problem based on the bitrate-quality points across the convex hulls. More precisely, the encoding ladder application optionally normalizes the bitrates and quality scores of the bitrate-quality points to the same range (e.g., a common range) to generate normalized bitrates and normalized quality scores. The encoding ladder application

organizes the resolutions, bitrates, quality scores, optional normalized bitrates, and optional normalized quality scores of the bitrate-quality points into arrays, where the indices of the arrays identify the corresponding bitrate-quality points.

[0164] The encoding ladder application defines the objective and constraints of the parameterized constrained optimization problem via a parameterized objective function and parameterized constraints. The parameterized objective function represents a weighted tradeoff between a weighted average of normalized quality scores across rungs of a candidate encoding ladder and the sum of normalized bitrates across the rungs. The number of rungs, the rung quality weights, and the tradeoff weights are parameters of the parameterized objective function. The parameterized constraints include both implicit logic constraints and practical logical constraints. Implicit logic constraints ensure the validity of candidate encoding ladders. For example, a monotonically increasing bitrate constraint ensures that the bitrates of encoded videos assigned to the rungs of a candidate encoding ladder monotonically increase from a lowest rung to a highest rung. Practical logical constraints capture operational restrictions and/or preferences that are associated with capabilities of client devices, network capacity, a CDN, human perception of visual quality, etc. For example, a parameterized bitrate spacing constraint ensures that bitrates of encoded videos assigned to the rungs of a candidate encoding ladder are separated by no more than a relative bitrate spacing.

[0165] The encoding ladder application generates multiple ladder configurations, where each ladder configuration is a different combination of values for the number of rungs, the rung quality weights, the tradeoff weights, and various parameters of the parameterized constraints (e.g., the relative bitrate spacing). For each ladder configuration, the encoding ladder generates an objective function and associated constraints based on the parameterized objective function and the parameterized constraints, respectively. The encoding ladder application uses a constrained optimization algorithm to solve each of the objective functions subject to the associated constraints, thereby generating a different assignment matrix for each ladder configuration. Each assignment matrix specifies a different assignment of bitrate-quality points to each rung of a different candidate encoding ladder that can be used as an encoding ladder for the media title.

[0166] In some embodiments, a numerical evaluation application performs numerical evaluations of the candidate encoding ladders using throughput distributions and/or bitrate demand distributions corresponding to historical streaming sessions. The numerical evaluation application filters out any number (including zero) of candidate encoding ladders representing sub-par tradeoffs between streaming QoE and storage footprint to generate a filtered candidate encoding ladder set.

[0167] In some embodiments, a streaming evaluation application performs simulation-based evaluations of the candidate encoding ladders in the filtered candidate encoding ladder set. The streaming evaluation application generates a different synthetic streaming header for each of the candidate encoding ladders in the filtered candidate encoding ladder set based on encoded chunk metadata associated with the encoded videos specified in or estimated (e.g., based on a curve) for virtual encoded videos associated with the candidate encoding ladders. Metadata estimated for a virtual encoded video refers herein to metadata that is estimated for an encoded video that could potentially be generated based on the source video 102. The synthetic streaming header for a given candidate encoding ladder includes a different streaming metadata set for each run in the candidate encoding ladder. Each streaming metadata set specifies a sequence of encoded chunks for a corresponding encoded video or a corresponding virtual encoded video, the bitrates of the encoded chunks, and the quality scores of the encoded chunks.

[0168] The streaming evaluation application uses an adaptive streaming simulator to emulate the behavior of an ABR algorithm using each of the candidate encoding ladders and the corresponding encoded chunk metadata over multiple simulated streaming sessions characterized by different streaming session traces. Each streaming session trace specifies network throughput as a function of time for a different historical streaming session. The result of each simulation is a request sequence of encoded chunks of the media title. For each request sequence, the streaming evaluation application computes a different set of values for a set of metrics that are relevant to streaming QoE. The streaming evaluation application performs any number and/or types of comparison operations between the sets of values for the set of metrics to select the candidate encoding ladders from the filtered candidate encoding ladder set that represents the best streaming QoE to storage footprint tradeoff across the different simulated streaming sessions. As used herein, the “best

streaming QoE to storage tradeoff" refers to a streaming QoE to storage tradeoff that most closely matches a target streaming QoE/storage footprint. The streaming evaluation application generates an encoding ladder for the media title based on the selected candidate encoding ladder.

5 **[0169]** At least one technical advantage of the disclosed techniques relative to the prior art is that, with the disclosed techniques, encoding ladders can be generated based on an overall objective of reducing the storage footprint of an encoding ladder while increasing the visual quality levels associated with the encoded videos included in the encoding ladder by concurrently accounting for different ladder constraints
10 when generating the encoding ladder in the first instance. With such an approach, opportunities to use a single encoded video that satisfies multiple different ladder constraints can be identified and exploited when generating an encoding ladder, which improves the tradeoff between the weighted average of the visual quality levels associated with the encoded videos in the encoding ladder and the storage footprint
15 of the encoding ladder. Consequently, the tradeoff between a streaming quality of experience represented by an encoding ladder for a given media title and the storage footprint of the encoding ladder can be substantially improved relative to what can be achieved using prior art techniques. These technical advantages provide one or more technological improvements over prior art approaches.

20 **[0170]** 1. In some embodiments, a computer-implemented method for generating encoding ladders that are used to stream media titles comprises generating a first objective function based on a first ladder configuration and a parameterized objective function that approximates a tradeoff between a quality of experience and a cost term associated with a candidate encoding ladder; generating a first plurality of constraints
25 based on the first ladder configuration and a plurality of parameterized constraints; and executing a constrained optimization algorithm on the first objective function, the first plurality of constraints, and encoding point metadata associated with a set of encoded videos to generate a first candidate encoding ladder for a media title.

30 **[0171]** 2. The computer-implemented method of clause 1, wherein generating the first objective function comprises replacing a first parameter that is included in the parameterized objective function with a rung count specified in the first ladder configuration.

[0172] 3. The computer-implemented method of clauses 1 or 2, wherein generating the first objective function comprises replacing one or more parameters included in the parameterized objective function with one or more parameter values that indicate a relative importance of the quality of experience and the cost term
5 associated with the candidate encoding ladder.

[0173] 4. The computer-implemented method of any of clauses 1-3, wherein generating the first plurality of constraints comprises replacing one or more parameters included in the first plurality of constraints with at least one of a rung count, a footprint upper bound, a high quality score, a minimum quality spacing, a
10 maximum quality spacing, a low bitrate, or a relative bitrate spacing.

[0174] 5. The computer-implemented method of any of clauses 1-4, further comprising determining different sets of parameter values for a union of a first set of parameters included in the parameterized objective function and a second set of parameters included in the plurality of parameterized constraints to generate the first
15 ladder configuration and at least one other ladder configuration.

[0175] 6. The computer-implemented method of any of clauses 1-5, wherein the encoding point metadata includes at least one of a set of resolutions, a set of quality scores, a set of bitrates, a set of normalized quality scores, or a set of normalized bitrates that are associated with the set of encoded videos.

20 **[0176]** 7. The computer-implemented method of any of clauses 1-6, further comprising normalizing the set of bitrates and the set of quality scores to a common range to generate the set of normalized bitrates and the set of normalized quality scores.

[0177] 8. The computer-implemented method of any of clauses 1-7, wherein a
25 term included in the parameterized objective function comprises a weighted average of a subset of a set of normalized quality scores associated with the set of encoded videos.

[0178] 9. The computer-implemented method of any of clauses 1-8, wherein the cost term is associated with at least one of a storage footprint or a network bandwidth
30 consumption.

- [0179]** 10. The computer-implemented method of any of clauses 1-9, wherein executing the constrained optimization algorithm comprises generating an assignment matrix specifying a plurality of assignments between the set of encoded videos and a plurality of rungs to be included in the first candidate encoding ladder.
- 5 **[0180]** 11. In some embodiments, one or more non-transitory computer readable media include instructions that, when executed by one or more processors, cause the one or more processors to generate encoding ladders that are used to stream media titles by performing the steps of generating a first objective function based on a first ladder configuration and a parameterized objective function that approximates a
- 10 tradeoff between a quality of experience and a cost term associated with a candidate encoding ladder; generating a first plurality of constraints based on the first ladder configuration and a plurality of parameterized constraints; and executing a constrained optimization algorithm on the first objective function, the first plurality of constraints, and encoding point metadata associated with a set of encoded videos to
- 15 generate a first candidate encoding ladder for a media title.
- [0181]** 12. The one or more non-transitory computer readable media of clause 11, wherein generating the first objective function comprises replacing a first parameter that is included in the parameterized objective function with a rung count specified in the first ladder configuration.
- 20 **[0182]** 13. The one or more non-transitory computer readable media of clauses 11 or 12, wherein generating the first objective function comprises computing a plurality of values for a plurality of weights based on the first ladder configuration, wherein the plurality of weights are referenced by the first objective function and are associated with a plurality of rungs.
- 25 **[0183]** 14. The one or more non-transitory computer readable media of any of clauses 11-13, wherein generating the first plurality of constraints comprises replacing one or more parameters included in the first plurality of constraints with at least one of a rung count, a footprint upper bound, a high quality score, a minimum quality spacing, a maximum quality spacing, a low bitrate, or a relative bitrate spacing.
- 30 **[0184]** 15. The one or more non-transitory computer readable media of any of clauses 11-14, further comprising determining different sets of parameter values for a

union of a first set of parameters included in the parameterized objective function and a second set of parameters included in the plurality of parameterized constraints to generate the first ladder configuration and at least one other ladder configuration.

5 [0185] 16. The one or more non-transitory computer readable media of any of clauses 11-15, wherein the encoding point metadata includes at least one of a set of resolutions, a set of quality scores, a set of bitrates, a set of normalized quality scores, or a set of normalized bitrates that are associated with the set of encoded videos.

10 [0186] 17. The one or more non-transitory computer readable media of any of clauses 11-16, wherein a term included in the parameterized objective function comprises an average of a subset of a set of normalized bitrates associated with the set of encoded videos.

15 [0187] 18. The one or more non-transitory computer readable media of any of clauses 11-17, wherein the first plurality of constraints comprises at least one constraint associated with at least one resolution that the candidate encoding ladder represents.

20 [0188] 19. The one or more non-transitory computer readable media of any of clauses 11-18, further comprising generating a second objective function based on a second ladder configuration and the parameterized objective function; generating a second plurality of constraints based on the second ladder configuration and the plurality of parameterized constraints; and executing the constrained optimization algorithm on the second objective function, the second plurality of constraints, and the encoding point metadata to generate a second candidate encoding ladder for the media title.

25 [0189] 20. In some embodiments, a system comprises one or more memories storing instructions and one or more processors coupled to the one or more memories that, when executing the instructions, perform the steps of generating a first objective function based on a first ladder configuration and a parameterized objective function that approximates a tradeoff between a quality of experience and a cost term
30 associated with a candidate encoding ladder; generating a first plurality of constraints based on the first ladder configuration and a plurality of parameterized constraints;

and executing a constrained optimization algorithm on the first objective function, the first plurality of constraints, and encoding point metadata associated with a set of encoded videos to generate a first candidate encoding ladder for a media title.

5 [0190] Any and all combinations of any of the claim elements recited in any of the claims and/or any elements described in this application, in any fashion, fall within the contemplated scope of the present invention and protection.

[0191] The descriptions of the various embodiments have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those
10 of ordinary skill in the art without departing from the scope and spirit of the described embodiments.

[0192] Aspects of the present embodiments may be embodied as a system, method, or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software
15 embodiment (including firmware, resident software, micro-code, etc.), or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “module” or “system.” Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code
20 embodied thereon.

[0193] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic,
25 infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only
30 memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer

readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0194] Aspects of the present disclosure are described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general-purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine. The instructions, when executed via the processor of the computer or other programmable data processing apparatus, enable the implementation of the functions/acts specified in the flowchart and/or block diagram block or blocks. Such processors may be, without limitation, general-purpose processors, special-purpose processors, application-specific processors, or field-programmable gate arrays.

[0195] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0196] While the preceding is directed to embodiments of the present disclosure, other and further embodiments of the disclosure may be devised without departing

from the basic scope thereof, and the scope thereof is determined by the claims that follow.

WHAT IS CLAIMED IS:

1. A computer-implemented method for generating encoding ladders that are used to stream media titles, the method comprising:
 - generating a first objective function based on a first ladder configuration and a
5 parameterized objective function that approximates a tradeoff between a
quality of experience and a cost term associated with a candidate
encoding ladder;
 - generating a first plurality of constraints based on the first ladder configuration
and a plurality of parameterized constraints; and
 - 10 executing a constrained optimization algorithm on the first objective function,
the first plurality of constraints, and encoding point metadata associated
with a set of encoded videos to generate a first candidate encoding
ladder for a media title.
- 15 2. The computer-implemented method of claim 1, wherein generating the first
objective function comprises replacing a first parameter that is included in the
parameterized objective function with a rung count specified in the first ladder
configuration.
- 20 3. The computer-implemented method of claim 1, wherein generating the first
objective function comprises replacing one or more parameters included in the
parameterized objective function with one or more parameter values that indicate a
relative importance of the quality of experience and the cost term associated with the
candidate encoding ladder.
- 25 4. The computer-implemented method of claim 1, wherein generating the first
plurality of constraints comprises replacing one or more parameters included in the
first plurality of constraints with at least one of a rung count, a footprint upper bound, a
high quality score, a minimum quality spacing, a maximum quality spacing, a low
30 bitrate, or a relative bitrate spacing.
5. The computer-implemented method of claim 1, further comprising determining
different sets of parameter values for a union of a first set of parameters included in
the parameterized objective function and a second set of parameters included in the

plurality of parameterized constraints to generate the first ladder configuration and at least one other ladder configuration.

6. The computer-implemented method of claim 1, wherein the encoding point
5 metadata includes at least one of a set of resolutions, a set of quality scores, a set of
bitrates, a set of normalized quality scores, or a set of normalized bitrates that are
associated with the set of encoded videos.

7. The computer-implemented method of claim 6, further comprising normalizing
10 the set of bitrates and the set of quality scores to a common range to generate the set
of normalized bitrates and the set of normalized quality scores.

8. The computer-implemented method of claim 1, wherein a term included in the
parameterized objective function comprises a weighted average of a subset of a set
15 of normalized quality scores associated with the set of encoded videos.

9. The computer-implemented method of claim 1, wherein the cost term is
associated with at least one of a storage footprint or a network bandwidth
consumption.

20

10. The computer-implemented method of claim 1, wherein executing the
constrained optimization algorithm comprises generating an assignment matrix
specifying a plurality of assignments between the set of encoded videos and a
plurality of rungs to be included in the first candidate encoding ladder.

25

11. One or more non-transitory computer readable media including instructions
that, when executed by one or more processors, cause the one or more processors to
generate encoding ladders that are used to stream media titles by performing the
steps of:

30

generating a first objective function based on a first ladder configuration and a
parameterized objective function that approximates a tradeoff between a
quality of experience and a cost term associated with a candidate
encoding ladder;
generating a first plurality of constraints based on the first ladder configuration

and a plurality of parameterized constraints; and
executing a constrained optimization algorithm on the first objective function,
the first plurality of constraints, and encoding point metadata associated
with a set of encoded videos to generate a first candidate encoding
5 ladder for a media title.

12. The one or more non-transitory computer readable media of claim 11, wherein
generating the first objective function comprises replacing a first parameter that is
included in the parameterized objective function with a rung count specified in the first
10 ladder configuration.

13. The one or more non-transitory computer readable media of claim 11, wherein
generating the first objective function comprises computing a plurality of values for a
plurality of weights based on the first ladder configuration, wherein the plurality of
15 weights are referenced by the first objective function and are associated with a
plurality of rungs.

14. The one or more non-transitory computer readable media of claim 11, wherein
generating the first plurality of constraints comprises replacing one or more
20 parameters included in the first plurality of constraints with at least one of a rung
count, a footprint upper bound, a high quality score, a minimum quality spacing, a
maximum quality spacing, a low bitrate, or a relative bitrate spacing.

15. The one or more non-transitory computer readable media of claim 11, further
25 comprising determining different sets of parameter values for a union of a first set of
parameters included in the parameterized objective function and a second set of
parameters included in the plurality of parameterized constraints to generate the first
ladder configuration and at least one other ladder configuration.

30 16. The one or more non-transitory computer readable media of claim 11, wherein
the encoding point metadata includes at least one of a set of resolutions, a set of
quality scores, a set of bitrates, a set of normalized quality scores, or a set of
normalized bitrates that are associated with the set of encoded videos.

17. The one or more non-transitory computer readable media of claim 11, wherein a term included in the parameterized objective function comprises an average of a subset of a set of normalized bitrates associated with the set of encoded videos.
- 5 18. The one or more non-transitory computer readable media of claim 11, wherein the first plurality of constraints comprises at least one constraint associated with at least one resolution that the candidate encoding ladder represents.
- 10 19. The one or more non-transitory computer readable media of claim 11, further comprising:
- generating a second objective function based on a second ladder configuration and the parameterized objective function;
- generating a second plurality of constraints based on the second ladder configuration and the plurality of parameterized constraints; and
- 15 executing the constrained optimization algorithm on the second objective function, the second plurality of constraints, and the encoding point metadata to generate a second candidate encoding ladder for the media title.
- 20 20. A system comprising:
- one or more memories storing instructions; and
- one or more processors coupled to the one or more memories that, when executing the instructions, perform the steps of:
- generating a first objective function based on a first ladder configuration
- 25 and a parameterized objective function that approximates a tradeoff between a quality of experience and a cost term associated with a candidate encoding ladder;
- generating a first plurality of constraints based on the first ladder configuration and a plurality of parameterized constraints; and
- 30 executing a constrained optimization algorithm on the first objective function, the first plurality of constraints, and encoding point metadata associated with a set of encoded videos to generate a first candidate encoding ladder for a media title.