Hit Ratio and Content Quality Tradeoff for Adaptive Bitrate Streaming in Edge Caching Systems

Nhu-Ngoc Dao, Duy T. Ngo, Ngoc-Thanh Dinh, Trung V. Phan, Nam D. Vo, Sungrae Cho, and Torsten Braun

***Abstract*—This paper addresses the tradeoff problem between hit ratio and content quality in edge caching systems for multi-user adaptive bitrate streaming (ABS) services. A dynamic policy for cache decision and quality level selection for each ABS content during every cache cycle is proposed. Achieving this policy is NP-complete. For this, the considered problem is transformed into a nested multidimensional 0/1 knapsack optimization problem which is then resolved by a cooperative transfer learning-accelerated genetic algorithm. Performance evaluation demonstrates an adaptation of the proposed algorithm on various video stream popularity models in terms of algorithmic convergence and cache balancing.**

***Index Terms*—Adaptive bitrate streaming, cache balancing, edge caching systems**

I. Introduction

In a modern mobile video streaming, heterogeneous user demands and preferences are supported by an adaptive bitrate streaming (ABS) service. The ABS service enables networks to dynamically adjust the quality level of content chunks (represented by video bitrate) to respond to the change of environmental conditions and resource availability [1]. To ensure efficient cooperation between network elements in the ABS systems, the International Organization for Standardization (ISO) publishes the ISO/IEC 23009-5:2017 standard [2] which defines a functional architecture, namecoded SAND. The SAND architecture enables network-assisted video streaming ability along with dynamic adaptive streaming over HTTP (DASH) protocol utilization. Here, content chunks are transcoded and/or cached on the delivery path by the network elements closer to users. These in-network services are referred to as the edge caching systems (ECSs).

However, the computing capacity of ECS network elements (called edge servers) tend to be totally overwhelmed by the increasing user demands and preferences [1]. As reviewed in [3], [4], existing works mainly optimized the ECS towards its service efficiency such as hit ratio and resource usage maximization. For instance, Wang *et al.* [5] exploit the collaboration among edge servers for an integrated cache placement and video retrieval to obtain hit ratio maximization and content access latency reduction. To extend this work, a joint cache and radio resource allocation scheme is proposed in [6], where wireless access from users to edge servers is controlled by using Stackelberg game theory to maximize spectrum efficiency and hit ratio. On the other hand, a multipath video streaming framework is designed in [7] with a scalable video coding capability to maximize the video playback rate. Although these approaches have successfully improved network service efficiency, their achievements come at a cost of content quality (i.e., user experience) reduction owing to resource constraints at the edge servers.

To address this catch-22 situation, this paper investigates the tradeoff problem between hit ratio and content quality (referred to as HICOT) in the ECS for multi-user ABS services. Specifically, cache decision and content quality selection are tightly integrated into a flexible policy. Here, the hit ratio is supervised by a logarithmic function whereas the content quality is modeled by an average of production with content popularity. Achieving this tradeoff policy is NP-complete. For this, the policy is transformed to a nested multidimensional 0/1 knapsack (nMKP) optimization problem which delivers cache decision on every ABS video content with their own optimal bitrate. We propose the HICOT algorithm that integrates a cooperative transfer learning technique into genetic mechanism to quickly resolve the nMKP problem. The advantages of the HICOT algorithm are summarized as follows:

- Its objective is developed to jointly consider two paradoxical objectives of ABS services, i.e., cache hit ratio and content quality. Once the optimal solution is established, the objective may shift its target towards either hit ratio awareness or content quality awareness by just adjusting the tradeoff factor.
- Although its optimization problem is NP-complete, the HICOT algorithm obtains approximately optimal solution with a quick convergence by accelerating a genetic mechanism using a cooperative transfer learning.
- Performance evaluation demonstrates that the HICOT algorithm well adapts to all three typical video stream popularity distributions such as Zipf, uniform, and random models.

II. System Model

A typical system model of ABS services is illustrated in Fig. 1. In this model, a DASH player sends status messages to the SAND servers to request for information about its desired content. A SAND server located at the edge server in the proximity of the user device then returns a response message indicating a maximum bitrate available to the user, which is calculated by the caching policy. Accordingly, a transcoder of the SAND server inquires the original content from the content providers and encodes the content at the assigned bitrate.

This research was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00889, A study on core technology of 5G mobile communication using millimeter wave band). *(Corresponding author: Sungrae Cho.)*

N.-N. Dao is with the Department of Computer Science and Engineering, Sejong University, South Korea (email: nndao@sejong.ac.kr).

D. T. Ngo is with the School of Electrical Engineering and Computing, University of Newcastle, Australia (email: duy.ngo@newcastle.edu.au).

N.-T. Dinh is with the School of Electrical and Telecommunication, Soongsil University, South Korea (email: thanhdcn@dcn.ssu.ac.kr).

T. V. Phan is with Technische Universität Chemnitz, Chair of Communication Networks, Germany (email: trung.phan-van@etit.tu-chemnitz.de).

N. D. Vo is with the Department of Academic Affairs, University of Danang, Vietnam (email: vdnam@ac.udn.vn).

S. Cho is with the School of Computer Science and Engineering, Chung-Ang University, South Korea (email: srcho@cau.ac.kr).

T. Braun is with the Institute of Computer Science, University of Bern, Switzerland (email: torsten.braun@inf.unibe.ch).

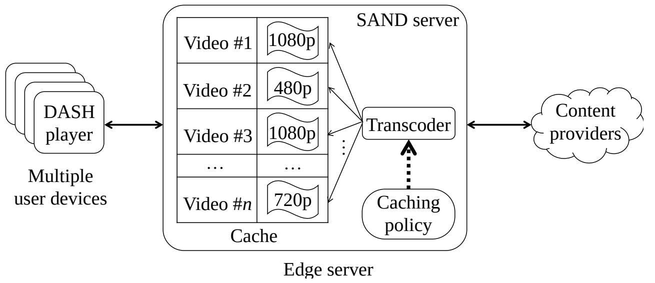


Fig. 1. ABS services adopting SAND architecture deployment in an ECS.

The system parameters at timeslot t are defined as follows. The backhaul bandwidth between the edge server and the content provider is W bps. Regarding user demands, let $\mathcal{N}$ and N denote the set of desired ABS contents and its cardinality, respectively. Assume that the ith ABS content has its probabilistic popularity p_i [4]. In addition, the ith ABS content can be transcoded at a bitrate l_i bps with a computational cost f_i Hz from the original content. The bitrate l_i belongs to a pre-determined bitrate set $\mathcal{L}$ with dimension L. In addition, the edge server is equipped with a Q-byte cache storage and it has a maximum computing capacity F Hz. A general form of the tradeoff problem can be expressed by

$$\max \kappa \mathbf{f}(\text{hit ratio}) + (1-\kappa)\mathbf{g}(\text{content quality}), \tag{1}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are functions of the hit ratio and the content quality, respectively; while κ $(0 \le \kappa \le 1)$ is a tradeoff factor to balance the impacts of these two functions.

III. Hit Ratio and Content Quality Tradeoff

A. Problem Statement

To obtain an increase of the hit ratio at the edge server, $\mathbf{f}(\cdot) \propto \sum_{i=1}^{N} \alpha_i p_i$ must be satisfied, where α_i is a binary parameter presenting cache placement decision of the ith content in the edge server. If $\alpha_i = 0$, the ith content is cached at the edge server during the considered timeslot; otherwise, the content is not cached. To control the priority of the hit ratio, a natural logarithm is applied on $\mathbf{f}(\cdot)$ in order to promote the impact of the hit ratio when it is small and reduce its impact when its volume is large enough [1]. That is, $\mathbf{f}(\cdot) = \log\left(\sum_{i=1}^{N} \alpha_i p_i\right)$.

For content quality function development, $\mathbf{g}(\cdot)$ is designed as a linear function of l_i. To generalize the quality of all cached contents in the edge server, an average bitrate is derived from all user requests. As a result, $\mathbf{g}(\cdot)$ at the edge server is given by

$$\mathbf{g}(\cdot) = \frac{1}{\sum_{i=1}^{N} \alpha_i p_i} \sum_{i=1}^{N} \left(\alpha_i p_i \sum_{j=1}^{L} \beta_{ij} l_{ij} \right)$$
$$\text{s.t.} \sum_{j=1}^{L} \beta_{ij} \le 1, \quad \beta_{ij} \in \{0,1\}, \forall i,$$

where β_{ij} is a parameter of bitrate selection for the ith content in $\mathcal{L}$. Accordingly, the tradeoff problem $(\mathcal{P})$ is formulated as

$$(\mathcal{P}) \quad \max_{\alpha_i, \beta_{ij}} \; \kappa \log\left(\sum_{i=1}^{N} \alpha_i p_i\right) +$$
$$+(1-\kappa)\frac{1}{\sum_{i=1}^{N} \alpha_i p_i} \sum_{i=1}^{N} \left(\alpha_i p_i \sum_{j=1}^{L} \beta_{ij} l_{ij} \right) \tag{2}$$

subject to

$$0 < \sum_{i=1}^{N} \alpha_i \sum_{j=1}^{L} \beta_{ij} l_{ij} \le Q, \ | \ \alpha_i, \beta_{ij} \in \{0,1\}, \tag{2a}$$

$$\sum_{i=1}^{N} \left(\alpha_i p_i K c_i + (1-\alpha_i) \sum_{j=1}^{L} \beta_{ij} l_{ij} \right) \le W, \tag{2b}$$

$$\sum_{i=1}^{N} \alpha_i f_i \le F, \tag{2c}$$

$$\sum_{j=1}^{L} \beta_{ij} \le 1, \forall i. \tag{2d}$$

In (2), (2a) ensures that total size of the cached contents does not over-capacitate the cache. Moreover, $\sum_{i=1}^{N} \alpha_i \sum_{j=1}^{L} \beta_{ij} l_{ij} > 0$; otherwise, no content is cached, i.e., an optimal calculation is unnecessary. (2b) illustrates the total bandwidth occupation for delivery of all ABS contents from the content provider to the edge servers on the backhaul link. $\alpha_i p_i K c_i$ is the amount of bandwidth consumed by the ith content without cache, where K and c_i are the number of user requests and bitrate of the original content, respectively. (2c) ensures the transcoding workload can be managed by the computing capacity of the edge server. (2d) insists that at most one bitrate can be selected for each content. It is seen that $(\mathcal{P})$ is *NP-complete* owing to its form of an integer programming problem.

B. HICOT Algorithm

Lemma 1. *Utility of selecting content k at bitrate l_{kj} is positive if $l_{kj} p_k$ is not less than an average of all $\sum_{j=1}^{L} \beta_{ij} l_{ij} p_i$ of the decided contents.*

Proof. Assume that the edge server has a decision on N_1 contents excluding content k. If the content k satisfies all constraints (2a)–(2d), the utility u_{kj} of selecting the content k at a bitrate l_{kj} can be derived from the differential of the objective (2) before and after selecting the content k, i.e.,

$$u_{kj} = \kappa \log\left(1 + \frac{p_k}{\sum_{i=1}^{N_1} \alpha_i p_i}\right) +$$
$$+(1-\kappa)\frac{1}{\sum_{i=1}^{N_1} \alpha_i p_i + 1} \left(l_{kj} p_k - \frac{\sum_{i=1}^{N_1} \left(\alpha_i p_i \sum_{j=1}^{L} \beta_{ij} l_{ij} \right)}{\sum_{i=1}^{N_1} \alpha_i p_i} \right). \tag{3}$$

(3) shows that the utility of selecting the content k at the bitrate l_{kj} varies depending on the selection order of the content. In spite of that, the variation of the utility with different orders impacts on all contents equally. In addition, the right term in (3) shows that this utility is positive if $l_{kj} p_k$ is not less than an average of all $\sum_{j=1}^{L} \beta_{ij} l_{ij} p_i$ of the decided contents. ■

Lemma 2. *Utility of selecting the content k at the bitrate l_{kj} is independent of the order of content selection in the final decision.*

Proof. Similarly, given that the edge server finally has its decision on all N contents including a content k, the utility of selecting the content k at a bitrate l_{kj} can be derived from

Algorithm 1 Hit Ratio and Content Quality Tradeoff

```
1:  Initiate $\mathcal{I} = \{I_1, I_2, \ldots, I_p\}, I_i \in \{0,1\}^{NL}$;
2:  Evaluate $\mathcal{P}(\mathcal{I}) = \{\mathcal{P}(I_1), \mathcal{P}(I_2), \ldots, \mathcal{P}(I_p)\}$ using Eq. (4);
3:  $I^{\max} = \arg\max_{I_i \in \mathcal{I}} \mathcal{P}(I_i)$;
4:  $t = 0, \epsilon = 1$;
5:  while $(t < T)$ AND $(\epsilon)$ do
6:      Crossover $C = \mathcal{X}(I_f, I_m), \{I_f, I_m\} \in \mathcal{I}$;
7:      Mutate and repair $C = \Omega(C)$ s.t. (2a)–(2d), $C \not\equiv \exists I_i \in \mathcal{I}$;
8:      Evaluate $\mathcal{P}(C)$;
9:      $I^{\min} = \arg\min_{I_i \in \mathcal{I}} \mathcal{P}(I_i)$. In $\mathcal{I}$, $I^{\min} \leftarrow C$;
10:     $I^{\max} = \arg\max_{I^{\max}, C} \{\mathcal{P}(I^{\max}), \mathcal{P}(C)\}$;
11:     Check the convergence and decide $\epsilon, \epsilon \in \{0,1\}$;
12:     $t = t + 1$;
13: return $I^{\max}$;
```

the differential of the objective (2) with and without selecting the content k, i.e.,

$$\kappa \log\left(1 + \frac{p_k}{\sum_{i=1}^{N} \alpha_i p_i - p_k}\right) +$$

$$+(1-\kappa)\frac{1}{\sum_{i=1}^{N} \alpha_i p_i - 1}\left(l_{kj} p_k - \frac{\sum_{i=1}^{N}\left(\alpha_i p_i \sum_{j=1}^{L} \beta_{ij} l_{ij}\right)}{\sum_{i=1}^{N} \alpha_i p_i}\right). \tag{4}$$

(4) is independent of the order of content k in N as all terms in (4) is specified in the final decision. ■

$(\mathcal{P})$ expresses a situation in which the edge server must decide which ABS contents at which bitrates to temporally cache on the memory subject to the cache size, backhaul bandwidth, and computing capacity limitations. Owing to Lemmas 1 and 2, $(\mathcal{P})$ can be regarded as an nMKP optimization, where a 0/1 knapsack solution is applied on content selection, and for each given selected content, another 0/1 knapsack solution is applied on its bitrate. In order to reduce the complexity of nMKP optimization, we transform $(\mathcal{P})$ into a typical MKP by considering that a content at each bitrate is a distinct item for selection. The utility of an item is given by (3). Consequently, we have:

$$\max_{\gamma_{kj}} \sum_{kj=1}^{NL} u_{kj} \gamma_{kj} \tag{5}$$
$$\text{s.t. (2a)–(2d), (3).}$$

Because the cache size Q, backhaul bandwidth W, and computing capacity F are large compared to the number of contents NL, we adopt a genetic mechanism to resolve the problem and propose the HICOT. A pseudocode of the HICOT is illustrated in Alg. 1.

Initially, a set of feasible solutions $\mathcal{I}$ is generated, where each element I_i is a vector of γ_{kj} and $I_i \in \{0,1\}^{NL}$. The utilities of the feasible solutions are calculated using Eq. (4). Among these feasible solutions, a solution that has the maximum utility is labeled as $I^{\max}$. The number of attempts T and the convergence indicator ϵ are setup, where T defines the maximum number of attempts in the genetic loop and ϵ indicates whether the algorithm has converged ($\epsilon = 0$) or not ($\epsilon = 1$) [Lines 1–4 in Alg. 1]. The algorithm is considered convergent if a particular number of continuous attempts returns no change in $I^{\max}$. Each attempt in the genetic loop [Lines 5–12 in Alg. 1], first selects a couple of I_f and I_m arbitrarily in $\mathcal{I}$. A crossover operation $\mathcal{X}(\cdot)$ is constructed by deriving γ_{kj} elements from I_f and I_m to generate a new vector C of the same size NL. The selected γ_{kj} is the content that has the best utility-constraint ratios r_{kj} [8] given by

$$r_{kj} = \frac{u_{kj}}{\frac{1}{3}\left(v_1 l_{kj} + v_2\left(\alpha_k p_k K c_k + (1-\alpha_k) l_{kj}\right) + v_3 f_k\right)}, \tag{6}$$

where v_1, v_2, and v_3 are the corresponding Lagrangian multipliers of dimensions Q, W, and F, respectively. To ensure C is unique and feasible, a mutation and repair operation $\Omega(\cdot)$ is performed on C by switching the values of several γ_{kj} elements. Consequently, the utility of C is calculated to update $I^{\max}$ as well as to replace the solution that has the minimum utility in $\mathcal{I}$. Lastly, the convergence of the algorithm is checked to update ϵ. When the genetic loop is completed, $I^{\max}$ is returned as the most appropriate solution.

C. Cooperative Transfer Learning based Acceleration

In Alg. 1, the initial feasible set $\mathcal{I}$ mainly impact the convergence. To accelerate the convergence of the HICOT algorithm, we utilize the cooperative transfer learning method among edge servers to initiate $\mathcal{I}$ for every genetic loop in Alg. 1. Specifically, a common feasible set $\mathcal{I}_c$ is maintained by learning the most efficient caching solutions transferred from all edge servers. A solution in $\mathcal{I}_c$ is labelled by a corresponding environment tuple $\langle Q, W, F\rangle$, which represents the context where the contributor edge server found this solution. In particular, each time an edge server obtains its own solution by performing the HICOT algorithm, the edge server compares its achieved utility (5) with the utility provided by the solution that has the best matching environment tuple (i.e., the shortest Euclidean distance) in $\mathcal{I}_c$. From this comparison, the solution that results in a higher utility is retained in $\mathcal{I}_c$. $\mathcal{I}_c$ is used as the initial feasible set $\mathcal{I}$ for the edge servers.

IV. Performance Evaluation

We generated 500 samples of 50 video streams adopting three typical video stream distribution assumptions: (i) With the Zipf distribution, the exponent is set to 1.161 following the 20-80 Pareto law [9]; (ii) With the uniform distribution, the popularity of every videos is perfectly set equal to 1/50, i.e., 2%, to follow a theoretical uniform distribution; (iii) With the random distribution, we used the `rand(·)` function available in MATLAB to generate video samples. The `rand(·)` function returns uniformly distributed random integer numbers in the interval [1, 50]. Video chunk size is set 2 s applying for all videos. The backhaul bandwidth between the edge server and the content provider is assumed to be 1.0 Gbps. The video streams are encoded at a bitrate in $\mathcal{L} = \{235, 375, 560, 750, 1050, 1750, 2350, 3000, 4300, 5800\}$ kbps [1]. The computational costs to transcode the original contents at the desired bitrates are proportional to the bitrates as $\{235, 375, 560, 750, 1050, 1750, 2350, 3000, 4300, 5800\}$ × 10 kHz [10]. Edge servers are equipped with a 2.4-GHz CPU.

Fig. 2 illustrates the convergence speed of HICOT algorithm in the three experiments. The termination condition is set to be a combination of the average relative change in the objective function value (10^{-3}) and the minimum number of stall generations (1000 iterations). In general, HICOT converges rapidly to an approximately optimal solution in fewer than 20,000 iterations. In the uniform distribution, the convergence is achieved after 100 iterations since all video demands have

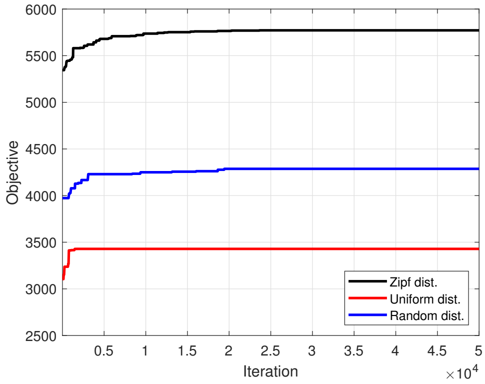
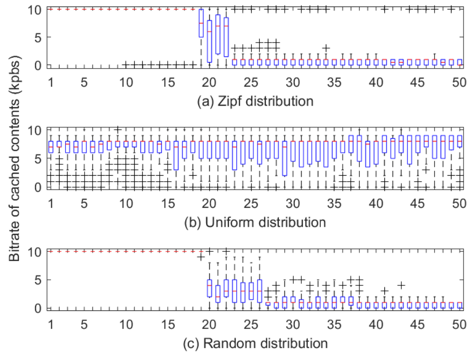


Fig. 2. Algorithmic convergence of HICOT.



(a) Zipf distribution

(b) Uniform distribution

(c) Random distribution

Fig. 3. Bitrate selection of 50 video contents in popularity descending order.

an approximate popularity. Meanwhile, HICOT requires more time to converge in the Zipf and random distributions as the popularity of video demands is diverse.

Fig. 3 presents the box plots of the bitrate selection. The red line is the mean of video bitrates while the dimension of the box indicates the fluctuation of bitrate selection depending on κ. It is observed that high-popularity video contents are prioritized to be cached with high quality while low-popularity videos might be cached with low quality or not be cached at all. A high bitrate selection fluctuation occurred to videos with medium popularity (e.g., videos indexed 20–25 with the Zipf and random distribution). In the central plot, the uniform distribution shows the mean and fluctuation of video bitrates are approximate among contents.

Fig. 4 shows the impact of the tradeoff factor κ on hit ratio and average video bitrate balancing. According to (2), two special cases $\kappa = 0$ and 1 of the HICOT algorithm make the objective be quality-aware and hit ratio-aware optimizations, respectively. As discussed in Section III, κ prioritizes the hit ratio compared to the average video quality. We observe that the tradeoff value that switches priority on hit ratio and quality is different according to video demand distributions. Hence, the tradeoff factor should be selected around the switching tradeoff value to get a dynamic balance between the hit ratio and the video quality. In the Zipf case, the hit ratio slightly increases before jumping to a high value compared to other distributions because of the 20–80 Pareto law.

Fig. 5 shows the impact of cache buffer size on the average bitrate. Since video demands with uniform distribution have approximate popularity, all videos are handled equally with the same caching priority that makes the average bitrate linear in the cache size. In the Zipf and random cases, high-popularity videos are cached with a high bitrate. Therefore, the average bitrate rapidly increases when the cache buffer size is small.

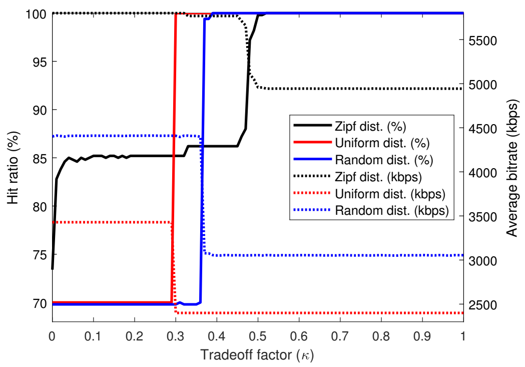


Fig. 4. Impact of the tradeoff factor on hit ratio and average bitrate.

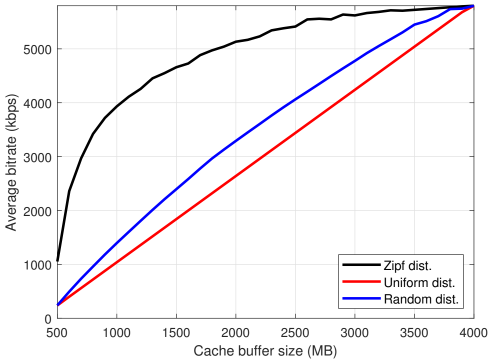


Fig. 5. Average bitrate of cached contents with various cache buffer sizes.

V. Conclusion

This paper proposes a dynamic policy for balancing between cache decision and quality selection for multi-user ABS services in an ECS. This tradeoff problem has been resolved by the proposed cooperative transfer learning-accelerated genetic algorithm named HICOT. Performance evaluation has been conducted to investigate the adaptation of the proposed HICOT algorithm against three typical video stream popularities. To extend the study, user mobility and audience retention rate will be considered in the future work.

References

[1] S. Pham, P. Heeren, D. Silhavy, and S. Arbanowski, "Evaluation of shared resource allocation using SAND for ABR streaming," in *Proc. of ACM MMSys'19*, Amherst, MA, June 18–21, 2019, pp. 165–174.

[2] "Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 5: Server and network assisted DASH (SAND)," ISO/IEC 23009-5:2017, Feb. 2017.

[3] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 1, pp. 562–585, 2018.

[4] H. S. Goian *et al.*, "Popularity-based video caching techniques for cache-enabled networks: A survey," *IEEE Access*, vol. 7, pp. 27 699–27 719, 2019.

[5] Y. Wang, Y. Zhang, M. Sheng, and K. Guo, "On the interaction of video caching and retrieving in multi-server mobile-edge computing systems," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1444–1447, 2019.

[6] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16 406–16 415, 2017.

[7] A. Elgabli, K. Liu, and V. Aggarwal, "Optimized preference-aware multi-path video streaming with scalable video coding," *IEEE. Trans. Mob. Comput.*, vol. 19, no. 1, pp. 159–172, 2020.

[8] S. Shah, "Genetic algorithm for the 0/1 multidimensional knapsack problem," *arXiv preprint arXiv:1908.08022*, 2019.

[9] D. Volchenkov and X. Leoncini, *Regularity and Stochasticity of Nonlinear Dynamical Systems*. Springer, 2018.

[10] D. Lee, J. Lee, and M. Song, "Video quality adaptation for limiting transcoding energy consumption in video servers," *IEEE Access*, vol. 7, pp. 126 253–126 264, 2019.