J. Vis. Commun. Image R. 79 (2021) 103210

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

Tile caching for scalable VR video streaming over 5G mobile networks☆

Kedong Liu [a,d], Yanwei Liu [a,*], Jinxia Liu [b], Antonios Argyriou [c]

[a] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[b] Zhejiang Wanli University, Ningbo, China
[c] University of Thessaly, Volos, Greece
[d] National Computer Network Emergency Response Technical Team/Coordination Center of China, China

ARTICLE INFO

Keywords:
VR video
Caching
Video tile
5G systems

ABSTRACT

Currently, VR video delivery over 5G systems is still a very complicated endeavor. One of the major challenges for VR video streaming is the expectations for low latency that current mobile networks can hardly meet. Network caching can reduce the content delivery latency efficiently. However, current caching schemes cannot obtain ideal results for VR video since it requests the viewport interactively. In this paper, we propose a tiled scalable VR video caching scheme over 5G networks. VR chunks are first encoded into multi-granularity quality layers, and are then partitioned into tiles to facilitate viewport data access. By accommodating the 5G network infrastructure, the tiles are cooperatively cached in a three-level hierarchal system to reduce delivery latency. Furthermore, a quality-adaptive request routing algorithm is designed to cater for the 5G bandwidth dynamics. Experimental results show that the proposed scheme can reduce the transmission latency over conventional constant bitrate video caching schemes.

1. Introduction

VR video applications are increasingly popular today and are gradually becoming one of the main types of video that people prefer. To create an immersive experience for the end user, VR video provides a super high resolution 360-degree field of view (FoV), and thus usually tends to consume a large amount of storage space and transmission bandwidth [40]. As a result, present-day online VR applications usually cannot meet user expectations because of the limited network bandwidth. Furthermore, due to the interactive viewport-requesting nature, VR video systems usually have a very strict latency requirement [1]. This brings a great pressure on the network, especially the wireless network that is the last hop in the entire delivery chain to the end user.

The 5G wireless communication system is going to be commercialized in the next few years. Though the capacity of 5G mobile networks has been significantly improved, it is still very difficult to transmit VR video over 5G networks due to the high volume of data. Especially, with multiple users competing for one shared network channel, it is more challenging to deliver high-quality VR video over 5G networks.

To deal with such 'elephant flows', i.e. the VR video traffic over 5G networks, many research works have exploited the novel techniques of 5G networks so as to improve the communication performance. The broadcast/multicast Point-to-Multipoint transmissions in 5G networks enable multi-user wireless VR applications. In [2], Prasad et al. investigated the specific challenges for delivering VR videos or games to a large number of audience using broadcasting technologies in 5G networks. A newer element of 5G in spectrum utilization, i.e. millimeter wave (mmWave) small cells, can provide the super high transmission bitrates. This is naturally in line with the demand of the ultra-low delay VR video transmission. However, millimeter wave data transmission is easy to be blocked. To cope with the problem of VR transmission intermittence caused by blockages, the authors in [3] proposed to add a mmWave mirror device to relay the blocked signal. By utilizing a multi-connectivity-based mmWave cellular network, Liu et al. [4] proposed a cross-layer optimization approach to improve VR video streaming performance. In 5G networks, mobile edge computing can also be utilized to reduce the increased computational requirements of the mobile device. Schmoll et al. in [5] demonstrated offloading of VR rendering to the mobile edge cloud for 5G applications. By offloading the viewport rendering to a mobile edge server, VR rendering computational delay can be significantly reduced by optimizing the trade-off between computational gains and communication costs.

☆ This paper has been recommended for acceptance by Zicheng Liu

* Corresponding author.
E-mail address: liuyanwei@iie.ac.cn (Y. Liu).

https://doi.org/10.1016/j.jvcir.2021.103210
Received 1 October 2019; Received in revised form 16 August 2020; Accepted 27 June 2021
Available online 9 July 2021

VR videos are usually experienced through a head-mounted display (HMD). The viewport (also indifferently called FoV) is only a fraction of 360-degree view of the scene. In one moment, only one viewport in the whole frame is requested for viewing. By moving the HMD dynamically with the head motion, different viewpoints of scenes in the VR video are viewed. This interactive nature of VR video is fundamentally different from that of the conventional planar video that renders the full-frame data on a screen. To cater for the way that data are requested by the HMD, the tiling scheme that is often unnecessary for conventional planar video, is utilized to enable the spatially random access functionality within one VR video frame. Tiling facilitates bandwidth reduction, since only the necessary viewport data are delivered to the user equipment (UE) at one moment. Tiling of VR video also raises new challenges for migrating the traditional full-frame caching approach to VR video. Interactive viewing requires timely data updates in the HMD. To measure the timeliness of data updates in the HMD, Motion-to-Photon (MTP) latency is defined as the delay between the movement of the user's head and the change of the VR device's display reflecting the user's movement. According to an investigation in [6], optimal VR experience requires MTP latency to be less than 20 ms. Contrary to the conventional planar video, this strict latency requirement is an additional challenge for high-quality VR video delivery. Besides the other processing (including the video decoding and viewport rendering at the receiver) delays, the transmission latency in 5G networks is still a crucial problem. Even worse, there is resource competition among the multiple wireless users.

Regarding video streaming latency, video caching has been proposed to push the duplicate videos near the end user and relieve the pressure on backhaul links. In [7], Xie et al. studied the effects of different access types on Internet video services and their implications on Content Delivery Network (CDN) caching. Franky et al. in [8] studied a video cache system which can reduce the video traffic and the loading time. Furthermore, two key problems were concerned by prior work [9][10]. One is the video content placement i.e., determining which content should be placed at which cache nodes for a given topology and file popularity distribution. Another is the video cache routing during the delivery, i.e. scheduling the video requests [11] inside the cache system.

As the last mile during video transmission, the mobile network is closest to the end user. Consequently, video caching was further extended to mobile networks [12][13]. With the development of 5G networks, 5G mobile in-network caching has also been considered as a technique suitable for reducing the video delivery latency [14][15]. To deal with the huge demand of VR video streaming in 5G systems, 5G caching technology was also used to optimize the VR video delivery. Sun et al. in [16] developed a framework for mobile VR delivery by utilizing the caching and computing capabilities of the mobile VR device. In [17], Sukhmani et al. presented an edge caching strategy for 5G VR applications. These approaches are effective in reducing the delivery latency to some extent. However, they neglected the VR video tiling during caching, and cannot obtain the optimal results for VR video delivery over 5G networks.

Several recent video caching works have considered the particular features of VR video. In [18], Liu et al. studied the joint EPC and RAN caching for tiled VR videos. Mahzari et al. in [19] proposed a caching policy based on the users' FoV, called FoV-aware caching policy, and trained a probabilistic model of common-FoV for each 360-degree video to improve caching performance. In [20], Papaioannou et al. studied the problem of tile-based panoramic video caching optimization, that determines which tiles and tile resolutions to cache. Even though the above studies can optimize VR video caching efficiency, they ignored the adaptation of VR video request routing to the dynamic channel.

In the current literature additional features of VR video have been utilized to optimize VR video streaming. Commonly, VR video is partitioned into several tiles spatially to facilitate viewport-adaptive streaming. Gaddam et al. in [21] applied a tiling scheme to deliver different quality levels for different parts of panoramic VR video. In [22], Skupin et al. used dynamic adaptive streaming over http (DASH) to transmit VR tiles for HMD. In [23], Concolato et al. presented an adaptive streaming of tiled High Efficiency Video Coding (HEVC) videos using MPEG-DASH. The above-mentioned approaches optimized the viewport by adaptively selecting the quality level of viewport to adapt to the network. Motivated by these approaches,*we propose to utilize quality-scalable tiled VR video as the cache source content and then couple multi-bitrate caching with fine-granularity network-adaptive streaming to further optimize the VR video delivery performance*.

In this paper, by linking the characteristics of salable VR video with mobile networks, a tiled scalable VR video caching scheme over 5G networks is presented. The contributions of this paper are summarized below.

(1) Taking into account the fact that only a small portion of 360-degree VR video is visible to a viewer at one moment, we propose to cache tiles of scalable VR video in 5G networks. Compared to the conventional full-frame video caching approaches, a tile caching strategy significantly improves the cache hit ratio, while reducing the viewport fetching latency by quickly responding to the interactive request.
(2) To leverage the 5G network architecture, the tiled multi-bitrate VR video chunks are cooperatively cached into a three-tier (including the source server) caching system. The matching of the tile popularity and quality with the hierarchy of caching system is optimized during the stage of VR video tile placement. Thus, the video tiles with higher popularity and higher quality (higher bitrate) are more likely to be cached in the radio access network (RAN) which is closer to the UEs. This raises the cache hit ratio and saves significant bandwidth.
(3) Based on the multi-bitrate VR video tile caching, a quality-adaptive viewport request routing algorithm is proposed that adapts to the fluctuations of 5G channel. Under the channel bandwidth constraint, the appropriate quality combination (bitrate combination) of viewport and non-viewport tiles in one frame that maximizes the homogeneous viewport quality is selected for satisfying the user's request. This scheme achieves the desired trade-off between the requested viewport quality and the corresponding data delivery latency.

The rest of the paper is organized as follows. The proposed tiled scalable VR video caching framework over 5G networks is presented in Section 2. Specifically, the tile cache placement of scalable VR video is introduced and quality-adaptive request routing algorithm is described. Experimental results are shown in Section 3. Finally, Section 4 concludes the paper.

2. Multi-bitrate VR Video Tile Caching Framework

The proposed scalable VR video tile caching system over 5G networks is shown in Fig. 1. Since the Stand-Alone (SA) new radio (NR) specification for 5G has been adopted [24], the proposed caching system is based on SA architecture. The caching system is composed of three parts: the Internet VR video server, 5G Core (5GC) network cache node and Next Generation RAN (NG-RAN) cache nodes. The source server on the Internet stores all of the multi-bitrate versions of VR video tiles. In the 5GC network, there is a cache node attached to the AMF (Access and Mobility Management Function)/UPF (User Plane Function). In the NG-RAN, 5G NR base stations (gNBs) are used as cache nodes. The cache space in gNBs is extended by accompanying the mobile edge computing (MEC) servers [25][26], which enable not only the enhanced computing capabilities but also the enlarged storage at the edge of the cellular NG-RAN. In the 5G SA architecture, the gNBs are connected to each other via the Xn interface. NG-RAN connects to 5GC network using the NG interface.

In the 5GC network, there is a logical centrally-deployed entity,

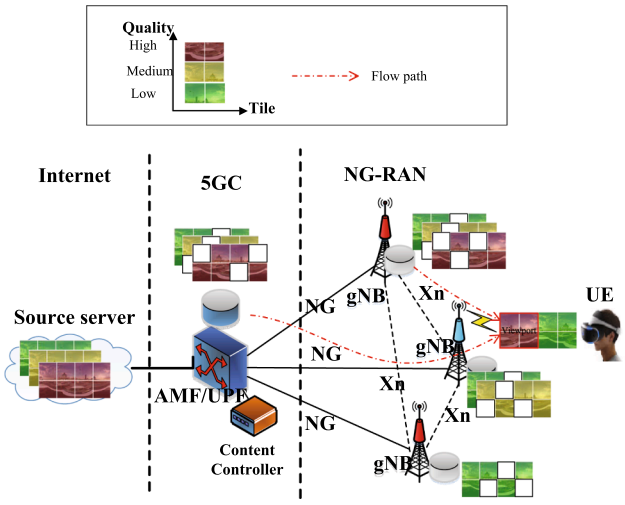

Fig. 1. Tiled scalable VR video caching system over 5G networks.

namely the content controller, which is connected to the AMF/UPF. The content controller is responsible for recognizing VR video viewport request from UEs and then executes the caching optimization algorithm based on information collected from each cache node.

To provide fine-granularity scalability of quality for VR video delivery, VR source videos were encoded by SHVC (Scalable High efficiency Video Coding). After encoding, the multiple bitrates VR video streams can be extracted from the compressed stream. In order to enable fast access to viewport data in one full-frame, one VR video is segmented into tiles spatially in one frame during encoding. Furthermore, in the temporal dimension, the encoded VR video tiles are divided into different chunks with equal playback length. Thus, for each chunk period in the VR video, different combinations of tile qualities over the whole frame can be obtained. Unequal quality allocation among viewport tiles and non-viewport tiles under the given bandwidth constraint is also supported.

Usually, contemporary caching systems are essentially networks of interconnected caches [41]. The cache placement solves which content to cache at each server and the content routing handles how to route content from caches to the end users. The caching and routing decisions are inherently coupled, as a request can only be routed to cache where the requested item is available. Therefore, the caching problem in its entirety includes decisions about content placement, and content routing once the cache server is fixed.

During the cache placement stage, the tiles of the VR videos are stored in the video source server, 5GC cache node (AMF/UPF) and NG-RAN cache nodes. Usually, to ensure that the end user can obtain the maximum allowable quality of VR video, high-bitrate tiles should be cached closer to UE and low-bitrate tiles can be cached farther from the UE. When one UE requests a VR video for watching, the caching system will utilize the request routing algorithm to select a combination of viewport quality and non-viewport quality over the whole frame under the constraint of available channel bandwidth. In particular, the tiles in the range of viewport are chosen with higher bitrates than those out of the range of viewport due to the higher access probability of these viewport tiles.

Now we formally introduce our framework. Assume that a total number of $\mathscr{K}$ VR videos will be cached in the 5G mobile network. We denote $p_0, p_1, \cdots, p_i, \cdots, p_I$ as the total $I+1$ cache nodes available for the VR video tile caching. As shown in Fig. 2, p_0 denotes the caching node in 5GC and $p_1, \cdots, p_i, \cdots, p_I$ denote the cache nodes in NG-RAN, respectively. $r_1, r_2, \cdots, r_j, \cdots, r_J$ denote the different bitrates of the VR video tiles. $v_t^{k,m,n}$ denotes the VR video tile at the mth row and nth column at time slot t for

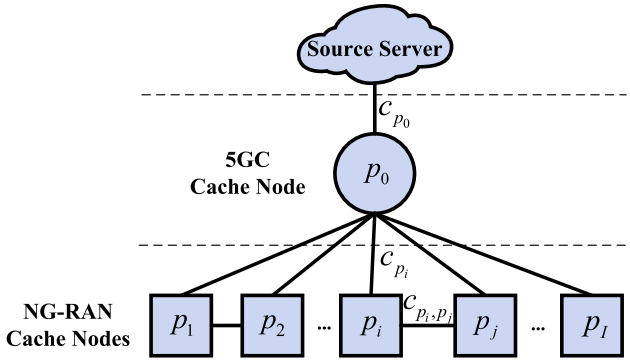

Fig. 2. Hierarchical topology of tile caching over 5G networks.

the kth VR video. For one tile $v_t^{k,m,n}$, we define a 0–1 variable $x_{t,p_i,j}^{k,m,n}$ to indicate whether it is cached in p_i with bitrate r_j. If p_i had already cached the tile $v_t^{k,m,n}$ with bitrate r_j, $x_{t,p_i,j}^{k,m,n} = 1$; otherwise $x_{t,p_i,j}^{k,m,n} = 0$. Based on the above definitions, the caching result of the tiles with different bitrates for one video sequence in all caching nodes can be described as a vector of 0–1 variables, that is $\mathbf{X_{p,r}} = \{x_{1,0,1}^{1,1,1}, x_{1,0,1}^{1,1,2}, x_{1,0,1}^{1,2,1}, \cdots, x_{t,p_i,j}^{k,m,n}, \cdots, x_{\mathscr{T},p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\}$, where $\mathscr{T}$ is the total number of time slots, $\mathscr{M}$ the tile row number and $\mathscr{N}$ the tile column number in one frame. For the cache placement, there are a lot of candidate choices, and they form a candidate solution set $\widetilde{\mathbf{X}}$. In the routing stage, we define a 0–1 decision variable $y_{t,p_i,j}^{k,m,n}$ to indicate whether the request of tile $v_t^{k,m,n}$ in p_i with bitrate of r_j is routed to the end user. At each time slot, the candidate request routing result for all tiles in one frame is denoted as $\mathbf{Y_t} = \{y_{t,0,1}^{1,1,1}, y_{t,0,1}^{1,1,2}, y_{t,0,1}^{1,2,1}, \cdots, y_{t,p_i,j}^{k,m,n}, \cdots, y_{t,p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\}$. Similar to the cache placement, the candidate solution set of request routing is denoted by $\widetilde{\mathbf{Y}}_\mathbf{t}$.

The hierarchical topology of tile caching is illustrated as a graph in Fig. 2. Now let c_{p_i} denote the unit cost for transferring a VR video tile from the 5GC cache node to an NG-RAN cache node p_i, c_{p_0} the unit cost when transferring VR video tile from the source server to 5GC cache node, and c_{p_i,p_j} the unit cost when transferring a VR video tile between the NG-RAN cache nodes p_i and p_j. By saving the delivery bandwidth cost, the tiled multi-bitrate VR video chunks can be cooperatively cached into a three-tier system to make the tile popularity match with the hierarchy topology of the network. In this premise, the viewport request can be routed to the suitable cache node where the requested tile is available. We consider video quality within the viewport of each user as the objective for VR video delivery optimization. Thus, the joint optimization problem of tiled multi-bitrate VR video tile cache placement and quality-adaptive request routing can be mathematically formulated as

$$\max_{\mathbf{X_{p,r}}\in\widetilde{\mathbf{X}},\mathbf{Y_t}\in\widetilde{\mathbf{Y}}_\mathbf{t}} \sum_{v_t^{k,m,n}\in V_t^{k,VP}} r_j \cdot y_{t,p_i,j}^{k,m,n}, \forall t \leqslant \mathscr{T} \tag{1}$$

$$\text{subjectto} \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} r_j \cdot y_{t,p_i,j}^{k,m,n} \leqslant W_t, \forall t \leqslant \mathscr{T} \tag{1a}$$

$$\max_{v_t^{k,m,n}\in V_t^{k,VP}} \left\{ \frac{r_j \cdot y_{t,p_i,j}^{k,m,n} \cdot t_c}{w_i} \right\} \leqslant T_d \tag{1b}$$

$$\sum_{k=1}^{\mathscr{K}} \sum_{t=1}^{\mathscr{T}} \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} \sum_{j=1}^{J} r_j \cdot t_c \cdot x_{t,p_i,j}^{k,m,n} \leqslant B_{p_i} \tag{1c}$$

$$y_{t,p_i,j}^{k,m,n} \in \{0,1\}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K} \tag{1d}$$

$$x_{t,p_i,j}^{k,m,n} \in \left\{0,1\right\}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K} \tag{1e}$$

$$y_{t,p_i,j}^{k,m,n} \leqslant x_{t,p_i,j}^{k,m,n}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K} \tag{1f}$$

$$\mathbf{Y_t} = \left\{y_{t,0,1}^{1,1,1}, y_{t,0,1}^{1,1,2}, y_{t,0,1}^{1,2,1}, \cdots, y_{t,p_i,j}^{k,m,n}, \cdots, y_{t,p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\right\} \tag{1g}$$

$$\mathbf{X_{p,r}} = \left\{x_{1,0,1}^{1,1,1}, x_{1,0,1}^{1,1,2}, x_{1,0,1}^{1,2,1}, \cdots, x_{t,p_i,j}^{k,m,n}, \cdots, x_{\mathscr{T},p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\right\}, \tag{1h}$$

where $V_t^{k,VP}$ denotes the set of VR video tiles within the viewport at time slot t for the kth VR video, W_t denotes the downlink bandwidth of the mobile networks for the UE during the time slot t, w_i denotes the available bandwidth from the caching node p_i to the UE, T_d denotes the maximum limitation of transmission latency that is allowed by the VR video application, t_c denotes the playback length of chunk at the t-th time slot, and B_{p_i} denotes the storage constraint for cache node p_i.

In our optimization formulation Eq. (1), constraint Eq. (1a) is the bandwidth restriction. It ensures that the sum of bitrates for all the requested VR video tiles at t-th time slot should be no more than the downlink bandwidth W_t. Constraint Eq. (1b) is the latency restriction for VR tile delivery. It means that the transmission latency of every tile should be equal to or smaller than the maximum transmission latency T_d. The inequality in Eq. (1c) represents the total amount of tile data placed in a cache node must not exceed its storage capacity. Constraints Eq. (1)d and Eq. (1e) describe the cache placement decision variable $x_{t,p_i,j}^{k,m,n}$ and the routing decision variable $y_{t,p_i,j}^{k,m,n}$, respectively. The inequality in Eq. (1f) means that in order for the tile request from UE to be routed to a cache node, the tile needs to be placed in the latter. It indicates that the requests from UEs should be responded with the cached VR video tiles as much as possible. Finally, the equations in Eq. (1)g and Eq. (1h) characterize the cache placement decision variable vector $\mathbf{X_{p,r}}$ and the routing decision variable vector $\mathbf{Y_t}$, respectively.

Originally, the cache placement and viewport requests are two events that happen sequentially in the VR video delivery chain, and their time granularity of update is also different. Usually, cache content is periodically updated with a interval that depends on the tile popularity changes. Unlike traditional request routing, viewport request routing needs to be scheduled immediately for each request. Based on the previous discussion on cache data placement and viewport requests, the multi-bitrate VR video tile caching optimization problem is divided into two decoupled sub-problems, which are the multi-bitrate tile placement sub-problem and the viewport request routing sub-problem. First, under the latency constraint and the caching space constraint, different bitrate versions of VR video tiles are cooperatively cached to different hierarchical cache nodes in terms of their popularity. Second, under the latency constraint, the requested VR video tiles that constitute one viewport for the user are fetched to the UE from different cache nodes by respecting the real-time bandwidth constraint of the wireless network.

2.1. VR Tile Placement Optimization

We first deal with the problem of VR video tile placement optimization. The optimization goal of VR video tile placement is to reduce the total bandwidth cost of the caching system when compared to fetching tiles from the source server. In our previous work [18], we proposed a tile-based VR video cache optimization framework for 4G LTE networks. In the following, we extend our previous work [18] to multiple-bitrate VR video tile caching in 5G networks. Similar with 4G mobile network caching, there are basically four ways to fetch a VR video viewport for viewers in 5G networks:

- If the cache node p_i can fulfill the request from UE locally for VR video tile $v_t^{k,m,n}$, the unit cost saving is $c_{p_0} + c_{p_i}$.
- If the request cannot be fulfilled by the directly connected gNB p_i, but can be fulfilled by other adjacent gNB that is directly connected to the gNB to which UE belongs, for instance the cache node $p_j (p_i \neq p_j)$, the unit cost savings can be computed as $c_{p_0} + c_{p_i} - c_{p_i,p_j}$.
- If the request can be fulfilled by the 5GC cache node p_0, the unit cost saving is c_{p_0}.
- If the request can only be fulfilled from the source server on the Internet, the unit cost saving is zero.

We define the bandwidth cost savings as $G_{p_i,j}^{k,m,n}$ when the request for VR video tile $v_t^{k,m,n}$ with bitrate r_j at node p_i is fulfilled by the EPC cache node. $G_{p_i,j}^{k,m,n}$ is given by

$$G_{p_i,j}^{k,m,n} = c_{p_0} \times x_{t,p_0,j}^{k,m,n} \tag{2}$$

Also, the maximal saving cost $H_{p_i,p_j,j}^{k,m,n}$ when the request for VR video tile $v_t^{k,m,n}$ with bitrate r_j at node p_i is fulfilled by another node p_j is defined as

$$H_{p_i,p_j,j}^{k,m,n} = \max_{p_j \in \mathbb{I} \setminus \{p_i\}} \left\{ \left(c_{p_0} + c_{p_i} - c_{p_i,p_j}\right) \cdot x_{p_i,p_j,j}^{k,m,n} \right\} \tag{3}$$

where $\mathbb{I}$ is the set of cache nodes which can be expressed as $\mathbb{I} = \{p_0, p_1, \cdots p_i, \cdots p_j, \cdots p_I\}$.

Based on the above discussion, the total saved bandwidth cost $\tau$for all cached tiles compared to the way that obtains VR video tiles from the source server can be calculated as

$$\begin{aligned} \tau &= \tau(\mathbf{X_{p,r}}) \\ &= \sum_{k=1}^{\mathscr{K}} \sum_{t=1}^{\mathscr{T}} \sum_{p_i=0}^{\mathbb{I}} \sum_{j=1}^{J} \lambda_{t,p_i,j}^{k,m,n} \cdot r_j \cdot t_c \cdot \left[x_{t,p_i,j}^{k,m,n} \cdot \left(c_{p_0} + c_{p_i}\right) \right. \\ &\qquad \left. + \left(1 - x_{t,p_i,j}^{k,m,n}\right) \cdot \max\left\{ G_{p_i,j}^{k,m,n}, H_{p_i,p_j,j}^{k,m,n} \right\} \right] \end{aligned} \tag{4}$$

where $\lambda_{t,p_i,j}^{k,m,n}$ denotes the request probability of VR video tile $v_t^{k,m,n}$ with bitrate r_j at node p_i. The request probability $\lambda_{t,p_i,j}^{k,m,n}$ for VR video tile $v_t^{k,m,n}$ from UE who connects to cache node p_i is given by

$$\lambda_{t,p_i,j}^{k,m,n} = \pi_{p_i}^{k} \cdot \theta_{t,j}^{k,m,n} \tag{5}$$

where $\pi_{p_i}^{k}$ indicates the probability of requesting the kth VR video from the UE who connects to cache node p_i, which follows the Zipf's law [27] and $\theta_{t,j}^{k,m,n}$ denotes the request probability for $v_t^{k,m,n}$ in the kth VR video, which can be obtained from the viewport popularity data by estimating the saliency map of the VR video [28][29]. The subscript j in $\theta_{t,j}^{k,m,n}$ indicates the jth bitrate version of $v_t^{k,m,n}$. In this paper, we assume the probabilities of requesting for $v_t^{k,m,n}$ in the kth VR video for different bitrate versions are the same. According to the above knowledge, the video tile placement optimization sub-problem of maximizing the saving cost τ can be mathematically formulated as

$$\max_{\mathbf{X_{p,r}} \in \mathbf{X}} \tau \tag{6}$$

$$\text{subjectto} \sum_{k=1}^{\mathscr{K}} \sum_{t=1}^{\mathscr{T}} \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} \sum_{j=1}^{J} r_j \cdot t_c \cdot x_{t,p_i,j}^{k,m,n} \leqslant B_{p_i}$$

$$x_{t,p_i,j}^{k,m,n} \in \left\{0,1\right\}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K}$$

$$\mathbf{X_{p,r}} = \left\{x_{1,0,1}^{1,1,1}, x_{1,0,1}^{1,1,2}, x_{1,0,1}^{1,2,1}, \cdots, x_{t,p_i,j}^{k,m,n}, \cdots, x_{\mathscr{T},p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\right\},$$

The problem formulation in Eq. (6) can be explained as follows. Given a set of multiple bit-rate versions of tiles, that each tile has a bit-rate value (weight) and a profit value $\lambda_{t,p_i,j}^{k,m,n}$, and a set $\mathbb{I}$ of cache spaces, that each is

with capacity of B_{p_i}, a subset of multiple bit-rate versions of tiles can be found by maximizing the total profit such that they can be placed into the cache space set $\mathbb{I}$, without exceeding the capacities. The above observation shows that Eq. (6) is exactly in line with the definition of standard multiple-knapsack problem. Hence, the multi-bitrate video tile placement optimization is a typical 0–1 multiple-knapsack problem.

Due to its combinatorial nature, the 0–1 multiple-knapsack is a NP-hard problem. It is well known that NP-hard problems cannot be solved in a polynomial time. However, we can find a subset of solutions that can be the largest target value from the candidate solution set under the premise of satisfying various resource constraints. In the past years, researchers proposed many approximation algorithms to solve the 0–1 multiple-knapsack problem for obtaining near-optimal solutions [30]. When the number of constraints and decision variables is large in the formulation, the efficiency of traditional approximation optimization approaches [30] is limited in both execution time and quality of final solutions. Comparably, genetic algorithms (GA) have shown to be very well suited for solving larger knapsack problems [31][42][43]. As we all know, GA has the advantage of the global optimization and the parallelism in seeking the solutions to the optimization problem, which indicates the solution-searching process can be implemented in parallel. Thus, to find the final result of VR video tile placement optimization problem, we adopt a genetic algorithm.

Usually, the standard GA adopts the evolutionary biology techniques, such as inheritance, mutation, selection, and crossover. The evolution happens in generations with starting from a population of randomly generated individuals (also represented by chromosome). In each generation, the fitness of every individual in the population is evaluated first, and multiple individuals are then selected from the current population and modified to form a new population. The new population is used for evolution in the next iteration of the algorithm. The algorithm terminates when a maximum number of generations has been produced.

For the specific multi-bitrate tile cache placement problem, the objective is to maximize the saving cost τ(a nonnegative value), which can quantitatively measure how well a given cache placement solution is, and thus the fitness function is defined as $f(x) = \tau$in GA. Correspondingly, the final result is to find the cache placement solution $\mathbf{X_{p,r}}$, which has the highest value of fitness function.

To represent the solution space of the problem in the GA, each possible solution is regarded as a individual in one generation of populations. The individual is generally represented by a binary encoding string that also called chromosome. In our algorithm, the solution is naturally characterized by a binary vector. Thus the binary encoding string $\mathbf{X} = \{x_{1,0,1}^{1,1,1}, x_{1,0,1}^{1,1,2}, x_{1,0,1}^{1,2,1}, \cdots, x_{t,p_i j}^{k,m,n}, \cdots, x_{\mathscr{T},p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}}\}$is used as the chromosome and $x_{t,p_i j}^{k,m,n}$ is taken as the gene in the chromosome. The chromosome length l denotes the number of 0–1 variables $x_{t,p_i j}^{k,m,n}$ in one solution result and $l = \mathscr{K} \times \mathscr{M} \times \mathscr{N} \times \mathscr{T} \times (p_I + 1) \times J$. First, some input parameters are assigned values towards achieving a tradeoff between the quality of solution and the convergence speed of the algorithm. Specifically, the population size s_{pop} is set to 50, the probability of performing crossover p_c is equal to 0.8, the probability of the mutation p_m is set to 0.02 and the maximal number of generations n_{ge} for terminating the algorithm is set to 500. Then, the first generation of the population is initialized by generating the candidate solutions of the caching result. Next, the fitness value τ of each chromosome $\mathbf{X}$ is calculated. If the chromosome $\mathbf{X}$ doesn't satisfy the constraints in Eq. (6), the fitness value τ will be zero. In the following, the crossover and mutation operations are performed to generate a new generation. Finally, after n_{ge} loops, we can achieve the caching result $\mathbf{X_{p,r}}$. Since the GA belongs to a non-deterministic class of algorithms, the obtained solution may vary for each run of the algorithm with the same input parameters. Thus the final result $\mathbf{X_{p,r}}$ is rather sub-optimal. Regarding the details of the specific algorithm flow the reader can refer to [18].

2.2. Quality-Adaptive Request Routing

In the VR video delivery system, the near-optimal cache placement provides low-delay response for VR video tile requests. To further adapt the bit-rate of VR video tile that is currently requested to the fluctuated 5G channel, a quality-adaptive tile request routing can be employed. The quality-adaptive viewport requests are scheduled in a time granularity according to each request. The optimization objective for quality-adaptive request routing is to maximize the homogeneous video quality within the user's viewport under the constrained channel bandwidth. In other words, the optimization goal of the request routing problem is to maximize the bitrates of all the tiles covered by the users' viewport. Under the premise of near-optimal multi-bitrate VR video tile placement, the problem of maximizing the bitrate of tile that the user currently demands by request routing optimization can be mathematically formulated as

$$\max_{\mathbf{Y_t} \in \widetilde{\mathbf{Y_t}}} \sum_{v_t^{k,m,n} \in V_t^{k,VP}} r_j \cdot y_{t,p_i j}^{k,m,n}, \forall t \leqslant \mathscr{T} \tag{7}$$

$$\text{subjectto} \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} r_j \cdot y_{t,p_i j}^{k,m,n} \leqslant W_t, \forall t \leqslant \mathscr{T}$$

$$\max_{v_t^{k,m,n} \in V_t^{k,VP}} \left\{ \frac{r_j \cdot y_{t,p_i j}^{k,m,n} \cdot t_c}{w_i} \right\} \leqslant T_d$$

$$y_{t,p_i j}^{k,m,n} \in \{0, 1\}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K}$$

$$y_{t,p_i j}^{k,m,n} \leqslant x_{t,p_i j}^{k,m,n}, \forall m \leqslant \mathscr{M}, \forall n \leqslant \mathscr{N}, \forall t \leqslant \mathscr{T}, \forall p_i \leqslant p_I, \forall j \leqslant J, \ \forall k \leqslant \mathscr{K},$$

$$\mathbf{Y_t} = \left\{ y_{t,0,1}^{1,1,1}, y_{t,0,1}^{1,1,2}, y_{t,0,1}^{1,2,1}, \cdots, y_{t,p_i j}^{k,m,n}, \cdots, y_{t,p_I,J}^{\mathscr{K},\mathscr{M},\mathscr{N}} \right\},$$

Since VR video browsing requires timely viewport data update, the whole frame data needs to be fetched to the user end. Even though only the viewport data are used for playback, the other data outside the viewport region are utilized as the standby data that can be used to timely update viewport when HMD moves rapidly. Hence, this approach requires a reasonable bitrate allocation over the whole frame under total bitrate constraint to ensure relatively higher viewport bitrate. Besides that, different cache nodes can provide different bitrates of the tile. Based on a prior information above, the request routing optimization problem can be solved by a heuristic searching algorithm. In this paper, we propose a heuristic α-search algorithm to maintain the smooth video quality across the tiles in the range of viewport. Two definitions are first given to help understand the proposed algorithm.

Definition 1. The sum of bitrates of all tiles within the viewport region for the kth VR video should be no more than $\alpha(0 < \alpha \leqslant 1)$ times W_t. The sum of the kth VR video tiles' bitrates outside the viewport region should be less than the result of $(1-\alpha)$times W_t for time slot t. In addition, the bitrate difference among the different tiles should be as small as possible. Let us denote as R_1 and R_2 the sum of bitrates for tiles within viewport region and outside viewport region for the kth VR video, respectively. Assume $r_t^{k,m,n}$ denotes the corresponding bitrate of $v_t^{k,m,n}$, then

$$R_1 = \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} \sum_{v_t^{k,m,n} \in V_t^{k,VP}} r_t^{k,m,n} \leqslant \alpha \cdot W_t \tag{8}$$

$$R_2 = \sum_{m=1}^{\mathscr{M}} \sum_{n=1}^{\mathscr{N}} \sum_{v_t^{k,m,n} \notin V_t^{k,VP}} r_t^{k,m,n} < (1-\alpha) \cdot W_t \tag{9}$$

Input: Information of bandwidth W_t, w_i and w_{min}.
Output: $\mathbf{Y_t}$ and A set of bitrates $\Re$ of different tiles in the requested full-frame VR video chunk at time slot t

Initialize $\Delta\alpha$, $\Re = \emptyset$, and $r_{\min}$;
$R_2 = n_{nv} \cdot r_{\min}$;
$R_1 = W_t - R_2$;
$\alpha = \frac{R_1}{W_t}$;
repeat //sub loop 1
  $R_1 = \alpha \cdot W_t$;
  $\bar{r}_1 = R_1/n_v$;
  $lat_1 = \frac{\bar{r}_1 \cdot t_c}{w_{\min}}$;
  $\alpha = \alpha - \Delta\alpha$;
until $lat_1 \le T_d$;
repeat
  // sub loop 2 ;
  Use a hierarchical searching approach to select a combination $\mathbb{Z}_v = \{\Re_1, \Re_2, \cdots, \Re_{n_v}\}$ of cached video tiles which satisfies $\sum_{i=1}^{n_v} \Re_i \le R_1$, and $\Re_i \in x_{t,p_i,j}^{k,m,n} \cdot r_j,\ \forall p_i, \forall j,\ v_t^{k,m,n} \in V_t^{k,VP}$;
  $lat_2 = \max_{1 \le i \le n_v} \{\frac{\Re_i \cdot t_c}{w_i}\}$;
  if $lat_2 > T_d$ **then**
    $\mathbb{Z}_v = \emptyset$;
  Record the specific $y_{t,p_i,j}^{k,m,n}$ that corresponds to $\Re_i$;
until $lat_2 \le T_d$;
if $\mathbb{Z}_v = \emptyset$ **then**
  $\alpha = \alpha - \Delta\alpha$;
  goto 5;
$\Re = \Re \cup \mathbb{Z}_v$;
Use a hierarchical searching approach to select a combination of cached $\mathbb{Z}_{nv} = \{\Re_1, \Re_2, \cdots, \Re_{n_{nv}}\}$ video tiles which satisfies $\sum_{i=1}^{n_{nv}} \Re_i \le R_2$, and record the specific $y_{t,p_i,j}^{k,m,n}$ that corresponds to $\Re_i$;
$\Re = \Re \cup \mathbb{Z}_{nv}$;
return $\mathbf{Y_t}$ and $\Re$;

Definition 2. The sum of R_1 and R_2 should be no more than the available bandwidth W_t from NG-RAN to UE at time slot t. That is

$$R_1 + R_2 \leqslant W_t \tag{10}$$

Based on a smaller step size $\Delta\alpha$, we can carefully search over the appropriate bitrates of the viewport region and non-viewport region for the requested VR video chunk at time slot t. Initially, $\Delta\alpha$is set to 0.1. The solution set $\Re$is set to null. Then, we set $R_2 = n_{nv} \cdot r_{\min}$, where n_{nv} is the number of tiles in the non-viewport region and $r_{\min}$ is the lowest bitrate of the tile that the VR video delivery system can provide. To capture the dynamics of the caching system payloads, the information of bandwidth W_t, w_i and w_{min} are probed in real-time based on the packet transmission rate (PTR) [32]. The detailed α-searching algorithm is shown in Algorithm 1. In terms of Definition 2, $R_1 + R_2 \leqslant \alpha \cdot W_t + (1-\alpha) \cdot W_t = W_t, R_1$ and α can be calculated after determining R_2. Next, in the sub-loop 1 in Algorithm 1, the bitrate of the viewport region is updated. Specifically, the average bitrate $\bar{r}_1$ of the VR video tiles within the viewport region can be first calculated based on Definition 1. After that, the transmission latency $lat_1 = \frac{\bar{r}_1 \cdot t_c}{w_{\min}}$ is estimated, where $w_{\min}$ is the minimal bandwidth over all delivery paths for the VR video tiles within the viewport region. In the last step in sub-loop 1, lat_1 is compared to the maximum latency T_d. If lat_1 is no more than T_d, α will be refreshed to $\alpha - \Delta\alpha$and go to the

next step. Otherwise, the sub-loop 1 is terminated with the result R_1.

According to the cache placement result, the sub-loop 2 comes to select a set of bitrates $\mathbb{Z}_v = \{\Re_1, \Re_2, \cdots, \Re_{n_v}\}$ for tiles within viewport region satisfying $\sum_{i=1}^{n_v} \Re_i \leqslant R_1$ and $\Re_i \in x_{t,p_i,j}^{k,m,n} \cdot r_j,\ \forall p_i,\ \forall j,\ v_t^{k,m,n} \in V_t^{k,VP}$, where n_v is the number of tiles in viewport region. During the bit-rate selection for each tile, $\Re_i$ is determined by the routing decision variable $y_{t,p_i,j}^{k,m,n}$ and it is equal to $y_{t,p_i,j}^{k,m,n} \cdot r_j$. Hence, while we find the appropriate bit-rate of $\Re_i$ the corresponding request routing result $y_{t,p_i,j}^{k,m,n}$ is also obtained. The selection of $\mathbb{Z}_v$ is based on a hierarchical searching rule that the tile request is progressively forwarded to the upper-layer parent node when the appropriate tile whose bitrate satisfies the requirement is not locally available. Then the transmission latency is calculated as $lat_2 = \max_{1 \leqslant i \leqslant n_v} \left\{ \frac{\Re_i \cdot t_c}{w_i} \right\}$ and whether lat_2 is no more than T_d is judged next. If $lat_2 > T_d$, $\mathbb{Z}_v$ will be set to null. If $lat_2 \leqslant T_d$ and $\mathbb{Z}_v$ is null, α will be refreshed to $\alpha - \Delta\alpha$ and go to the sub-loop 1 for next search round. Otherwise, $\mathbb{Z}_v$ will be added to the result set $\Re$. In the following, a set of bitrates $\mathbb{Z}_{nv} = \{\Re_1, \Re_2, \cdots, \Re_{n_{nv}}\}$ for tiles outside the viewport region satisfying $\sum_{i=1}^{n_{nv}} \Re_i \leqslant R_2$ will be selected in terms of the hierarchical searching rule, and $\mathbb{Z}_{nv}$ is added to the result set $\Re$ that will be finally produced.

Algorithm 1. The proposed α-searching algorithm

3. Experimental results

To evaluate the performance of the proposed scheme, we developed a custom software in Java to run the caching optimization algorithm for quality-scalable VR video delivery over 5G networks. The test video data-set includes five video clips in JVET [33] and one hundred video clips that were downloaded from Youtube [34]. They are re-sampled with spatial resolution of 3840 × 1920. Besides, scalable video coding model SHM 12.0 [35] was used to encode the 360-degree VR videos for multi-bitrate tile caching. The test videos were encoded into three-layers with SNR (signal-to-noise ratio) scalability for different quantization parameter (QP) values of 25, 32 and 40.

We assume that the popularity of full frame VR video is in line with Zipf law and the Zipf parameter α_z is set to 0.75. The kth VR video is requested with the probability $\pi^k = \beta / k^{\alpha_z}$, where $\beta = \left(\sum_{k=1}^{\mathscr{K}} k^{-\alpha_z}\right)^{-1}$. The capacity ratio for each cache node is set to 60%.

In the caching system, the bandwidth information was dynamically configured in the simulation by a trace-driven methodology. The 5G channel model of 802.11ad [36] [37] was used to simulate the new radio features in the mobile caching system. The dynamic 5G network bandwidth trace data were collected from the 5G 802.11ad (WiGig) which provides multi-gigabit throughput. The details of the trace data collection methodology can be found in [37]. In this work, we collected two types of bandwidth trace data for one user in a multi-user channel setting. One is the stable bandwidth with average value about 700Mbps and another is the fluctuating bandwidth that ranges from 80Mbps to 700Mbps with different average values including 400Mbps, 500Mbps, and 600Mbps.

During VR video viewing, the user's viewport switches frequently. In the simulation, we assume 100 potential mobile users and their viewport requests obeyed a Poisson arrival and departure model with a mean inter-arrival time of 3 ms. The number of concurrent active requests was estimated by an M/M/∞queuing model [38] that follows Little's theorem [12] [39] with $N_r = \lambda_r \cdot T_a$, where T_a is the request active time and $1/\lambda_r$ is the mean request inter-arrival time. 50000 requests for viewports were simulated based on the popularity ranking of VR video tiles. The key experimental parameters in the simulation are shown in Table 1.

To verify the benefits of proposed joint scalable VR video tile caching (JTSVC) and quality-adaptive request routing scheme against the state-of-the-art solutions, we examined the following caching schemes.

Table 1
Experimental parameters.

Parameters	Values
Viewport size	1080 × 1200
Chunk length	1s
RAN cache number	40
Cache size per base station	10G
UE number per base station	100
T_d	12 ms
c_{p_0}	100
c_{p_i}	5
c_{p_i,p_j}	[2,10]
W_t	80Mbps-700Mbps
w_i	80Mbps-700Mbps
w_{min}	80Mbps-700Mbps
Request arrivals	Poisson, mean inter-arrival time per request = 3 ms, request active time = 200 ms

- JTSVC-α scheme: It is the proposed scheme that adopts cooperative 5GC and NG-RAN scalable VR video tile placement and the α-searching algorithm of viewport request routing.
- JTSVC-D scheme: For this scheme, the proposed joint 5GC and NG-RAN cache placement is used for scalable VR video tile caching. In the viewport request routing stage, the dynamic request routing algorithm [11] is used, where requests are first forwarded to the interconnected cache node and then the upper layer parent server when the content is not locally available. In dynamic request forwarding, once either of the multi-bitrate versions is found, the request will be regarded as being hit.
- LRU-α scheme: This scheme caches multi-bitrate VR video tiles with Least Recently Used (LRU) strategy and routes requests with α-searching algorithm.
- LRU-D scheme: This scheme caches multi-bitrate VR video tiles based on LRU strategy and routes requests with dynamic request forwarding algorithm.
- TL-α scheme: This scheme caches the layered VR video tiles based on minimizing the resolution error metric [20] and routes requests with our proposed α-searching algorithm.

Cache Hit Ratio. Cache Hit Ratio is the primary measurement metric of cache performance. It directly reflects the probability that the requested data resides in cache. Conventionally, cache hit ratio is computed in terms of full-frame video chunks. For VR video, the tile is taken as the basic unit in the cache system. Thus, request hit ratio is

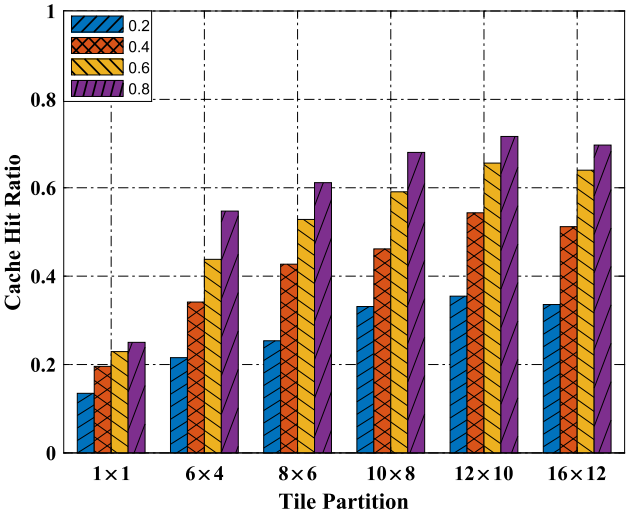

Fig. 3. The effect of tile size on cache hit ratio.

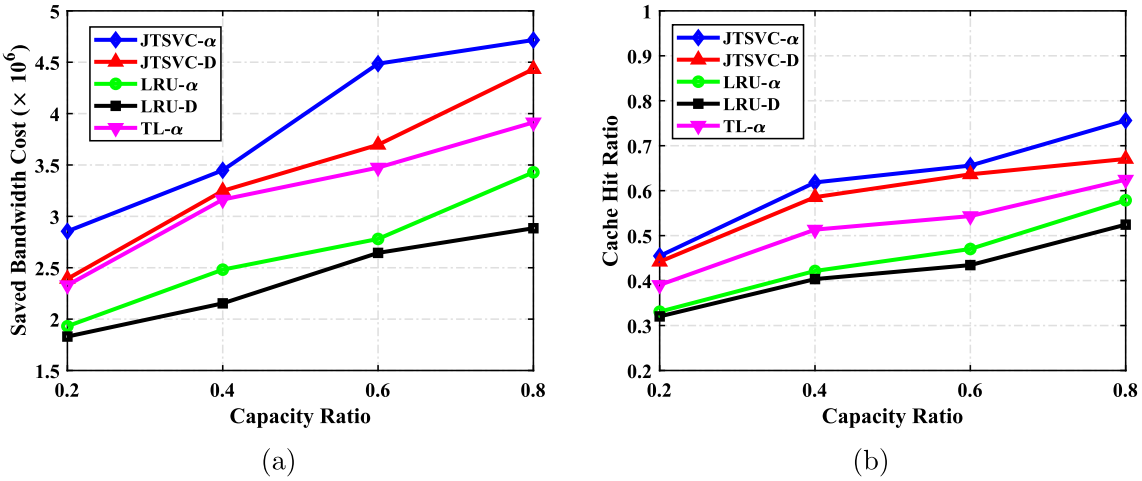

Fig. 4. Effect of cache capacity on caching performance.

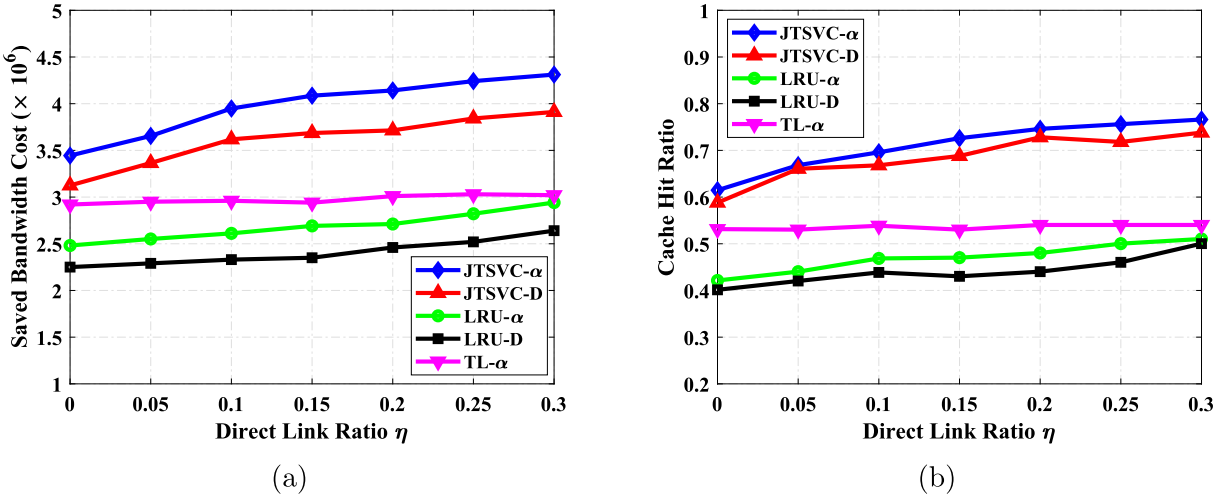

Fig. 5. Effect of direct connectivity between gNBs in 5G on caching performance.

counted on the tile. It is defined as the number of cache hits divided by the total number of requests for the tile.

3.1. VR Video Caching Performance

For tiled VR video encoding, the motion prediction range of the encoding tile is constrained in the reference image to support the random access capability of the full frame data stream. Specifically, the encoding block near the boundary of the tile will be not allowed to refer to the blocks outside the co-located tile in reference image [23]. This results in a minor loss in compression efficiency. The smaller tile size is, the lower the compression efficiency is. Different compression efficiency results in different data volume of each tile. On the other hand, tile size dictates viewport access flexibility in the whole frame. Thus, tile size affects not only the caching hit ratio but also the data volume that needs to be cached.

To test the effects of tile size on caching performance, VR videos were encoded in six different forms of tile partitions in the JTSVC-α scheme. With the full frame size of 3840×1920, the VR video was segmented into 1×1 (full frame), 6×4 (that means the full frame will be segmented into 6 tiles horizontally and 4 tiles vertically), 8×6, 10×8, 12×10 and 16×12. For different total cache capacities, the effects of the tile size on caching performance are different. The cache hit ratios were collected under a capacity ratio of 0.2, 0.4, 0.6 and 0.8 for different tile sizes. Fig. 3 shows the cache hit ratio results for different tile size caching experiments. It can be seen from Fig. 3 that the cache hit ratio gradually rises with the decreasing tile size (increasing tile numbers). The tile partition of 12×10 achieves the highest cache hit ratio. When the tile number rises to 16×12, the cache hit ratio decreases since the corresponding data size increases and results in a significantly negative effect on the cache hit ratio. As a result, the tile partition of 12×10 was adopted as ideal in our experiments. In Fig. 3, the 1×1 tile partition denotes the full-frame caching approach. The full-frame caching obtains the worst cache hit ratio among all partitions. It illustrates that the tile should be the basic cache data unit that exactly caters to the interactive partial data request feature of VR video, and thus the cache hit ratio of tile-based caching is significantly improved over full-frame caching.

To evaluate the effect of cache capacity on caching performance, several tests on different cache capacities were performed. The saved bandwidth cost and the cache hit ratio were measured with a set of capacity ratios varying from 20% to 80%. Fig. 4(a) shows the saved bandwidth cost for different schemes. In Fig. 4(a), larger cache capacity achieves more gains in bandwidth costs, and the proposed JTSVC-α scheme results in the most significant bandwidth cost savings among all five schemes, regardless of the cache capacity. Fig. 4 (b) plots the cache hit ratio for increasing capacity ratio under the five schemes. It can be

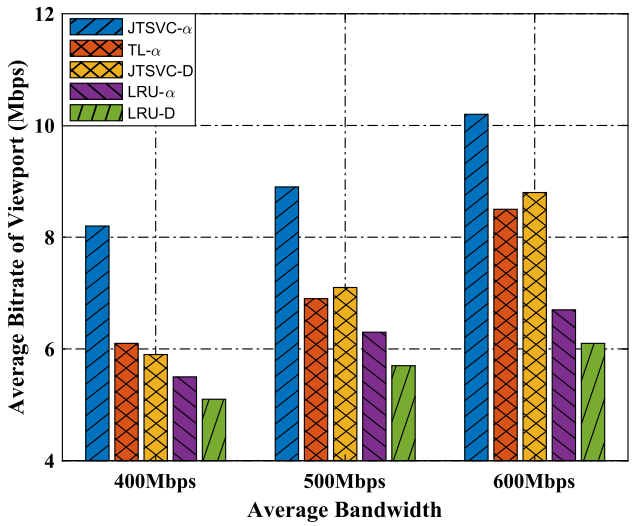

Fig. 6. Average bitrate of viewport that users received under different level of dynamic bandwidth.

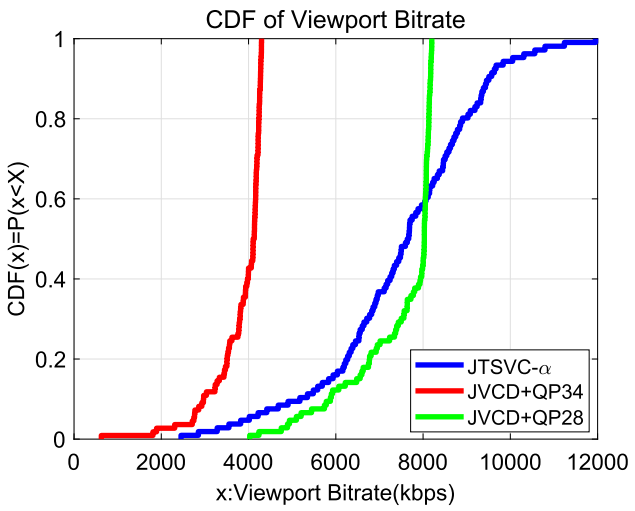

Fig. 7. CDF of bitrates of the requested viewports.

seen that from Fig. 4 (b) all the five schemes achieve higher cache hit ratio with increasing capacity ratio. With more cache space, more VR video tiles can be cached in the mobile network. It is natural that the cache hit ratio increases as a result. The TL-α scheme obtains lower cache hit ratio than the JTSVC-α scheme and JTSVC-D scheme, since it cached tiles in EPC and RANs independently. What is more important is that the gap in cache hit ratio among the proposed JTSVC-α scheme and the other schemes achieves the highest value for capacity ratio of 80%. This observation indicates that the proposed JTSVC-α scheme can obtain better performance for cache deployments with large capacity.

In the 5G SA architecture, an obvious innovation is the direct connectivity among gNBs. In the proposed multi-bitrate VR video tile caching scheme, the cache cooperation among the connected gNBs is supported with the Xn interface communication. We define a direct link ratio indicator η to denote the ratio between the number of direct links and that of all possible direct links in the system. Fig. 5 shows the saved bandwidth cost and the cache hit ratio under different direct link ratio η and a cache capacity ratio of 0.4. It can be seen from Fig. 5 that the saved bandwidth cost and the cache hit ratio are gradually increasing when the direct link ratio increases. It illustrates that the proposed JTSVC-α scheme can benefit more from more direct connection links among gNBs. Moreover, the JTSVC-α scheme achieved slightly higher saving bandwidth cost than the JTSVC-D scheme for any η value. This is because the JTSVC-α scheme usually selects the suitable bitrate for the tile being requested and this saves more bandwidth compared to a non-adaptive tile selection in JTSVC-D. Contrary to JTSVC-α scheme, the TL-α scheme is almost not affected by the direct link ratio, since it does not collaboratively cache the tiles in 5G networks.

3.2. VR Video Streaming Performance

The viewport request arrival rate has a significant impact on the processing load and delay (the delay for buffering and scheduling requests) in the cache system. When the load increases, the bandwidth for each user will be reduced. Hence, the bandwidth implicitly shows the state of processing load. In our simulation we assume that the cache nodes have enough computational capability and neglect the impact of the processing delay on performance. We simulate w_i with time-varying bandwidth in different average values of 400Mbps, 500Mbps, and 600Mbps with the actual trace data. Thus, the simulation of dynamic bandwidth implicitly captures the effect of the request arrival rate on the streaming performance. Fig. 6 shows the average bitrate of viewport that received by users under different dynamic bandwidths in the cache system. It can be seen from Fig. 6 that the performance gap between the proposed scheme and other schemes are gradually reduced with increasing link bandwidth. It indicates that the proposed scheme has much more improvement room for lower link bandwidth conditions. In Fig. 6, the TL-α scheme received more bit-rates of viewport than the JTSVC-D scheme at the low available bandwidth (400Mbps). It is because TL-α scheme considered the perception-based bitrate allocation during caching optimization. At the low bandwidth regime, TL-α requested as much high-quality tiles of viewport as possible. Comparably, at the high bandwidth regime, the collaborative caching played a major role in viewport quality improvement, and hence the JTSVC-D scheme received more bit-rate of viewport than the TL-α scheme.

To evaluate the effectiveness of the quality-adaptive routing scheme against the constant-quality tiled video caching (TVC) scheme, two groups of TVC schemes were tested with constant QPs (QP = 28 and QP = 34) under a fluctuating bandwidth ranging from 80Mbps to 700Mbps with an average value of 400Mbps. In our simulation, all the schemes used the same cache space. The Cumulative Distribution Function (CDF) of the viewport bitrate and viewport delivery latency are shown in Figs. 7 and 8, respectively.

It can be seen from Fig. 7 that the viewport bitrates reached a maximum value of 4200kbps and 8100kbps respectively for the two fixed-bitrate TVC schemes with dynamic request routing (TVCD). The bitrates for a minority of viewports were lower than 4200kbps and 8200kbps for the two fixed-bitrate schemes, respectively. Specifically, about 50% of viewport bitrates are lower than 4200kbps for the TVCD + QP34 scheme and about 40% of viewport bitrates are lower than 8100kbps for the TVCD + QP28 scheme. In comparison, the bitrates of requested viewports of the JTSVC-α scheme are mostly in the range of 6000kbps and 10000kbps. This is because the JTSVC-α scheme caches the multi-bitrate versions and provides the adaptive bitrates of the requested tiles to the dynamic wireless channel. Consequently, the JTSVC-α scheme can achieve higher viewport qualities than the other constant bitrate caching schemes.

Fig. 8 shows the CDF of the viewport latencies. It can be seen that the latencies of the requested viewports for the TVCD + QP28 scheme are at most up to 14 ms. This is because a few high bitrate tiles cannot be delivered to UEs under the MTP latency constraint of 20 ms (including 6 ms reserved for rendering) over the time-varying channel. In these cases, the delays that exceed 14 ms are all considered to be equal to 14 ms. For the TVCD + QP34 scheme, the delivery delays for the requested tiles are almost all lower than 14 ms, albeit with lower viewport quality. Similarly, even though some of tile latencies of the JTSVC-α scheme are greater than those of the TVCD + QP34 scheme, they are still lower than

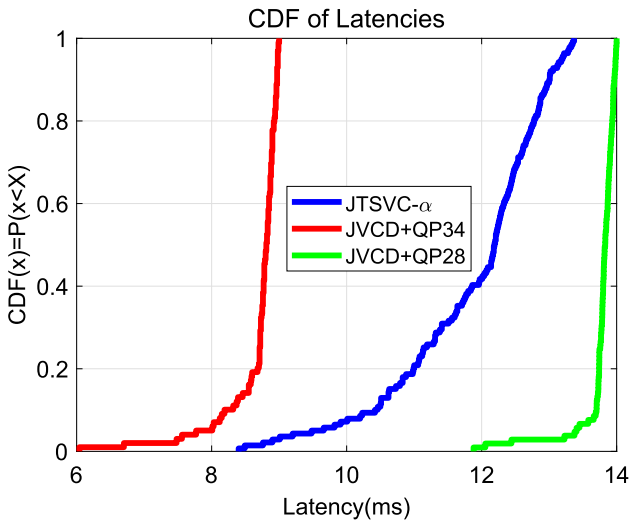

Fig. 8. CDF of viewport latencies.

14 ms. This indicates that the JTSVC-α scheme provides UEs appropriate qualities of the viewports to satisfy the strict latency requirement.

4. Conclusion

In this paper, a scalable VR video tile caching scheme for 5G mobile networks is proposed. By considering the particular characteristics of VR video, different quality layers of VR video are first segmented spatially into tiles with different bitrate versions. Then, the multi-bitrate tiles are cooperatively cached into 5GC and NG-RANs to reduce the VR video streaming latency. On top of our multi-bitrate VR video tile caching scheme, we propose a quality-adaptive request routing scheme that is used to ensure the requested VR video bitrate can accommodate mobile network fluctuations. In request routing, unequal bitrate allocation among tiles in the whole frame is performed to optimize the homogeneous viewport quality under a channel bandwidth constraint. Experimental results show that the proposed scheme can increase the cache hit ratio and save more bandwidth for VR video streaming than other constant bitrate caching schemes over 5G networks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China under Grant 61771469.

References

[1] S. Ohl, M. Willert, O. Staadt, Latency in distributed acquisition and rendering for telepresence systems, IEEE Trans. Visual. Comput. Graphics 21 (12) (Dec. 2015) 1442–1448.
[2] A. Prasad, M. Uusitalo, D. Navrátil, M. Säily, Challenges for enabling virtual reality broadcast using 5G small cell network, in: 2018 IEEE Wireless Communications and Networking Conference Workshops, IEEE, 2018.
[3] Abari O., Bharadia D., Duffield A., Katabi D., Enabling High-Quality Untethered Virtual Reality, 14th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, March 27-29, 2017.
[4] Y. Liu, J. Liu, A. Argyriou, S. Ci, MEC-assisted Panoramic VR Video Streaming over Millimeter Wave Mobile Networks, IEEE Transactions on Multimedia 21 (5) (May 2019) 1302–1316.
[5] R. Schmoll, S. Pandi, P.J. Braun, F.H.P. Fitzek, Demonstration of VR/AR offloading to Mobile Edge Cloud for low latency 5G gaming application, in: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2018.
[6] J. Zhao, R.S. Allison, M. Vinnikov, S. Jennings, Estimating the motion-to-photon latency in head mounted displays, in: Proc. 5th IEEE Virtual Reality, Mar., 2017.
[7] Xie G., Li Z., Ali Kaafar M., Wu Q., "Access types effect on internet video services and its implications on CDN caching, IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 5, pp. 1183–1196, January 2017.
[8] O.E.A. Franky, D. Perdana, R. Negara, D. Sanjoyo, G. Bisono, System design, implementation and analysis video cache on internet service provider, in: 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), IEEE, 2016.
[9] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.
[10] M. Dehghan et al., "On the complexity of optimal routing and content caching in heterogeneous networks, Proc. IEEE INFOCOM, pp. 1–14, Apr. 2015.
[11] J. Dai, Z. Hu, B. Li, J. Liu, B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution, Proc. IEEE Conf. Comput. Commun. (Infocom), pp. 2444–2452, Mar. 2012.
[12] H. Ahlehagh, S. Dey, Video-aware Scheduling and Caching in the Radio Access Network, IEEE Transactions on Networking 22 (5) (Oct. 2014) 1444–1462.
[13] N. Golrezaei, K. Shanmugam, A.G. Dimakis, A.F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in Proc. IEEE INFOCOM, Mar. 2012, pp. 1107–1115.
[14] J. Qiao, Y. He, X. Shen, Proactive caching for mobile video streaming in millimeter wave 5G networks, IEEE Transactions on Wireless Communications 15 (10) (August 2016) 7187–7198.
[15] X. Wang, M. Chen, T. Taleb, A. Ksentini, V.C.M. Leung, Cache in the air: Exploiting content caching and delivery techniques for 5G systems, IEEE Commun. Mag. 52 (2) (Feb. 2014) 131–139.
[16] Y. Sun, Z. Chen, M. Tao, H. Liu, Communication, computing and caching for mobile VR delivery: Modeling and trade-off, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018.
[17] Sukhmani S., Sadeghi M., Erol-Kantarci M., El Saddik A., Edge Caching and Computing in 5G for Mobile AR/VR and Tactile Internet, IEEE MultiMedia, vol. 26, no. 1, pp. 21–30, Jan.-March 1, 2019.
[18] K. Liu, Y. Liu, J. Liu, A. Argyriou, Y. Ding, Joint EPC and RAN caching of tiled vr videos for mobile networks, in: International Conference on Multimedia Modeling, Springer, 2019, pp. 92–105.
[19] A. Mahzari, A. Nasrabadi, A. Samiei, R. Prakash, FoV-aware edge caching for adaptive 360 video streaming, in: 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 173–181.
[20] G. Papaioannou, I. Koutsopoulos, Tile-based Caching Optimization for 360 Videos, in: Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, 2019, pp. 171–180.
[21] V.R. Gaddam, M. Riegler, R. Eg, P. Halvorsen, Tiling in interactive panoramic video: Approaches and evaluation, IEEE Trans. Multimedia 18 (9) (Sep. 2016) 1819–1831.
[22] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, T. Schierl, HEVC tile based streaming to head mounted displays, in: 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2017.
[23] C. Concolato, J. Le Feuvre, F. Denoual, E. Nassor, N. Ouedraogo, J. Taquetet, Adaptive streaming of HEVC tiled videos using MPEG-DASH, IEEE Transactions on Circuits and Systems for Video Technology 28 (99) (Aug. 2017) 1981–1992.
[24] 3rd Generation Partnership Project (3GPP), Technical Specification Group Services and System Aspects, Vocabulary for 3GPP Specifications (Release 15), 3GPP TR 21.915 V1.1.0, Mar. 2019.
[25] ETSI, "Mobile Edge Computing - Introductory Technical White Paper, Sep. 2014.
[26] J. Poderys, M. Artuso, C. Michael Oest Lensbol, H. Lehrmann Chris-tiansen, J. Soler, "Caching at the Mobile Edge: A Practical Implementation, IEEE Access, vol. 6, pp. 8630-8637, 2018.
[27] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: Evidence and implications, Proc. IEEE 18th Annu. Joint Conf. Comput. Commun. Soc. 1 (1999) 126–134.
[28] V. Sitzmann, et al., Saliency in VR: how do people explore virtual environments? IEEE Trans. Vis. Comput. Graph. 24 (4) (2018) 1633–1642.
[29] Y. Ding, Y. Liu, J. Liu, K. Liu, et al., Panoramic Image Saliency Detection by Fusing Visual Frequency Feature and Viewing Behavior Pattern, Pacific-Rim Conference on Multimedia, Hefei, China, Sep 21–22 (2018).
[30] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, Chichester, U.K., 1990.
[31] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[32] N. Hu, P. Steenkiste, Evaluation and Characterization of Available Bandwidth Probing Techniques, IEEE J. Selected Areas in Comm. 21 (6) (Aug. 2003) 879–894.
[33] J. Boyce, E. Alshina, A. Abbas, et al., "JVET-D1030 r1: JVET Common Test Conditions and Evaluation Procedures for 360° Video, 2016.
[34] [Online]. https://www.youtube.com/channel/UCzuqhhs6NWbgTzMuM09WKDQ. Accessed on: July 1, 2021.

[35] SHVC Reference Software SHM. [Online]. https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/. Accessed on: July 1, 2021.
[36] 3GPP, Study on channel model for frequencies from 0.5 to 100 GHz, 3GPP, Tech. Rep. 38.901 V14.3.0, Dec. 2017.
[37] L. Sun, et al., Multi-path multi-tier 360-degree video streaming in 5G networks, in: Proc. ACM Multimedia Syst. Conf. (MMSys), 2018, pp. 162–173.
[38] D.P. Bertsekas, R.G. Gallager, P. Humblet, Data Networks 2, Prentice-Hall International, New Jersey, 1992.
[39] J.D.C. Little, S.C. Graves, law Little's, International Series in Operations Research & Management Science, 115, USA:Springer-Verlag, New York, NY, 2008, pp. 81–100.
[40] Afzal S., Chen J., Ramakrishnan K.K., "Characterization of 360-degree Videos, in Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, August 25, 2017, Los Angeles, CA, USA.
[41] G.S. Paschos, G. Iosifidis, M. Tao, D. Towsley, G. Caire, The Role of Caching in Future Communication Systems and Networks, IEEE Journal on Selected Areas in Communications 36 (6) (June 2018) 1111–1125.
[42] P.C. Chu. A Genetic Algorithm Approach for Combinatorial Optimization Problems, Ph.D. thesis at The Management School, Imperial College of Science, London, 1997.
[43] G.R. Raidl, An improved genetic algorithm for the multiconstrained 0–1 knapsack problem, in: Proceedings of the 5th IEEE International Conference on Evolutionary Computation, IEEE Press, 1998, pp. 207–211.