用于降低点对点CDN中视频传输成本的最优短视频排序与传输调度

Zhipeng Gao[a], Chunxi Li[a] 和YongxiangZhao[a], *

[a]电子与信息工程学院，北京交通大学，中国北京

文章信息

关键词：短视频流媒体 点对点CDN 传输成本优化 短视频排序 传输调度 最小费用最大流

摘 要

短视频平台的爆炸式增长带来了全球流量的巨大激增，使内容提供商承受沉重的财务负担。点对点内容分发网络（PCDN）通过利用资源受限的边缘节点提供了一种高性价比的替代方案，但这些对等节点有限的存储空间和并发服务容量，难以应对短视频消费所特有的剧烈时序需求峰值。本文提出利用一种新颖的自由度——服务器驱动播放序列所固有的灵活性——来降低传输成本。我们将最优视频排序与传输调度（OVOTS）问题建模为整数线性规划，以联合优化个性化视频排序和传输调度。通过策略性地调整播放列表顺序，我们的方法能够主动平滑时序流量峰值，最大限度地将请求卸载至低成本对等节点。为求解OVOTS问题，我们从理论上将其严格归约为一个辅助的最小费用最大流（MCMF）问题。借助柯尼希边染色定理，我们证明了这两种形式的严格等价性，并提出带边着色的最小费用最大流（MMEC）算法，这是一种全局最优的多项式时间解。大量仿真表明，MMEC显著优于基线策略：与随机调度相比，成本最多降低67%；与模拟退火方法相比，成本最多降低36%。我们的研究表明，播放序列灵活性是PCDN架构中一种稳健且极为有效的成本优化范式。

arXiv:2603.03938v1 [cs.NI] 4 Mar 2026

1. 引言

短视频平台（如TikTok和Instagram Reels）的爆发式增长彻底改变了数字内容消费方式，导致全球视频流量出现前所未有的激增 [1, 2]。为应对这一激增，内容提供商传统上通过内容分发网络（CDN）分发视频数据 [3, 4],，从而产生高昂成本以满足用户需求。例如，抖音每日活跃用户超过10亿，年流量成本超过60亿元人民币 [5]。为降低这些巨额开支，内容提供商正越来越多地采用高性价比的点对点内容分发网络（PCDN）架构 [6]。通过将低成本、资源受限的对等节点（例如家用路由器和机顶盒）与稳健但昂贵的传统内容分发网络节点相结合，内容提供商可以有策略地将部分视频流量卸载到对等网络，从而降低总体传输成本 [7, 8]。

然而，由于低成本对等节点的带宽和存储限制与短视频的突发访问模式之间存在根本性的不匹配，在PCDN架构中降低短视频传输成本仍然面临挑战。一方面，对等节点通常是消费级设备，天然受限于有限的上行带宽和存储容量。 [9] 例如，超过80%的此类节点平均服务速度低于5兆比特每秒，而速度超过8兆比特每秒的节点不足1%。 [10]此外，它们通常依赖于剩余的主机设备的存储空间，因而其缓存容量仅为吉字节量级 [12]。另一方面，短视频往往会在短时间内收到大量并发请求，随后流行度迅速下降 [1, 13,14]。根据我们在第2节中的测量结果，短视频的流行度通常会在30分钟内消退。因此，这些低成本节点的并发服务能力完全不足以应对这种短时间内集中的请求尖峰。超出的请求不可避免地会被转回成本高昂的CDN节点，从而严重削弱PCDN架构的经济优势。

为降低整体视频传输成本，现有方案主要通过主动缓存和优化传输调度来最大限度地卸载流量。首先，主动缓存策略 [15, 16, 17] 严重依赖对视频流行度的预测，随后主动更新缓存。然而，鉴于短视频流行度变化迅速，这种方法可能导致频繁的缓存替换，迅速耗尽对等节点有限的 I/O 和带宽资源。其次，传统传输调度 [6, 10, 11] 会将视频请求负载均衡地分配至可用的对等节点，但通常只能被动响应用户固定的视频观看需求。因此，面对短视频在全系统范围内出现的大规模突发访问模式，这两种策略的效果都有限，因为它们无法改变用户需求在时间上的固有集中性。

本文从一个全新的角度着手降低短视频传输成本，利用短视频播放序列固有的灵活性，

*通讯作者zhpgao@bjtu.edu.cn （Z. Gao）；chxli@bjtu.edu.cn （C. Li）；yxzhao@bjtu.edu.cn （Y. Zhao）
ORCID:

将其转化为一种新的优化自由度。这一思路利用了短视频平台由服务器驱动的特性：[18, 19], 中央服务器会定期向用户推送预先设定的播放列表，以确定具体的推荐视频、播放序列以及每个视频指定的传输节点[1] [14, 20, 21]。关键在于，由于该播放序列由服务器端生成，服务器可以策略性地调整个性化播放列表中视频的顺序，而不改变实际推荐视频的集合，从而确保优化过程对用户透明。为确保优化对用户保持透明，排列调整严格限于权重相同的推荐内容，因此能够保持基线用户体验质量（QoE）。利用这种隐含的灵活性，我们提出一种联合优化框架，同时实现以下目标：1）错开不同用户播放相同视频的时间，以主动平滑时序需求峰值；2）智能地为每个视频分配最优对等节点，以最大限度地降低总体传输成本。通过削平高度集中的流量峰值，我们的框架能够将原本会溢出至高成本CDN节点的请求有效地交由并发服务容量有限的低成本对等节点处理。

为实现该方法，我们抽象短视频传输架构的核心组件，将其建模为一个系统，该系统包括充当控制器的中央服务器、多个短视频用户、一组低成本且容量受限的对等节点，以及一个稳健但成本较高的CDN节点，用于服务超出对等节点容量的请求。为从分析上刻画观看需求与PCDN服务容量，我们采用离散时隙模型，假设每个用户在每个时隙[2]请求并观看一个短视频。相应地，PCDN节点的并发服务容量定义为该节点在单个时隙内能够服务的视频请求的最大数量。中央服务器基于两项主要输入运行：用户-视频推荐集合（为每个用户指定的具体视频）和节点-视频存储信息（缓存在每个PCDN节点上的视频以及PCDN节点的并发服务容量）。基于这些输入，服务器为每个用户生成个性化播放列表，该列表不仅预先确定播放顺序，还为每个视频指定传输节点。

基于已建立的系统模型，我们将最优视频排序与传输调度（OVOTS）问题表述为一个整数线性规划（ILP），旨在通过确定每位用户的最优播放序列以及每个视频对应的节点分配，最小化总传输成本。为克服标准 ILP 求解器固有的计算复杂性，我们严格证明，OVOTS问题可等价地归约为一个辅助的最小费用最大流（MCMF）问题 [22]，方法是将并发服务容量映射为流网络中的链路容量。此外，利用二分图上的 König 边着色定理 [23]，我们证明，MCMF问题的任何最优解都可以无损地转换为原始 OVOTS问题的最优解。在这一理论基础的指导下，我们提出带边着色的最小费用最大流（MMEC）算法，该算法分两个严格按序进行的阶段运行。首先，调度阶段基于 Bellman-Ford算法 [24]，使用逐次最短路径算法求解 MCMF 实例，以确定最优用户—节点配对；随后，排序阶段对已确定的配对采用 Kempe 链着色算法 [25]，将视频映射到特定时隙（即生成最终播放序列）。通过利用该问题表述特有的数学结构，这种解耦方法能够保证获得全局最优的多项式时间解，相较于传统的指数时间 ILP 求解器具有显著的可扩展性优势。

最后，为了评估所提联合优化框架的有效性，我们使用遵循 Zipf 流行度分布的合成轨迹开展了大量仿真实验，以忠实模拟现实世界中的短视频访问模式。具体而言，我们将核心算法 MMEC 与三种具有代表性的基线方法进行比较：随机排序与随机传输调度（RORS）、随机排序与最优传输调度（ROOS），以及模拟退火优化（SAO）。实验结果表明，与 RORS、ROOS 和 SAO 相比，MMEC 始终表现更优，整体传输成本最高分别降低了 67%、36% 和 36%。此外，我们的分析显示，所提算法对于用户规模、视频库大小和对等节点数量等关键系统参数的变化具有很强的鲁棒性。这些发现凸显了视频排序与传输调度联合优化作为一种关键范式，在优化 PCDN架构中的短视频交付成本方面具有巨大潜力。

本文其余部分组织如下。第2 节介绍点对点内容分发网络架构的背景，并通过分析短视频的突发流量访问模式阐明本研究的动机。第3节建立系统模型并对 OVOTS 问题进行建模。第4节进行详细的理论分析，包括将问题归约为辅助的最小费用最大流模型，并介绍 MMEC 算法。第5节介绍实验设置，并通过数值仿真评估算法性能。最后，第6 节总结全文，并概述未来研究的潜在方向。

[1]虽然短视频系统通常将推荐引擎与传输调度器分离，但我们的框架将节点指定与播放列表生成紧密结合，以便实现数学上易于处理的联合优化。[2]这一假设是出于数学上的可处理性考虑。在实践中，可以将视频细化为均匀的视频分块，并相应地将时隙与单个分块的播放时长对齐。

2. 背景与动机

2.1. 点对点内容分发网络

点对点内容分发网络已成为一种极具前景的架构，可降低整体视频传输成本 [6, 7]。随着短视频流量激增 [1, 2, 13]，，传统内容分发网络所需的巨额硬件、网络带宽和维护成本 [3, 4]已成为内容提供商难以承受的财务负担 [5, 10]。传统内容分发网络依赖昂贵的商用数据中心，而点对点内容分发网络的核心理念则是利用分布在网络边缘、零散且未充分利用的资源 [26, 27]。这些边缘节点主要由从第三方提供商租用的闲置计算资源构成，例如智能家居设备、家用路由器和机顶盒。因此，点对点内容分发网络在成本效率方面具有显著优势。

尽管具有这种成本效益，但每个对等节点的资源容量都受到严重限制，这给短视频传输带来了新的挑战。具体而言，单个对等节点的局限性主要体现在两个方面：并发服务容量和存储容量。根据近期对商业 PCDN 系统的大规模测量 [10, 11]，，绝大多数对等节点的服务速度都非常低。例如，超过 80% 的对等节点提供的平均服务速度低于 5 兆比特每秒，而能够超过 8 兆比特每秒的节点不足 1% [10]。相比之下，专用内容分发网络服务器可以轻松提供数十至数百吉比特每秒的吞吐量 [28]。此外，与配备海量企业级磁盘阵列（通常容量处于太字节或拍字节范围）的集中式内容分发网络服务器不同 [3, 4, 28]，，单个对等节点必须完全依赖主机设备的剩余存储空间，这通常将其缓存容量限制在吉字节量级 [12]。

因此，这些根本性的双重限制严重阻碍了有效的视频传输优化，尤其是在应对大规模流量峰值时。首先，严格的并发带宽限制意味着单个对等节点只能服务极少量的并发请求。当高度热门的短视频引发突发流量时，这些对等节点有限的并发服务容量会瞬间饱和，从而导致严重拥塞。其次，严格的存储限制决定了单个对等节点只能缓存整个视频库中极小的一部分。因此，对等节点在物理上不可能维持稳定而全面的热门内容工作集。这种固有的稀缺性导致缓存未命中率持续居高不下，尤其是在用户需求快速变化时。每当所请求的视频不在本地缓存中时，传输就必须绕过对等网络，回退到昂贵的CDN节点。这种被迫回退直接抵消了流量卸载效率，最终推高了PCDN架构原本旨在最小化的总体传输成本。

表1用于测量的24小时数据集基本统计信息

数据集	记录总数	独立用户数	独立视频数
KuaiSAR	770,387	8,398	379,055
私人	24,222,188	255,253	3,198,137
PPTV	17,873,041	2,712,701	116,252

2.2. 短视频的闪购式模式

在实时推荐引擎和基于滑动的连续播放界面的驱动下，短视频平台相较于传统视频点播系统（如 YouTube [29]和Netflix [30]）呈现出显著的突发访问模式。具体而言，一条短视频往往会在短暂的时间窗口内遭遇大量并发请求，随后流行度迅速下降。为定量说明这一特征，我们使用三个不同的数据集开展了数据驱动的测量研究。KuaiSAR [31]是从快手采集的公开短视频数据集 [32]；私有数据集是由主流内容提供商提供的短视频轨迹；PPTV则是来自PPTV的传统视频点播轨迹 [33]。这些24小时数据集的基本统计信息汇总于表1。

为捕捉强烈的时间局部性，我们首先将24小时划分为2小时时间间隔，并提取每个时间间隔内请求频率最高的前0.1%视频。随后，我们计算 [34]这些热门视频集合在不同时间窗口之间的杰卡德相似度。所得相似度矩阵以图1中的热力图呈现，揭示了短视频服务与传统视频点播服务在请求模式上的显著差异。短视频数据集中的杰卡德相似度即使在相邻时隙之间也会骤然下降。相比之下，在传统视频点播数据集中，相似度在全天范围内始终保持异常高的水平。为进一步揭示这种闪购式现象的微观严重程度，我们放大观察私人数据集的晚间高峰时段（例如21:30至23:20），并以10分钟的细粒度间隔评估杰卡德相似度（如图2所示）。值得注意的是，相似度仅在20至30分钟内就降至0.5以下。这一实证观察证实，短视频突发流量的生命周期极其短暂。

突发流量访问模式与严格的存储限制相结合，使传统的缓存优化方法效果甚微。为追踪快速变化的趋势而不断更新缓存，会导致严重的缓存抖动，迅速耗尽对等节点稀缺的带宽，并抵消 PCDN 的经济效益。因此，单纯依靠空间路由或反应式缓存无法阻止大量流量溢出至成本高昂的 CDN节点。为此，我们希望利用短视频平台所固有的一种独特自由度——服务器驱动的推荐播放列表。通过智能调整用户的时间播放序列，我们可以直接在时间域进行干预，

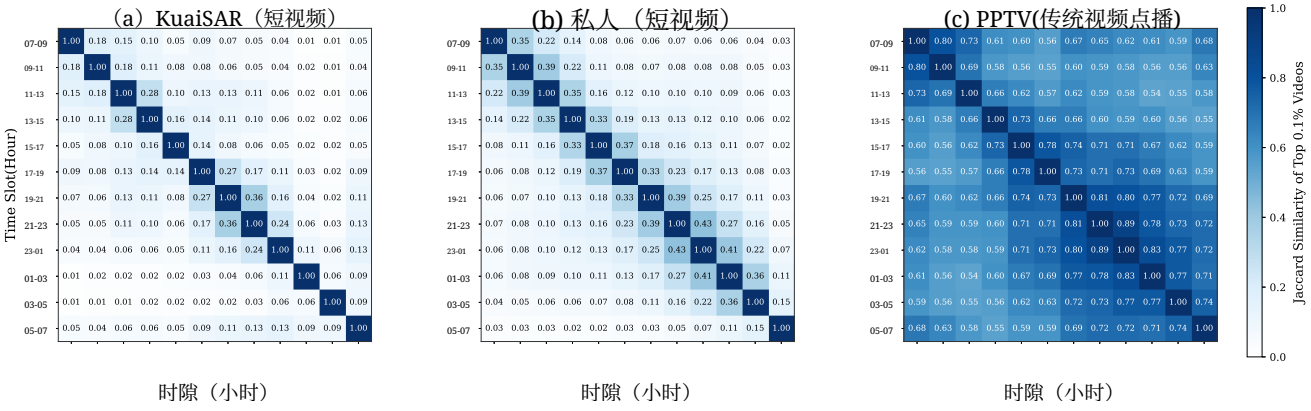


图1： 短视频与传统视频点播数据集中，跨2小时时隙的请求量最高0.1%视频的杰卡德相似度，展示了短视频相对于传统视频点播的闪现式请求模式和快速受欢迎度衰减。

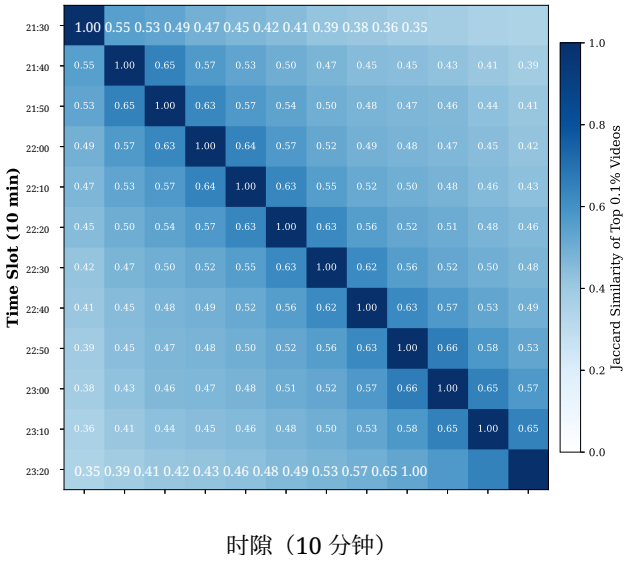


图2： 晚间高峰时段（21:30–23:20）内，跨细粒度10分钟时隙的请求量最高0.1%视频的杰卡德相似度。相似度在短30分钟内降至0.5以下，展示了短视频突发流量。

主动塑造需求曲线，自然而然地平滑并发流量峰值。

3. 系统模型与问题表述

本节介绍我们联合优化方法的理论框架。首先，我们详细说明应用场景，阐释如何通过优化播放序列来最小化总体传输成本。随后，我们对系统模型进行形式化描述，并将最优视频排序与传输调度（OVOTS）问题表述为整数线性规划（ILP）。最后，为克服标准 ILP 固有的指数复杂度，我们引入一种节点拆分变换，将 OVOTS 转化为一种计算上易于处理的辅助形式。

3.1. 应用场景

本文研究的短视频流媒体系统通过视频排序与传输调度的联合框架进行优化，旨在利用低成本资源尽可能满足用户的视频观看需求，从而使总传输成本最小化。如图3所示，该系统架构由控制平面和数据平面组成，包含三个主要实体：短视频服务器、短视频用户和PCDN节点。

控制平面中的短视频服务器充当中央控制器。给定用户-视频推荐信息和节点-视频存储信息后，短视频服务器执行MMEC算法，为每位用户生成个性化播放列表，并确定从哪个存储节点下载哪个视频。具体而言，用户-视频推荐信息记录每位用户打算观看的视频ID，而节点-视频存储信息记录每个PCDN节点缓存的视频ID以及各PCDN节点的并发服务容量。这两者通常是由上游推荐与缓存算法确定的外部输入，不在本文的研究范围内。

用户按照短视频服务器生成的播放列表，依次下载并观看短视频。本文重点在于最大限度地降低视频传输成本，因此假设分发播放列表的传输时延可以忽略不计。此外，我们假设所有用户均匀地观看视频。观看过程被划分为连续的时隙（例如 $t = 1$、2、3，如图 3所示）。在每个时隙内，用户从预先分配的PCDN节点下载所需视频并观看，并在时隙结束时通过屏幕滑动无缝切换到下一个视频。

数据平面中的PCDN节点根据其存储容量、并发服务能力和成本进行分类。第一类包括对等节点（例如， n_1, n_2），

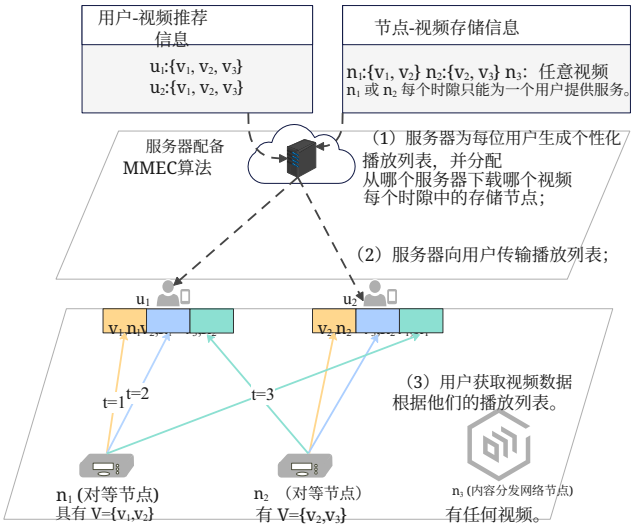


图3： 所提出的联合优化框架的系统架构和动机示例。

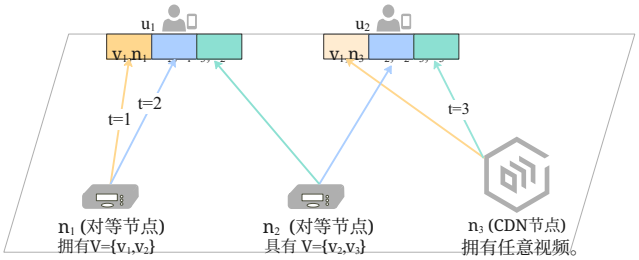


图4： 相同时间序列下发生容量冲突的示例。

例如通常从第三方提供商处租用的家用路由器或机顶盒。这些节点的存储空间和并发服务容量都受到严格限制，只缓存少量视频，并且每个时隙只能同时为有限数量的用户提供服务。第二类是高可靠CDN节点（例如，n_3），假设其拥有近乎无限的存储空间和带宽，能够同时为所有用户提供服务。然而，CDN节点产生的传输成本显著高于对等节点。

为了说明我们的联合优化如何降低服务成本，请看图3和图4中的动机示例。用户 u_1 和 u_2 都打算观看同一组推荐视频 $\{v_1, v_2, v_3\}$。如果服务器为两位用户分配相同的时间顺序序列 $< v_1, v_2, v_3 >$（如图4所示），就会发生容量冲突：在 $t = 1$，两者都试图获取 v_1，但 n_1 一次只能为一位用户提供服务，而且 n_2 没有缓存 v_1。因此，服务器不得不将 u_2的请求重定向到成本高昂的CDN节点 n_3。在$t = 3$也会发生类似的冲突。相比之下，通过采用我们提出的优化方法，服务器将 u_2的播放列表重新排序为优化序列 $< v_2, v_3, v_1 >$，如图3所示。这种策略错峰安排确保所有时隙中的视频请求都由低成本对等节点 n_1 和 n_2处理，从而有效消除对成本高昂的CDN节点的需求，并最大限度地降低总体传输成本。

因此，针对这样一个短视频流媒体系统，我们的目标是在满足PCDN节点的存储和容量约束的同时，为每个用户找到最优播放列表，并为每个视频分配一个节点，从而最小化总服务成本。

3.2. 系统描述

为了对系统进行形式化建模，我们首先定义用于描述用户和视频内容的集合。令 $\mathcal{U} = \{1, 2, \ldots, U\}$ 表示系统中的用户集合，其中U 表示用户总数。内容库由短视频文件集合 $\mathcal{V} = \{1, 2, \ldots, V\}$表示。为便于分析，我们假设每个视频的大小均匀。此外，观看需求通过用户-视频推荐信息来表示，记为集合 $\mathcal{R} = \{R_u \mid u \in \mathcal{U}\}$。其中，每个元素 R_u 都是 $\mathcal{V}$ 的一个子集，表示用户 u 打算观看的具体视频。

接下来，我们描述数据平面中的PCDN节点。用 $\mathcal{N} = \{0, 1, 2, \ldots, N\}$表示所有可用节点的集合。为区分节点的物理属性和服务成本，我们将节点0至 $N-1$ 定义为低成本对等节点，而最后一个节点则作为可靠的高成本CDN节点。网络中的视频缓存状态由节点-视频存储信息集合 $\mathcal{S} = \{S_n \mid n \in \mathcal{N}\}$定义，其中每个元素 S_n 指定了节点 n缓存的视频子集。具体而言，假设CDN节点存储整个视频库，即 $S_N = \mathcal{V}$。相比之下，对等节点的存储容量有限，只能缓存少量视频数据。

这些节点的运行约束由其各自的服务能力和传输成本决定。令 c_n 表示节点 n在单个时隙内向一名用户交付视频时所产生的传输成本。如前所述，CDN节点的成本显著高于任何对等节点，即对所有 $n < N$均有 $c_N \gg c_n$。此外，每个节点 n受到最大并发容量 d_n的限制，该容量表示其在单个时隙内可同时服务的最大用户数量。对等节点通常具有非常有限的容量 d_n，而CDN节点则被假定拥有足够的容量，以便在需要时并发服务 $\mathcal{U}$ 中的所有用户，$d_N = |\mathcal{U}|$。

最后，我们对短视频流媒体系统在有限时间范围内的时间动态进行建模，该时间范围被划分为一组离散时隙 $\mathcal{T} = \{1, 2, \ldots, T\}$。不失一般性，为确保能够将每个推荐视频映射到一个不同的时隙，我们假设每位用户的推荐集合大小等于该时间范围内的时隙总数，即$|R_u| = T$，$\forall u \in \mathcal{U}$。整个过程分为调度阶段和流媒体传输阶段。假设调度阶段在控制平面内进行

表 2所用符号。

符号	说明
$\mathcal{U}$	短视频用户集合
$\mathcal{V}$	短视频集合
$\mathcal{R}$	用户-视频推荐信息
R_u	用户 u 打算观看的视频集合
$\mathcal{N}$	PCDN节点集合
$\mathcal{S}$	节点-视频存储信息
$\mathcal{T}$	离散时隙集合
S_n	存储在 PCDN节点 n 上的视频集合
c_n	PCDN节点的传输成本 n 每个视频每个时隙
d_n	最大并发传输容量 节点 n 的

且发生在第一个时隙之前。在此阶段，短视频服务器处理推荐信息 $\mathcal{R}$、存储信息 $\mathcal{S}$和并发容量 d_n ，以在全局范围内生成个性化播放列表和节点分配。随后，实际的视频流传输从 $t=1$ 到 T进行。在这些时隙期间，用户以均匀的速率依次下载并观看分配给他们的视频，并按照预先计算的调度方案，在每个时隙结束时切换到下一个视频。

3.3. 问题构建

本节将构建最优视频排序与传输调度（OVOTS）问题，旨在联合确定每个用户的最优播放序列，并为每个视频请求分配一个目标节点来提供服务，从而最小化总交付成本。

为对此问题进行数学建模，我们引入一个二元决策变量 $x_{u,v,n,t} \in \{0,1\}$。具体而言，$x_{u,v,n,t}=1$ 表示中央服务器将视频v 安排在用户 u的播放列表中的第 t位，并指示该用户在时隙 t从节点 n 获取该视频；$x_{u,v,n,t}=0$ 则表示其他情况。决策变量 $x_{u,v,n,t}$ 主要受用户的观看需求以及 PCDN节点的存储容量和并发服务容量约束。

为确保生成的播放列表有效，只有当某个视频明确包含在用户的推荐集合中时，服务器才能将其安排到该用户的播放列表中。这一要求构成了以下边界约束。

$$x_{u,v,n,t} \le \mathbb{I}(v \in R_u), \quad \forall u \in \mathcal{U}, v \in \mathcal{V}, n \in \mathcal{N}, t \in \mathcal{T} \quad (1)$$

其中 $\mathbb{I}(\cdot)$ 是一个指示函数，若条件成立则返回 1 ，否则返回 0 。基于均匀播放速率假设，每位用户每个时隙恰好消费一条短视频。因此，对于任意用户 u 在任意时隙 t，必须从恰好一个指定节点获取一条视频，如式（2）所示。

$$\sum_{v\in\mathcal{V}}\sum_{n\in\mathcal{N}} x_{u,v,n,t} = 1, \quad \forall u \in \mathcal{U}, t \in \mathcal{T} \quad (2)$$

此外，为严格遵循用户的推荐集合，不遗漏也不重复，其中的每条视频 v 都必须在用户 u的推荐集合 R_u 内，并在整个时间范围 $\mathcal{T}$内恰好安排并下载一次。这在数学上表示为式（3）。

$$\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} x_{u,v,n,t} = \mathbb{I}(v \in R_u), \quad \forall u \in \mathcal{U}, v \in \mathcal{V} \quad (3)$$

从视频可用性的角度来看，只有当视频 v 实际缓存在节点 n的本地存储 S_n中时，才能指派该节点为其提供服务，具体可表述如下。

$$x_{u,v,n,t} \le \mathbb{I}(v \in S_n), \quad \forall u \in \mathcal{U}, v \in \mathcal{V}, n \in \mathcal{N}, t \in \mathcal{T} \quad (4)$$

至关重要的是，为严格遵守并发服务限制，任何PCDN节点 n 在任一时隙 t服务的并发用户总数 d_n不得超过其最大并发服务容量。无论请求的是哪些具体视频，这一容量约束均由式（5）严格保证

$$\sum_{u\in\mathcal{U}}\sum_{v\in\mathcal{V}} x_{u,v,n,t} \le d_n, \quad \forall n \in \mathcal{N}, t \in \mathcal{T} \quad (5)$$

给定由矩阵 $\boldsymbol{X}=\{x_{u,v,n,t}\}$ 定义的有效联合排序与调度策略，全局视频传输成本是所有节点在所有时隙内服务请求所产生的传输成本之和。将此总成本记为 $C_{\boldsymbol{X}}=\sum_{u\in\mathcal{U}}\sum_{v\in\mathcal{V}}\sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_n x_{u,v,n,t}$，最终可将OVOTS问题表述为以下整数线性规划（ILP）问题。

$$\begin{aligned} \mathcal{P}: \quad & \min_{\boldsymbol{X}} \quad C_{\boldsymbol{X}} \\ s.t. \quad & x_{u,v,n,t} \in \{0,1\}, \quad \forall u \in \mathcal{U}, v \in \mathcal{V}, n \in \mathcal{N}, t \in \mathcal{T} \\ & \text{Eqs. (1)} - \text{(5)} \end{aligned} \quad (6)$$

问题等价变换：直接求解上述形式化定义的问题 $\mathcal{P}$ 在计算上难以处理，这是由于标准 ILP 求解器 [35, 36]固有的指数复杂度。为便于设计多项式时间解，我们对物理 PCDN节点应用节点分裂变换，将优化问题 $\mathcal{P}$ 转换为一个等价的辅助问题 $\mathcal{P}'$ 。具体而言，将每个具有最大并发容量 d_n、存储配置S_n和单位传输代价 c_n 的物理节点 $n \in \mathcal{N}$ 扩展为一组由 d_n 个相互独立的虚拟节点组成的集合。每个虚拟节点$n' \in \mathcal{N}'$ 均继承其原始物理节点 n相同的存储配置 $S_{n'}=S_n$和传输成本 $c_{n'}=c_n$ ，但其单位并发服务容量严格为 $d_{n'}=1$。令 $\mathcal{N}'$ 表示扩展后的虚拟节点集合，其大小为 $N'=\sum_{n\in\mathcal{N}} d_n$。通过将决策变量重新定义为$x_{u,v,n',t}$（定义在虚拟节点集合 $\mathcal{N}'$上），并将代价函数重新定义为 $C_{\boldsymbol{X}}=$

$\sum_{u\in\mathcal{U}}\sum_{v\in\mathcal{V}}\sum_{n'\in\mathcal{N}'}\sum_{t\in\mathcal{T}} c_{n'} x_{u,\,v,\,n',\,t}$，问题 $\mathcal{P}'$ 可表述如下。

$$
\begin{aligned}
\mathcal{P}'\!:\quad & \min_{\boldsymbol{X}} \quad C_{\boldsymbol{X}} \\
s.t.\quad & x_{u,v,n',t} \in \{0,1\}, \quad \forall u \in \mathcal{U}, v \in \mathcal{V}, n' \in \mathcal{N}', t \in \mathcal{T} \\
& x_{u,v,n',t} \le \mathbb{I}(v \in R_u), \forall u \in \mathcal{U}, v \in \mathcal{V}, n' \in \mathcal{N}', t \in \mathcal{T} \\
& \sum_{v\in\mathcal{V}} \sum_{n'\in\mathcal{N}'} x_{u,v,n',t} = 1, \quad \forall u \in \mathcal{U}, t \in \mathcal{T} \\
& \sum_{t\in\mathcal{T}} \sum_{n'\in\mathcal{N}'} x_{u,v,n',t} = \mathbb{I}(v \in R_u), \quad \forall u \in \mathcal{U}, v \in \mathcal{V} \\
& x_{u,v,n',t} \le \mathbb{I}(v \in S_{n'}), \forall u \in \mathcal{U}, v \in \mathcal{V}, n' \in \mathcal{N}', t \in \mathcal{T} \\
& \sum_{u\in\mathcal{U}} \sum_{v\in\mathcal{V}} x_{u,v,n',t} \le 1, \quad \forall n' \in \mathcal{N}', t \in \mathcal{T}
\end{aligned}
\tag{7}
$$

问题 $\mathcal{P}'$ 在数学上保留了 $\mathcal{P}$的所有可行解及其精确最优目标值。这种结构变换有效地将复杂的多容量调度问题映射为易于处理的二分图分配结构，从而直接支持下一节提出的图论归约。

4. 理论分析与所提算法

在本节中，我们提出一种基于归约的算法框架，以高效求解转换后的优化问题 $\mathcal{P}'$。我们的方法核心是将整数线性规划映射为一个辅助的最小费用最大流（MCMF）问题。我们从理论上证明，该 MCMF 实例的最优目标值与 $\mathcal{P}'$的最优目标值严格等价。基于这一严谨的等价关系，我们提出一种多项式时间两阶段算法——带边着色的最小费用最大流（MMEC），用于重构最优视频排序与传输调度。

4.1. 最小费用最大流问题构造

为解决联合优化问题 $\mathcal{P}'$的时间耦合特性，我们暂时放宽离散时隙约束，并在整个时间范围内汇总系统容量，以构造一个有向流网络$\mathcal{G}=(\mathcal{K},\mathcal{E})$。

如图5所示，顶点集 $\mathcal{K}$ 包含一个源点 S、一个汇点 D、一组需求节点 $\{(u,v) \mid u \in \mathcal{U},\ v \in R_u\}$ 表示每个用户的视频请求，以及扩展后的虚拟节点集 $\mathcal{N}'$。有向边集 $\mathcal{E}$ 根据以下物理映射规则构造:

- **需求生成边：** 对于每个用户 $u \in \mathcal{U}$和每个明确推荐的视频 $v \in R_u$，我们绘制一条边 $S \to (u,v)$，其容量为 1，成本为 0。这严格限制每个推荐视频只能由相应用户下载一次。

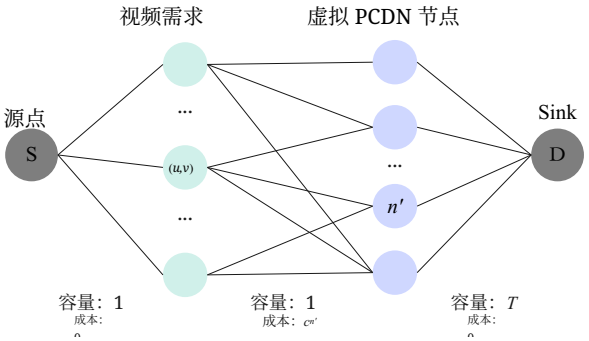


图5： 表示聚合视频交付请求和虚拟节点容量的辅助流网络。

- **视频传输边：** 当且仅当视频 v 物理缓存于虚拟节点 n'（即 $v \in S_{n'}$）时，才存在一条边 $(u,v) \to n'$。每条此类边的容量均设为 1，传输成本为 $c_{n'}$。

- **节点容量边：** 每个虚拟PCDN节点 $n' \in \mathcal{N}'$ 都通过一条边 $n' \to D$连接到汇点 D，该边的容量为 T，成本为 0。这一设计巧妙地强制规定：每个时隙具有严格单位并发容量的虚拟PCDN节点，在整个时间范围 $\mathcal{T}$内最多只能聚合服务 T个视频请求。

令 $\mathcal{F}$表示定义在网络$\mathcal{G}$上的最小费用最大流问题，该问题旨在将所需的确切总流量（即$\sum_{u\in\mathcal{U}}|R_u|$）从源点 S 推送至汇点 D，同时最小化总体传输成本 [37]。

4.2. 等价性分析

现在，我们通过严格证明辅助最小费用最大流问题 $\mathcal{F}$ 与转换后的调度问题$\mathcal{P}'$之间的等价性，来确立该归约的最优性。

定理1. 令 $OPT_{\mathcal{P}'}$ 为调度问题 $\mathcal{P}'$的最小总交付成本，令 $OPT_{\mathcal{F}}$ 为所构造的网络流问题 $\mathcal{F}$的最小成本。则 $OPT_{\mathcal{P}'} = OPT_{\mathcal{F}}$。

证明. 证明依赖于建立双向界关系。我们首先证明 $OPT_{\mathcal{F}} \le OPT_{\mathcal{P}'}$，方法是说明问题 $\mathcal{P}'$的任意最优解都对应于问题 $\mathcal{F}$ 中的一个成本相同的可行流。随后，我们证明 $OPT_{\mathcal{P}'} \le OPT_{\mathcal{F}}$，方法是说明 $\mathcal{F}$ 的任意最优流都可以无损地分解为 $\mathcal{P}'$的一个可行调度矩阵。

(1) $OPT_{\mathcal{F}} \le OPT_{\mathcal{P}'}$。

解的构造： 给定问题 $\mathcal{P}'$ 的最优调度矩阵 $\boldsymbol{X}^* = \{x^*_{u,v,n',t} \mid u \in \mathcal{U}, v \in \mathcal{V}, n' \in \mathcal{N}', t \in \mathcal{T}\}$，我们在网络 $\mathcal{G}$ 上构造一组流 F。对于每个 $x^*_{u,\,v,\,n',\,t} = 1$，我们沿路径 $S \to (u,v) \to n' \to D$ 路由一个单位的流。因此，映射到

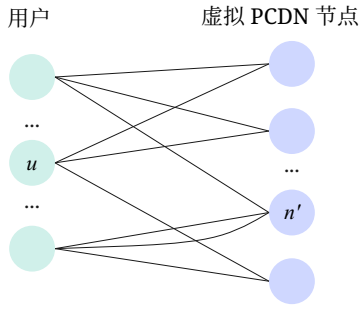


图 6： 由最优流 F^*构造的二分多重图 $\mathcal{G}_{\mathcal{B}}$。平行边表示用户与虚拟 PCDN 节点之间的多个不同视频请求。

网络上的总流量为

$F = \{f(S \to (u,v) \to n' \to D) = 1 \mid x^*_{u,v,n',t} = 1\}$。

约束满足与成本匹配： 首先，F中的流不会违反需求生成边 $S \to (u,v)$的单位容量，因为 $\mathcal{P}'$的约束确保每个用户在 $\mathcal{T}$中恰好获取一次每个推荐视频。其次，流不会违反视频传输边 $(u,v) \to n'$的单位容量，因为每个特定的视频请求都恰好由一个指定的虚拟节点提供服务。第三，流不会违反节点容量边 $n' \to D$的聚合容量 T，因为 $\mathcal{P}'$ 中的容量约束规定，每个虚拟节点 n' 在每个时隙最多为一个用户提供服务，在 T 个时隙内累计最多处理 T 个请求。由于流映射与决策变量严格一一对应，因此总流成本恰好等于调度成本，即 $Cost(F) = C_{X^*} = OPT_{\mathcal{P}'}$。由于 F是 $\mathcal{F}$的有效可行解，$\mathcal{F}$ 的最优成本不会高于 F的成本。因此，$OPT_{\mathcal{F}} \le OPT_{\mathcal{P}'}$。

(2) $OPT_{\mathcal{P}'} \le OPT_{\mathcal{F}}$。

解的构造： 给定问题 $F^* = \{f^*(S \to (u,v) \to n' \to D) = 1\}$的最优整数流解 $\mathcal{F}$，我们构造一个二分多重图 $\mathcal{G}_{\mathcal{B}} = (\mathcal{U}, \mathcal{N}', \mathcal{E}_B)$，如图6所示，其中 $\mathcal{U}$表示用户，$\mathcal{N}'$ 表示虚拟PCDN节点。对于 F^*中的每个单位流 $f^*(S \to (u,v) \to n' \to D) = 1$，我们添加一条连接 $u \in \mathcal{U}$ 和 $n' \in \mathcal{N}'$的边，并以视频 v 为其唯一标记，例如 $e_{u、v、n'}$。

约束满足与成本匹配： 在二分多重图 $\mathcal{G}_{\mathcal{B}}$中，每个顶点 $u \in \mathcal{U}$ 的度恰好为 T，因为每个用户都需要下载并观看T 个视频。此外，由于边$n' \to D$的流容量有限，每个顶点 $n' \in \mathcal{N}'$的度不能超过 T。因此，$\mathcal{G}_{\mathcal{B}}$ 的最大度不超过$\Delta(\mathcal{G}_{\mathcal{B}}) \le T$。根据 *Kőnig* 边染色定理，任何最大度为 Δ 的二分多重图都是 Δ-可边染色的。这从理论上保证了我们可以使用恰好 T 种颜色为 $\mathcal{G}_{\mathcal{B}}$ 中的所有边着色，使得任意两条相邻边的颜色都不同。

通过将 T 颜色与 $\mathcal{T}$ 中的 T 离散时隙建立双射，我们设置决策变量 $x_{u,v,n',t} = 1$，若边 $e_{u,v,n'}$ 被着色为颜色 t，从而得到一个完整的调度矩阵 $\boldsymbol{X}$。Kőnig 定理严格保证每个时隙中每个用户获取的视频不超过一个，且每个虚拟PCDN节点为每个时隙服务的用户不超过一个，从而满足 $\mathcal{P}'$ 的所有约束。由于着色过程中物理路由路径保持不变，传输成本得以完全保留（$C_{\boldsymbol{X}} = Cost(F^*) = OPT_{\mathcal{F}}$）。因此，$\boldsymbol{X}$ 是 $\mathcal{P}'$ 的一个可行解，这意味着$OPT_{\mathcal{P}'} \le OPT_{\mathcal{F}}$。

结合式 (1) 和式 (2) 中的界，我们最终得出结论：$OPT_{\mathcal{P}'} = OPT_{\mathcal{F}}$。 □

4.3. 算法描述

基于前面各小节确立的理论基础，我们提出带边着色的最小费用最大流（MMEC）算法。如算法1所示，该框架将联合优化问题高效地解耦为两个严格按顺序执行的阶段：基于流的传输调度阶段和基于染色的视频排序阶段。

第一阶段（第 3–13 行）旨在确定将视频请求分配给虚拟 PCDN 节点的最优方案，以使总传输成本最小化。首先，MMEC 构建辅助网络 $\mathcal{G}$，并初始化残量图 $\mathcal{G}_r$ 以及空流集合 F。随后，MMEC 在 $\mathcal{G}_r$ 上迭代寻找成本最低的路径，直至不存在其他有效路径。每次迭代中，算法使用 Bellman-Ford 算法寻找一条从源点 S 到汇点 D 的成本最低路径。接着，将沿该最小成本路径的单位流（记为 $f(S \to (u,v) \to n' \to D) = 1$）加入集合 F。由此可在数学上确定最优的用户—节点配对，即虚拟节点 n' 被永久分配用于满足用户 u对视频 v的请求。随后，通过递减相应边的容量，更新残量图 $\mathcal{G}_r$。最后，MMEC 输出流集合 F，作为最小费用最大流问题的最优解。

尽管阶段 1 最优地解决了传输调度问题（即由哪些虚拟 PCDN 节点服务哪些视频需求），但并未指定最终的播放顺序。阶段 2（第 14–20 行）通过将最优流 F 转换为可行且无冲突的时隙调度，解决了这一时间映射问题。MMEC 首先构建一个二分多重图 $\mathcal{G}_{\mathcal{B}} = (\mathcal{U}, \mathcal{N}', \mathcal{E}')$，其中顶点集$\mathcal{U}$ 和 $\mathcal{N}'$ 分别表示用户和虚拟 PCDN 节点。对于 F中的每个单位流，若其将视频 v分配给虚拟节点 n' 以服务用户 u，便会向 $\mathcal{E}'$中添加一条唯一的边 $e_{u,v,n'}$。随后，对于 $\mathcal{G}_{\mathcal{B}}$中的每条边，MMEC 使用 Kempe 链边着色算法识别并分配一个可用颜色 $t \in \mathcal{T}$，确保任意相邻边的颜色均不相同。最后，MMEC 输出调度矩阵$\boldsymbol{X}$，其中 $x_{u,v,n',t} = 1$ 直接表明，视频 v 将在时隙 t内由用户 u从虚拟节点 n' 获取。

最后，为原始问题生成解$\mathcal{P}$，MMEC 在第 21 至 25 行执行映射步骤。它通过将每个虚拟节点 n' 还原至其

物理父节点 n，自然地将虚拟调度矩阵 n 映射回物理域。因此，它输出最终的联合调度矩阵 $\boldsymbol{X} = \{x_{u,v,n,t}\}$，直接为原始问题 $\mathcal{P}$ 提供最优解。

算法 1 带边着色的最小费用最大流（MMEC） 输入：用户-视频推荐集合 $\{R_u\}$、物理节点存储配置 $\{S_n\}$、成本 c_n以及容量 d_n输出：问题 $\mathcal{P}$ 的最优联合调度矩阵 $\boldsymbol{X}$ 1：将物理节点 $\mathcal{N}$ 扩展为虚拟节点 $\mathcal{N}'$，其中 $N' = \sum_{n\in\mathcal{N}} d_n$ 2：初始化调度矩阵 $\boldsymbol{X}' \leftarrow \emptyset$ 3：阶段 1：基于流的传输调度（求解最小费用最大流） 4：按照第 4.1 节的定义，基于虚拟节点 $\mathcal{N}'$构建网络图 $\mathcal{G}$ 5：初始化流集合 $F \leftarrow \emptyset$ 6：初始化残量图 $\mathcal{G}_r \leftarrow \mathcal{G}$ 7：当 True 时执行 8：使用 Bellman-Ford算法在$\mathcal{G}_r$ 中寻找最小成本路径 $S \to (u,v) \to n' \to D$ 9：若不存在路径，则跳出 10：结束 if 11： $F \leftarrow F \cup \{f(S \to (u,v) \to n' \to D) = 1\}$ **12：更新残量图 $\mathcal{G}_r$中的容量 13：结束 while 14：阶段 2：基于着色的视频排序（求解边着色） 15：基于流集合 F构建二分多重图 $\mathcal{G}_{\mathcal{B}} = (\mathcal{U}, \mathcal{N}', \mathcal{E}')$ 16：初始化表示时隙的颜色集合 $\mathcal{T}$ $= \{1, 2, \ldots, T\}$ 17：对 $\mathcal{G}_{\mathcal{B}}$ 中的每条边 e_u、v、n' 执行 18：使用 Kempe 链着色算法分配颜色 $t \in \mathcal{T}$ 19：$\boldsymbol{X}' \leftarrow \boldsymbol{X}' \cup \{x_{u,v,n',t} = 1\}$ 20：结束 for 21：初始化最终物理调度矩阵 $\boldsymbol{X} \leftarrow \emptyset$ 22：对每个 x_u、v、n'、$_t = 1 \in \boldsymbol{X}'$ 执行 23：将虚拟节点 n' 映射回其物理节点 n 24：$\boldsymbol{X} \leftarrow \boldsymbol{X} \cup \{x_{u,v,n,t} = 1\}$ 25：结束 for 26：返回 $\boldsymbol{X}$**

4.4. 复杂度分析

为所提出的 MMEC 算法提供全面的理论基础，我们对其计算效率进行了严格评估。整体复杂度由两个依次执行的阶段决定。

阶段 1 的计算复杂度取决于辅助流网络 $\mathcal{G} = (\mathcal{K}, \mathcal{E})$ 的规模。具体而言，顶点集大小上限为 $|\mathcal{K}| = UT + N' + 2 = O(UT + N')$，由视频需求节点、虚拟存储节点、源点和汇点组成。边集大小上限为需求生成边、视频传输边和节点容量边的总和。由于传输边的数量 $|\mathcal{E}_{trans}|$ 严格受需求节点数与虚拟节点数乘积的限制（$|\mathcal{E}_{trans}| \le UTN'$），因此边总数为 $|\mathcal{E}| = O(UTN')$。利用 Bellman-Ford 算法的逐次最短路径过程恰好执行 UT 次（所需流量的每个单位执行一次）。由于单次 Bellman-Ford 算法执行需要 $O(|\mathcal{K}| \cdot |\mathcal{E}|) = O(U^2T^2N' + UTN'^2)$ 时间，阶段 1 的累计复杂度达到 $O(U^3T^3N' + U^2T^2N'^2)$。阶段 2 涉及为构建的二分多重图 $\mathcal{G}_B = (\mathcal{U}, \mathcal{N}', \mathcal{E}')$ 中的边着色。该图中的边数恰为 $|\mathcal{E}'| = UT$。在最坏情况下，Kempe 链着色算法通过搜索交替路径，在 $O(U + N')$ 时间内解决单条边冲突。对全部 UT 条边执行此操作，得到阶段 2 的总复杂度为 $O(U^2T + UTN')$。

综合所有操作阶段，MMEC 算法的整体计算复杂度主要由阶段 1（求解最小费用最大流问题）决定。因此，总时间复杂度上界为 $O(U^3T^3N' + U^2T^2N'^2)$。

5. 性能评估

本节通过一系列数值仿真，对所提出的 MMEC 算法进行全面评估。我们首先定义实验环境、数据生成模型和参数设置。随后，我们介绍基线算法，并分析在不同系统参数设置下，联合视频排序与传输调度框架所带来的性能提升。

5.1. 实验设置

问题实例生成。为确保核心性能评估清晰且易于处理，我们构建了一个同质化的 PCDN，其中所有对等节点都具有均匀的并发服务容量（记为d）、相同的存储限制（记为 s）以及相同的单位传输成本[3]。具体而言，对于所有$n \in \{1, \ldots, N-1\}$，任一普通对等节点的单位传输成本均标准化为 $c_n = 1$，而从中心 CDN 节点获取内容的单位成本则设为 $c_N = 5$，以反映骨干网带宽消耗所带来的高昂成本。

在此 PCDN 中，整体交付成本很大程度上取决于视频缓存与用户需求模式之间错综复杂的相互作用。为生成真实的用户视频请求，我们假设整个视频库的流行度服从齐夫分布，其偏斜参数为 α，这严格反映了现实短视频平台中观察到的高度集中的观看行为。基于该分布，我们为每位用户 u 独立抽取恰好T 个视频，以构建其个性化观看需求集 R_u。

同时，我们采用按流行度排序的循环放置策略，生成缓存于

[3]附录中详细介绍了补充实验，充分展示了我们的框架在异构对等节点架构下的稳健适用性A。

表3控制变量实验的参数设置

参数名称	默认值	范围	Step
用户数量（U）	100	50–140	10
对等节点数量（N）	50	10–100	10
视频数量（V）	300	100–500	50
时隙数量（T）	10	6–20	2
每个存储容量的（s）	6	2–10	1
并发服务容量每个对等节点（d）	2	1–5	1
流行度偏斜（α）	0.6	0.1–1.0	0.1

各个对等节点中的视频集合， S。具体而言，我们按照流行度降序排列视频库，并以循环方式依次将这些视频分配给各个对等节点。这种结构化初始化实现了两个关键目标：首先，确保每个视频都明确地缓存在至少一个低成本对等节点上；其次，将高流行度视频分散到各个对等节点，从根本上避免任何单个对等节点出现并发容量瓶颈。[4]

方法与参数设置。 为系统地研究总体传输成本对不同系统规模、资源容量和用户需求模式的敏感性，我们采用控制变量法。如表3所示，我们依次在指定范围内改变单个目标参数，同时严格保持所有其他系统变量为其默认值。

5.2. 对比算法

为了全面评估我们的联合优化框架及所提出的MMEC算法的有效性，我们将其性能与三种具有代表性的基线算法进行比较。这些算法经过审慎选择，旨在体现不同程度的系统优化，既作为性能基准，也用于消融研究。

随机视频排序与随机传输调度（RORS）。这是一种朴素且完全未经优化的基线算法。在RORS中，服务器根据每位用户的推荐集合 R_u，为其生成完全随机的播放序列。对于每个所需视频，服务器会从当前缓存该内容且具有可用并发容量的对等节点子集中，随机指定一个服务节点。

随机视频排序与最优传输调度（ROOS）。该算法是关键的消融研究基线算法。它剥离了视频排序优化，仅保留传输调度优化。在 ROOS 中，播放序列仍为随机生成；但服务器会基于这一固定序列计算最优传输调度，以最小化整体传输成本。将 MMEC 与 ROOS 直接比较，可以量化由智能调整用户播放序列所带来的巨大性能通过智能调整用户播放序列而获得的收益。

模拟退火优化（SAO）。这一基线算法代表一种经典的元启发式方法，直接用于求解原始联合整数线性规划（ILP）模型。该方法从一个贪心初始解开始，其中播放序列经过随机化，并将每个视频请求分配给可用节点中成本最低的节点。随后，算法通过随机交换视频时隙并重新分配服务节点，迭代探索解空间，以跳出局部最优。SAO配置经过严格调优，采用初始温度 T_{init} =1000、冷却因子 $\gamma = 0.95$、最多 $I = 1000$ 次迭代，以及每个温度步长进行 $L = 20$次内部扰动。

5.3. 数值结果

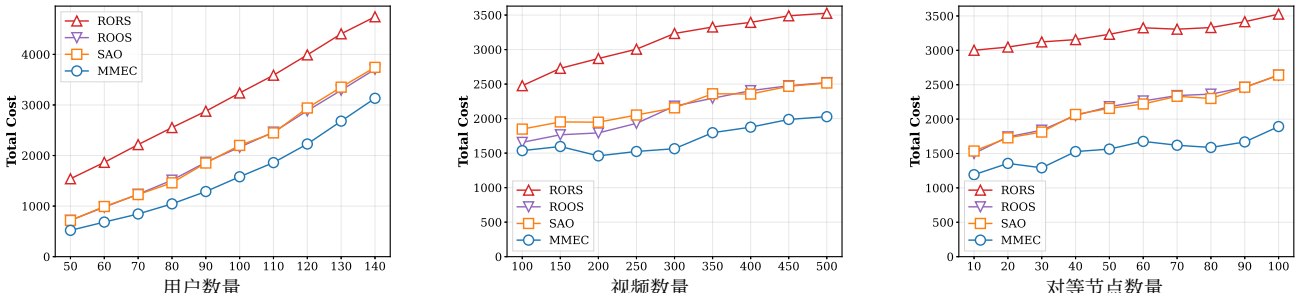
5.3.1. 用户数量的影响

图7a展示了用户数量 U 从50增至140时的整体传输成本。正如预期，随着视频需求增加，所有受评估算法的成本均单调上升，但随着系统负载加重，性能差距显著扩大。朴素的RORS基线算法由于随机播放序列导致对等节点发生严重并发拥塞，成本增幅最为显著。值得注意的是，图中ROOS消融基线与SAO元启发式算法的性能几乎完全重合。这表明，联合视频排序与传输调度问题具有高度耦合的特性，导致解空间崎岖，使传统启发式搜索（SAO）陷入局部最优。相比之下，所提出的MMEC算法有效突破了这一算法瓶颈，表明智能视频排序与传输调度同样关键。通过在数学上解耦这两个阶段，MMEC能够智能地平滑并发需求峰值，从而最大限度地提高对等节点的利用率。因此，MMEC相较于RORS的成本最高可降低67%（在 $U = 50$），相较于SAO/ROOS最高可降低36%（在 $U = 70$）。尽管 $U = 140$ 对等节点严重耗尽不可避免地迫使系统依赖成本高昂的中心CDN，并缩小了这些相对优势，但MMEC仍分别实现了35%和17%的显著降幅。

5.3.2. 视频数量的影响

图 7b 展示了随着视频库大小 V 从 100 增至 500 时的总体传输成本。在整个范围内，MMEC 始终优于所有基线算法。对于所有评估方法，成本最终都会上升，因为在对等节点存储容量固定的情况下，目录变大必然会降低缓存命中率，从而不得不依赖成本高昂的 CDN节点。然而，MMEC 在 $V = 100$ 到 $V = 300$之间呈现出明显的“成本谷值”，绝对成本维持在约 750。这一现象的原因是，过小的 V 会使用户请求过度集中，导致特定对等节点出现严重的并发容量瓶颈。MMEC 实现了峰值

[4]附录中还评估了其他对等节点存储初始化策略B，以进一步验证系统的稳健性。



（a）总成本与用户数量的关系 U。 （b）总成本与视频库大小的关系 V。 （c）总成本与对等节点数量的关系 N.

（d）总成本与时隙数量的关系 T。 （e）总成本与并发服务能力的关系-每个对等节点的存储容量。 （f）总成本与每个对等节点的存储容量的关系。

图 7： 在不同系统参数下，所提出的 MMEC算法与基准算法（RORS、ROOS 和 SAO）在总传输成本方面的性能比较。

成本降幅在 $V = 300$处达到最高，较 RORS 约降低 55%，较 SAO/ROOS 集群约降低 31%。

5.3.3. 对等节点数量的影响

图 7c展示了随着对等节点数量 N 从 10 增至 100 时的总传输成本。为严谨评估算法应对资源碎片化的鲁棒性，我们假设本实验中系统总存储容量和并发服务容量保持不变。因此，增加 N 会将可用资源拆分为更小、能力更弱的单个节点，从而限制每个视频的服务能力，并使所有受评估算法的绝对成本均上升。值得注意的是，随着 N 增加，RORS、SAO 和 ROOS 的成本急剧上升，这表明当解空间高度碎片化时，它们难以应对。相比之下，MMEC 凭借数学上有保证的全局优化，有效处理了这种碎片化，使成本变化保持最稳定。因此，MMEC 相较于 RORS 的最高相对成本降幅约为 59%（当 $N = 10$时），相较于 SAO/ROOS 的降幅最高可达 32%（当 $N = 70$时）。

5.3.4. 时隙数量的影响

图 7d展示了总体传输成本随时隙数量 T（表示每位用户的视频请求总数）从6扩展到20时的变化。所有评估算法的成本均线性增长，这反映了全系统需求的基本扩展规律，因为更大的 T 会直接提高总传输量。朴素的RORS基线算法呈现出最快的成本上升。SAO与ROOS基线方法再次表现出近乎相同的次优性能。相比之下，无论时间维度如何扩展，MMEC都能在数学上保证全局最优序列。因此，在 $T = 14$时，MMEC相对于RORS的峰值相对成本降幅约为54%，相对于SAO和ROOS基线方法的降幅均为31%。

5.3.5. 每个对等节点的并发服务容量的影响

图7e展示了当每个对等节点的并发服务容量从1扩展至5时的总体传输成本。正如预期，随着容量扩大，所有被评估算法的成本均下降，因为更高的并发服务容量使对等节点能够承接更多并发请求。朴素的RORS基线算法始终最为昂贵，且降幅相对平缓。SAO与ROOS基线方法再次呈现出相同的次优轨迹。相比之下，MMEC智能地编排播放序列，以充分利用可用容量，保持最低的绝对成本，并在容量超过3时形成明显的平台期。因此，MMEC在 $d = 5$处相对RORS实现约63%的最大相对成本降幅，并在 $d = 2$处相对SAO/ROOS最高达到26%。

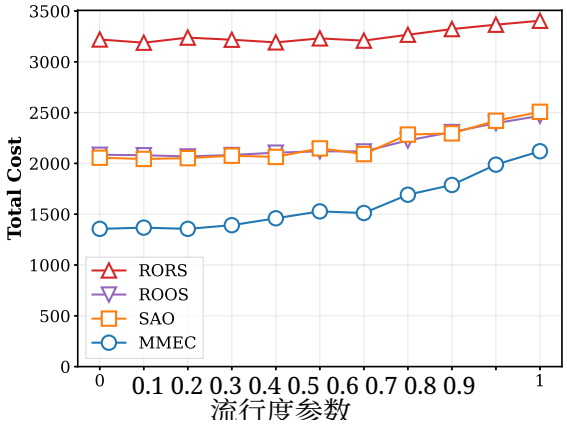


图 8： 不同算法下，Zipf流行度偏斜参数 α 对总视频传输成本的影响。

5.3.6. 每个对等节点的存储容量的影响

图7f展示了当每个对等节点的存储容量从2扩展至10时的总体传输成本。正如预期，所有被评估算法的成本均呈下降趋势，因为存储容量的扩大使对等节点能够缓存更多视频，从而直接降低对昂贵CDN节点的依赖。朴素的RORS基线算法始终产生最高成本。有趣的是，尽管SAO与ROOS基线方法起初相互重合，但当存储容量超过6时，ROOS略优于SAO，这表明启发式SAO难以在扩大的解空间中有效寻优。相比之下，MMEC高效利用扩大后的存储容量，推动成本迅速下降，并在容量超过8时最终形成明显的平台期。这一趋稳表明，主要性能瓶颈现已从对等节点的存储容量转移到其并发服务容量。因此，MMEC的最大相对成本降幅在容量为8时达到峰值，相对RORS约为53%，相对SAO基线算法约为31%；在容量为6时相对ROOS基线算法约为28%。

5.3.7. 流行度偏斜的影响

图8 展示了 Zipf流行度偏斜参数 α 从 0.0 变化到 1.0 时的总体传输成本。随着 α 增大，用户需求会高度集中于一部分超热门视频。这会导致存储这些视频的特定对等节点的并发服务容量遭到严重争用，超额请求因此不可避免地被转发到成本高昂的 CDN节点。朴素的 RORS 基线算法始终产生最高成本，而 MMEC 则展现出最强的优势。具体而言，MMEC 在 $\alpha = 0.2$时实现了最大的相对成本降幅：相较于 RORS 约降低 59%，相较于 SAO 和 ROOS 基线方法均约降低 36%。

6. 结论

本文针对短视频平台降低传输成本这一挑战，提出了一个视频排序与传输调度联合优化框架。不同于将用户请求视为刚性约束的传统响应式缓存机制，我们的方法利用服务器驱动的推荐播放列表固有的灵活性，主动塑造网络需求。

通过抽象短视频传输架构的核心组件，我们对短视频流媒体系统进行了建模，并提出 OVOTS问题，以最大限度地降低总体传输成本。我们证明了该问题可精确归约为一个辅助最小费用最大流问题，并基于 Kőnig 边染色定理建立了数学上最优的映射，从而提供了严谨的理论基础。基于这些见解，我们开发了MMEC算法，这是一种计算效率高的多项式时间解，可将传输调度与视频排序解耦。

大量实验评估验证了我们的框架能够有效应对突发的需求激增，并最大限度地提高存储资源严重受限的对等节点的利用率。MMEC算法在不同的系统规模、并发服务能力和存储限制下，均表现出卓越的性能和稳健性。具体而言，与标准基线方法相比，我们的方案实现了超过 67% 的峰值成本降幅，这表明播放序列置换是降低传输成本的一种极为有效、但此前尚未得到充分利用的范式。最后，未来研究的潜在方向包括将这一联合优化框架扩展至高度动态的环境，以应对用户留存率不可预测以及多路径传输等特征。

A. 异构对等节点

在主要的性能评估（第5节）中，为确保数学上的清晰性，我们假设处于同构PCDN环境。然而，在实际部署中，从第三方提供商租用的对等节点自然会表现出显著的异质性。在本附录中，我们扩展了评估范围，以研究对等节点具备异构资源能力时系统的性能和鲁棒性。

A.1. 异构存储容量的影响

为了评估我们的联合优化框架在真实异构对等节点中的稳健性，我们在两种不同的存储分布下比较算法性能：存储容量相同（均匀）以及存储容量随机变化（随机），同时严格保持网络存储容量总量相等。我们进行了 20 次独立仿真实验，以减少随机波动的影响。如图 9所示，结果表明，尽管随机 存储分布会自然地增加所有算法的总体传输成本，但所提出的 MMEC 始终保持着优异的性能。

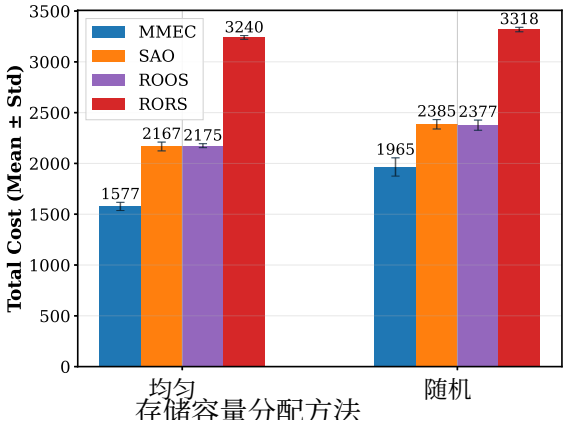


图9： 均匀和随机存储容量分配下的性能比较。

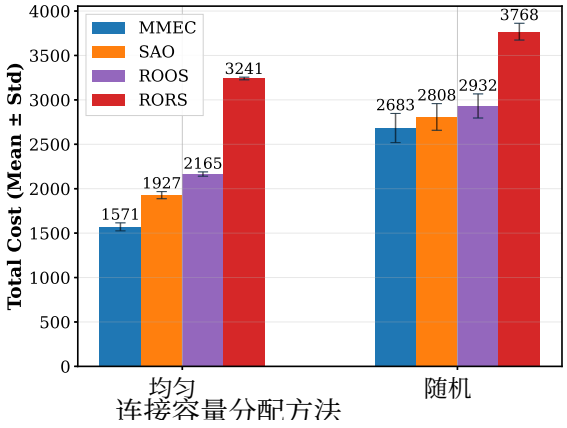


图10： 均匀和随机并发服务容量分配下的性能比较。

MMEC 有效消除了异构存储分布带来的负面影响，展现出强大的适应性，并能稳健应对实际边缘部署中固有的硬件差异。

A.2. 异质并发服务容量的影响

同样，为评估并发服务容量异质性的影响，我们在均匀和随机并发服务容量分配下评估系统性能，同时严格保持对等节点的总容量一致。如图 10 所示，结果表明，在随机设置下，容量异质性会显著提高所有算法的总体传输成本。这种性能下降是因为缓存热门内容的节点可能会随机受到极低并发限制的约束，立即造成瓶颈，迫使流量转向成本高昂的内容分发网络。尽管如此，所提出的 MMEC算法始终优于基线方法，它通过智能地调整播放序列，使用户请求在时间域内错峰到达。这种主动的时间错峰有效规避了不可预测的容量瓶颈，展现了 MMEC算法在高度异质的带宽环境中的出色鲁棒性。

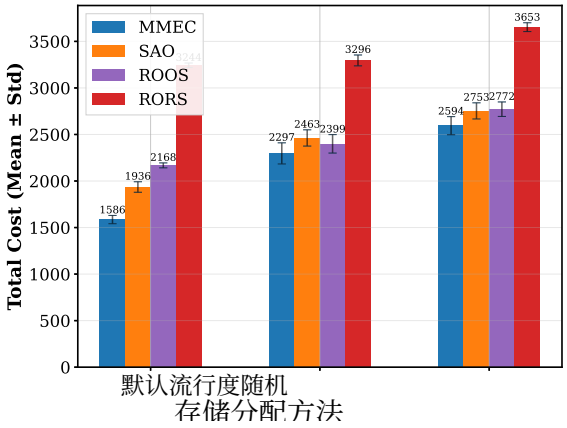


图11： 不同存储初始化策略下的性能比较。

B. 对等节点存储初始化策略

最后，我们研究了不同对等节点存储初始化策略的影响，即将我们的默认值按流行度排序的循环放置与两种替代方法进行比较：流行度（其中每个对等节点独立地根据齐夫分布采样视频）和随机（其中每个节点均匀随机地采样视频）。如图11所示，结果表明这两种替代策略均显著提高了总体传输成本。这一成本代价的产生，是因为流行度策略会导致少数超热门视频出现严重的冗余缓存，而随机策略则完全忽略了偏斜的用户需求，从而导致大量缓存未命中以及被迫回退到内容分发网络。尽管如此，即使在这些高度次优的缓存设置下，所提出的MMEC算法与RORS、ROOS和SAO基线方法相比，仍始终保持最低的总体传输成本。

致谢

本研究得到国家自然科学基金重点项目（项目编号62531003）资助。

CRediT作者贡献声明

高志鹏： 调查、方法论、数据分析、模拟、撰写——初稿。**李春曦：** 调查、方法论、数据分析、撰写——初稿。**赵永祥：** 调查、方法论、数据分析、撰写——审阅与编辑。

参考文献

[1] Huang, X., Wu, W., Hu, S., Li, M., Zhou, C., Shen, X., 2023. 面向组播的基于数字孪生的以用户为中心的资源管理

短视频流媒体。IEEE《信号处理精选主题期刊》18, 50–65。[2] Shang, Y., Gao, C., Li, N., Li, Y., 2025. 包含移动短视频平台行为、属性与内容的大规模数据集，载于：ACM Web会议论文集，第793–796页。[3] Ghabashneh, E., Rao, S., 2020. 探究CDN缓存与视频流媒体性能之间的相互作用，载于：IEEE INFOCOM会议论文集，第516–525页。[4] Zhang, G., Liu, W., Hei, X., Cheng, W., 2014. 揭开迅雷看看：理解混合CDN-P2P点播流媒体。IEEE《多媒体汇刊》17, 229–242。[5] Wei, D., Zhang, J., Li, H., Xue, Z., Peng, Y., Han, R., 2023. 短视频点对点CDN传输架构中的多路径智能预加载算法。IEEE《网络》38, 285–291。[6] Wei, D., Zhang, J., Liu, X., Li, H., Xue, Z., Huang, T., Jiang, L., Li, J., 2025. 大规模点对点CDN中面向视频服务的QoE优化的多路径调度。IEEE《通信精选领域期刊》43, 563–576。[7] Wei, D., Zhang, J., Li, H., Xue, Z., Peng, Y., Pang, X., Liu, Y., Han, R., Li, J., 2024. PScheduler：面向大规模点对点CDN中视频服务的QoE增强型多路径调度器，载于：IEEE INFOCOM会议论文集，第2508–2517页。[8] Farahani, R., Çetinkaya, E., Timmerer, C., Shojafar, M., Ghanbari, M., Hellwagner, H., 2023. ALIVE：一种面向直播视频流的时延与成本感知混合P2P-CDN框架。IEEE《网络与服务管理汇刊》21, 1561–1580。[9] Jung, J., Krishnamurthy, B., Rabinovich, M., 2002. 突发流量与拒绝服务攻击：特征及其对内容分发网络和网站的影响，载于：ACM WWW会议论文集，第293–304页。[10] Zhang, R.X., Wang, H., Shi, S., Pang, X., Peng, Y., Xue, Z., Liu, J., 2024. 增强全球最大PCDN系统的点播视频流媒体资源管理，载于：USENIX ATC会议论文集，第951–965页。[11] Wang, H., Zhang, R., Li, C., Xue, Z., Peng, Y., Pang, X., Zhang, Y., Ren, S., Shi, S., 2024. T wist：面向点播视频流媒体的多站点传输方案。ACM《网络学报》2, 1–19。[12] Liu, F., Shen, S., Li, B., Li, B., Yin, H., Li, S., 2011. Novasky：P2P存储云中的影院级画质视频点播，载于：IEEE INFOCOM会议论文集，第936–944页。[13] Huang, X., Hu, S., Yang, H., Wang, X., Pei, Y., Shen, X., 2024. 面向组播短视频流媒体中更优QoE的基于数字孪生的网络管理。IEEE《无线通信汇刊》23, 16187–16202。[14] Gao, Z., Li, C., Zhao, Y., Zhang, B., Li, C., 2022. 通过视频组播最小化无线资源的短视频列表重排，载于：IEEE IWCMC会议论文集，第943–948页。[15] Viola, R., Martin, A., Morgade, J., Masneri, S., Zorrilla, M., Angueira, P., Montalbán, J., 2020. 基于LSTM网络性能预测与成本效益权衡的视频传输预测式CDN选择。IEEE《广播汇刊》67, 145–158。[16] Peroni, L., Gorinsky, S., 2025. 尽力而为网络中视频流传输的端到端管线视角：综述与教程。ACM《计算调查》57, 1–47。[17] Tan, Z., Hu, W., Zhang, Y., Ding, H., 2019. 视频片段的在线流行度预测：迈向更高效的内容分发网络，载于：IEEE GLOBECOM 会议论文集，第1–6页。[18] Gao, Z., Li, C., Zhao, Y., Zhang, B., Li, C., 2025. 启动延迟感知的短视频排序：问题、模型与基于强化学习的算法。《点对点网络与应用》18, 74。[19] Quan, Y., Ding, J., Gao, C., Li, N., Yi, L., Jin, D., Li, Y., 2023. 缓解微视频推荐中的视频时长效应。ACM《信息系统汇刊》42, 1–24。[20] Simonovski, F., Hufen, S., Karl, L., Sayin, A., Wehner, N., Hoßfeld, T., Seufert, M., 2025. 快慢滑动：通过众包评估短视频的QoE，载于：IEEE QoMEX会议论文集，第1–7页。

[21] Gong, X., Feng, Q., Zhang, Y., Qin, J., Ding, W., Li, B., Jiang, P., Gai, K., 2022. 移动设备上的实时短视频推荐，载于：ACM CIKM 会议论文集，第 3103–3112 页。[22] Chen, L., Kyng, R., Liu, Y., Peng, R., Probst Gutenberg, M., Sachdeva, S., 2025. 近线性时间内的最大流与最小费用流。ACM 会刊 72，1–103。[23] Schrijver, A., 1998. 二分图边着色，时间复杂度为 o(δm)。SIAM 计算杂志 28，841–846。[24] Parimala, M., Broumi, S., Prakash, K., Topal, S., 2021. 图像模糊环境下求解网络最短路径问题的 Bellman–Ford 算法。复杂与智能系统 7，2373–2381。[25] Robertson, N., Sanders, D., Seymour, P., Thomas, R., 1997. 四色定理。组合理论杂志 B 辑 70，2–44。[26] Farahani, R., Bentaleb, A., Çetinkaya, E., Timmerer, C., Zimmer- mann, R., Hellwagner, H., 2022. 直播视频流传输的混合 P2P-CDN 架构：一种在线学习方法，载于：IEEE GLOBECOM 会议论文集，第 1911–1917 页。[27] Jiang, H., Li, J., Li, Z., Bai, X., 2009. 结合内容分发网络与点对点网络实现高效的大规模内容分发。国际混合信息技术杂志 2，4。[28] Nygren, E., Sitaraman, R.K., Sun, J., 2010. Akamai 网络：高性能互联网应用的平台。ACM SIGOPS 操作系统评论 44，2–19。[29] YouTube, 2026. YouTube。https://www.youtube.com/。访问日期：2026-02-24。[30] Netflix, 2026. 无限量电影、电视节目等。https://www.netflix.com/。访问日期：2026-02-24。[31] Sun, Z., Si, Z., Zang, X., Leng, D., Niu, Y., Song, Y., Zhang, X., Xu, J., 2023. KuaiSAR：统一搜索与推荐数据集，载于：ACM CIKM 会议论文集，第 5407–5411 页。[32] 快手, 2026. 快手是一个分享短视频和潮流趋势的社交网络。https://www.kwai.com/。访问日期：2026-02-24。[33] PPTV, 2026. PPTV。https://www.pptv.com/。访问日期：2026-02-24。[34] Bag, S., Kumar, S.K., Tiwari, M.K., 2019. 利用相关杰卡德相似度高效生成推荐结果。信息科学 483，53–64。[35] Gurobi Optimization, L., 2026. 决策智能技术领域的领导者。https://www.gurobi.com/。访问日期：2026-02-24。[36] Google, 2026. OR-Tools | Google 开发者。https://developers. google.com/optimization。访问日期：2026-02-24。[37] Ford, L.R., Fulkerson, D.R., 1962.《网络中的流》。普林斯顿大学出版社，普林斯顿，新泽西州。