IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 8, 2020, accepted April 20, 2020, date of publication April 28, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990900

A Heuristic Transferring Strategy for Heterogeneous-Cached ICN

YULAN ZHAO[1] AND JIANHUI LV[2], (Member, IEEE)
[1]School of Electrical and Information Engineering, Jilin Agricultural Science and Technology University, Jilin 132101, China
[2]International Graduate School at Shenzhen, Tsinghua University, Shenzhen 518057, China

Corresponding author: Jianhui Lv (lvjianhui2012@163.com)

ABSTRACT The in-network caching is a considerably significant feature of Information-Centric Networking (ICN), especially the heterogeneous-cached ICN has been widely investigated since it accords with the actual network deployment. For the heterogeneous-cached ICN, although there have been many proposals to improve network performance, it is very difficult for these approaches to reach the optimal network performance with multiple metrics consideration. Therefore, in this paper, we propose a heuristic transferring strategy which selects some Content Routers (CRs) and combines them to facilitate the optimal network performance under a constrained total cache budget. At first, we quantify energy consumption, CR load, cache hit ratio and throughput, because the optimal network performance depends on four objects, i.e., minimizing energy consumption and CRs load as well as maximizing cache hit ratio and throughput. Then, based on the given network constraints and objects, we convert the CR transferring problem into 0-1 Knapsack Problem (KP01). Finally, in order to effectively obtain the optimal solution, we propose a heuristic approach based on Ant Colony Optimization (ACO) and expectation efficiency to solve KP01. The simulation is driven by the real YouTube dataset from campus network measurement over GTS and Deltacom topologies, and the experimental results demonstrate that the proposed strategy is more efficient compared to three baselines.

INDEX TERMS Heterogeneous ICN, transferring strategy, KP01, ACO, expectation efficiency.

I. INTRODUCTION

Information-Centric Networking (ICN) has attracted much attention from the global research communities in the past decade (2009-2019) due to its clean-slate architecture [1], [2]. To the best of our knowledge (according to the rough statistics referring to ACM, IEEE, Springer and Elsevier databases), the overwhelming majority of ICN achievements belong to the category of caching because the in-network caching is a considerably significant and differentiated feature of ICN. In addition, the in-depth study on ICN caching can greatly not only improve network performance and but also enhance Quality of Experience (QoE) of users. For example, a good caching policy, which pulls the frequently used contents to be cached at the edge Content Routers (CRs) can reduce the delivery delay and make users obtain the requested content in a relatively short time. We might as well do an assumption where there is no further research on the caching except the inherent ICN caching strategy [3]. If so, on one hand, the name-based ICN routing perhaps cannot show the efficient interest forwarding and content delivery; on the other hand, it is very difficult to find the appropriate application scenarios (e.g., edge computing [4], big data [5] and 5G [6]) for ICN. As a conclusion, the research on ICN caching has the non-negligible value.

The caching strategies usually involve four types of research categories [7], i.e., what are the cached contents, how to cache the contents, how many contents are cached and which CRs are exploited. To be specific, the first one pays attention to the caching of hot contents based on popularity prediction, content replacement, caching granularity or others; the second one studies the on-path/off-path caching, the collaborative/non-collaborative caching, the distributed/centralized caching or others; the third one focuses on the cache allocation which distributes different CRs for different cache sizes; the last one concentrates on the CRs transferring which selects and combines some CRs from all CRs. In fact, the cache-enabled CR is very expensive and

The associate editor coordinating the review of this manuscript and approving it for publication was Guangjie Han.

energy-cost. For example, a CR with 10TB cache can cost 30,0000 dollars and consume 500W at the full work [8]. It means that the research on cache allocation and CRs transferring is more significant than that on the other two problems. However, some practical conditions conduct that the CRs transferring should be in a more formal sense compared to the cache allocation. On one hand, if all CRs are opened and used to cache the contents during the process of work, it is very wasteful because the cache utilization of some CRs is very low. On the other hand, the demander (consumer) of ICN network usually pays a constrained and fixed total cache budget while hopes to obtain the optimal network performance and QoE of users.

By reviewing the above statements, CRs Transferring Problem (CTP) is described as: selecting some CRs from all CRs and combining them to facilitate the optimal network performance under a constrained total cache budget. In this paper, we use four common evaluation metrics (i.e., energy consumption, CR load, cache hit ratio and throughput) to measure the optimal network performance, that is, minimizing energy consumption and CR load as well as maximizing cache hit ratio and throughput. It is obvious that CTP is subject to the multi-objective optimization. In order to simply and effectively solve the multi-objective optimization problem with a total cache budget, we convert it into 0-1 Knapsack Problem (KP01) [9]. From the mathematical perspective, KP01 is an NP-hard problem which is addressed by two types of approaches, i.e., exact approach and heuristic approach. Although the exact approach can always produce the optimal solution, it performs the poor convergence when deploying the large-scale ICN network. Under such context, the heuristic approach is usually employed to solve KP01. In fact, the existing heuristic approaches include the general heuristics with the self-established rules (e.g., expectation efficiency [10] and differential evolution [11]) and the intelligent heuristics with the ready-made natural laws (e.g., bee colony algorithm [12] and butterfly optimization algorithm [13]), in which the former shows good execution efficiency and the latter shows the relatively optimal solution. Given this, we plan to integrate their advantages and design a hybrid heuristic approach for solving KP01.

This paper investigates CTP in ICN, and the main contributions are summarized as follows. (i) We quantify energy consumption, CR load, cache hit ratio and throughput and use them as four evaluation metrics of the optimal network performance. (ii) Based on minimizing energy consumption and CR load as well as maximizing cache hit ratio and throughput, we convert the multiple-objective optimization problem into KP01. (iii) We propose a heuristic approach based on Ant Colony Optimization (ACO) and expectation efficiency to solve KP01 so that the optimal solution (network performance) can be obtained effectively.

The rest paper is structured as follows. Section II reviews the related work. In Section III, CTP is presented and analyzed. Section IV introduces a heuristic approach for the optimal solution. The experimental results are reported in Section V and finally Section VI concludes this paper.

II. RELATED WORK

There have been lots of caching strategies. In this section, we review the related work from four aspects, what are the cached contents, how to cache the contents, how many contents are cached and which CRs are exploited.

Many strategies on what are the cache contents have been proposed. In terms of cache replacement, in [14], a content popularity and user locality based cache replacement strategy was devised, which put the hot contents at the appropriate CRs by analyzing the user locality. In [15], an energy-efficient cache replacement strategy was proposed by analyzing contents distribution and users distribution. In [16], authors proposed a fluid-based strategy for cache replacement, in which the fluid dynamics theory to reveal the time-evolving formulation process of request influences for a CR. Furthermore, in terms of other perspectives, in [17], an object-oriented packet caching was proposed, where the caching granularity was the packet-level rather than the whole content. In [18], authors explored the possibilities of improving the storage efficiency of the cache space based on global detour algorithm instead of the more common latency improvement so that the cache space was utilized to its most. In [19], a distributed probabilistic caching strategy was introduced, in which each CR made cache decision individually and cached the passing content with a certain probability.

Some approaches on how to cache the contents have also been proposed. For example, in [20], authors presented a proactive selective neighbor caching approach to enhance mobility support, by a simple procedure for selecting the appropriate subset of neighbors which considered the mobility behavior of users. In [21], a collaborative caching scheme was proposed to maximize the overall hit ratio by partitioning content space and hash-routing. In [22], inspired by taking an orthogonal approach by pro-actively eliminating redundancy via an independent intra-AS procedure, authors proposed an intra-AS cache cooperation scheme to effectively control the redundancy level and allow neighbor CRs to collaborate in serving each other's requests. In [23], a collaborative on-path and off-path caching policy was devised, in which the on-path CRs optimally stored the requested contents and were supported by a strategically placed central off-path CRs for the additional level of caching. In [24], a lightweight regional cache collaboration approach was proposed to share the popular contents among small-range CRs with the least cache information exchanges. It enabled collaborators to exchange the aggregated prefixes instead of data structures to fundamentally reduce the collaborating cost of communication and maintenance. In [25], a Bloom filter based collaboration caching approach was devised. It considered different forms of caching for different types of contents by setting the content lift time in accordance with its request frequency.

A number of cache strategies on how many contents are cached have also been investigated. For example, in [26], the in-network optimal content placement and cache allocation problem was formulated by the linear programming. It took into account capacity constraints on the downlink and uplink for provisioning the cache. In [27], a two-steps mechanism based on economics and game theory was proposed to solve heterogeneous cache allocation problem, where a predefined payment rule by auctions was used to decide the selling price of the storage unit. In [28], a practically feasible centrality-based heuristic method was leveraged to obtain the sub-optimal cache location in the SPT-tree rather than the dynamic programming in the exact method. In particular, it preferred the top centrality node as cache location, which dramatically reduced the computing complexity of finding the cache location. In [29], a utility-driven cache partitioning approach was devised to do cache allocation among multiple content providers. It partitioned a cache into a lots of slices and each one is dedicated to a particular content provider. In [30], authors proposed a max-min utility fairness scheme for distributing cache resources under the context of video delivery, based on the benefit of individual user and the relevance of video. In [31], a heuristic algorithm with threshold was designed to allocate cache resources reasonably, where the chunks with high popularity were cached at the leaf-CRs. In [32], authors regarded cache resource allocation as the convex programming relaxation problem. Then, they used game theory to solve it by combining with a distributed gradient estimation scheme.

Regarding which CRs are exploited, there have also been a few proposals. The previous researches usually concentrate on two views, i.e., edge caching and intermediate caching. For example, in [33], an edge caching with mobility prediction was proposed, which put the contents at the edge CRs as many as possible. In [34], a centrality-measures based scheme was proposed to select the core CRs; and in [35] the density-based spatial clustering was exploited to detect the core CRs, which put the contents at the intermediate CRs. Different from [33]–[36] fully considered their advantages and proposed a heuristic greedy strategy, which cached the most popular contents at the edge CRs, recalculated the relative popularity of each piece based on the request miss stream and then determined the contents to be cached at the core CRs. In spite of this, they did not really pay attention on CTP. To the best of our knowledge, there are almost no researches on it and some possible reasons are listed as follows. (i) From the network perspective, most researchers maintain that some edge CRs or/and some intermediate CRs can satisfy the caching of contents. (ii) In terms of consumer, most deployments assume that the cache resources are sufficient. (iii) In a mathematical sense, CTP is an NP-hard problem and the optimal network performance is easy to be ignored. However, the research on CTP with the multi-objective optimization is an indispensable part in the practical scenarios, which motivates this paper.

III. CTP MODELLING

This section includes three parts. At first, we formalize CTP with four performance evaluation metrics. Then, we quantify these four metrics. At last, we convert CTP into KP01. In particular, to make the readers more easily follow this paper, the frequently used abbreviations are listed in Table 1.

TABLE 1. Abbreviations in alphabetical order.

Abbreviation	Full name
ACHR	Average Cache Hit Ratio
ACL	Average CRs Load
ACO	Ant Colony Optimization
AEC	Average Energy Consumption
AT	Average Throughput
BFGCS	Baseline in Future Generation Computer Systems
BMTA	Baseline in Multimedia Tools and Applications
BCN	Baseline in Computer Networks
CR	Content Router
CTP	CRs Transferring Problem
HAEE	Heuristic based on ACO and Expectation Efficiency
ICN	Information-Centric Networking
KP01	0-1 Knapsack Problem

A. CTP FORMALIZATION

In this paper, the heterogeneous-cached ICN topology with n CRs is expressed as $G = (V, E, C)$, where V is the set of CRs, E is the set of edges and C is the set of cache capacities. Here, V, E and C are defined as follows.

$$\begin{aligned} V &= \{CR_i \mid 1 \le i \le n\} \\ E &= \{e_{ij} \mid 1 \le i, j \le n\} \\ C &= \{c_i \mid 1 \le i \le n\} \end{aligned}$$

where c_i is the allocated cache size. In addition, as the cache scenario is heterogeneous, it exists $c_i \neq c_j$.

Before modelling CTP, we first give the definition of CTP, as follows.

Definition 1 (CTP): Given a constrained total cache budget C_{gt}, m CRs are selected and combined to facilitate the optimal network performance, where $m < n$ and $\sum_{i=1}^{n} c_i = C_{gt}$.

As a matter of fact, the optimal network performance in **Definition 1** is an abstract object. To embody the abstract object, it requires to employ several metrics for the evaluation of network performance. In this paper, we use energy consumption, CR load, cache hit ratio and throughput to evaluate the network performance. Based on this, the optimal network performance is concluded as obtaining the smallest energy consumption and CR load besides the highest cache hit ratio and throughput.

Let ec_i, lo_i, hr_i and tp_i denote the consumed energy, load, cache hit ratio and throughput in terms of CR_i respectively, and we have the definition of optimal network performance, as follows.

Definition 2 (Optimal Network Performance): Given $x_i = \{0, 1\}$, here $x_i = 0$ means that CR_i is selected otherwise not, and four objects need to be optimized, that is,

$$\begin{cases} Minimize \;\; E_{total} = \sum_{i=1}^{n} ec_i x_i \\ Minimize \;\; L_{total} = \sum_{i=1}^{n} lo_i x_i \\ Maximize \;\; H_{total} = \sum_{i=1}^{n} hr_i x_i \\ Maximize \;\; T_{total} = \sum_{i=1}^{n} tp_i x_i \end{cases} \tag{1}$$

where E_{total}, L_{total}, H_{total} and T_{total} are the consumed energy, load, cache hit ratio and throughput in terms of the whole network respectively.

With **Definition 2** consideration, the **Definition 1** is re-described in mathematic, as the following formula.

$$\begin{cases} Subject\ to\ \sum_{i=1}^{n} c_i x_i \leq C_{gt} \\ Minimize\ \ E_{total}, L_{total} \\ Maximize\ \ H_{total}, T_{total} \end{cases} \tag{2}$$

B. METRICS QUANTIFICATION

This section builds models for four metrics. At first, we quantify ec_i. As a rule, the consumed energy at CR depends on the transmitted traffic and the fixed configuration [37], where different network devices are configured with different details. In fact, the same network usually deploys the same network devices due to the convenient implementation, thus the main concern on the modelling of energy consumption is related to the transmitted traffic. Regarding the relationship between energy consumption and traffic, we consider it as the linearity. Given this, ec_i is defined as follows.

$$ec_i = Pt_i + \xi tra_i \tag{3}$$

where P is the fixed power of CR, ξ is the consumed energy in cased of processing a bit traffic, tra_i is the transmitted traffic via CR_i, and t_i is the working time of CR_i.

Then, we quantify lo_i that is expressed by the traffic, and we have

$$lo_i = tra_i \tag{4}$$

which indicates that the larger traffic causes more serious load.

Next, we quantify hr_i which is defined as follows.

$$hr_i = \frac{N_i^{suc}}{N_i^{rec}} \tag{5}$$

where N_i^{rec} is the number of interest requests received by CR_i and N_i^{suc} is the number of interest requests successfully satisfied by CR_i.

At last, we quantify tp_i which is defined as the processed quantity of traffic per unit time, and we have

$$tp_i = \frac{tra_i}{t_i} \tag{6}$$

In summary, equations (3), (4), (5) and (6) show the embodied forms of energy consumption, CR load, cache hit ratio and throughput respectively.

C. KP01 CONVERSION

According to the above, we know that CTP tries to optimize four objects with a constraint, which is considerably complex. Thus, we convert it into KP01 for the convenient and simple computation. Let gp_i denote the generated performance at CR_i, and we have

$$gp_i \propto (ec_i, lo_i, hr_i, tp_i)$$

which is an abstract formula. As four metrics have different units, we use the min-max method [38] to standardize ec_i, lo_i, hr_i and tp_i, and the results are denoted by ec'_i, lo'_i, hr'_i and tp'_i respectively. By reviewing equations (3-6), we observe that ec_i, lo_i and tp_i are linearly proportional to tra_i, while hr_i is not related to tra_i. Based on this, we modify gp_i as follows.

$$gp_i = \alpha hr'_i + \beta(tp'_i - ec'_i - lo'_i) \tag{7}$$

where α and β are two important parameters.

Let Op_{total} denote the total network performance, and CTP is converted into KP01, as follows.

$$\begin{cases} Subject\ to\ \ \sum_{i=1}^{n} c_i x_i \leq C_{gt} \\ Maximize\ \ Op_{total} = \sum_{i=1}^{n} gp_i \end{cases} \tag{8}$$

Combine KP01, and we redescribe CTP as follows. Given n CRs, where CR_i owns c_i and gp_i, and a knapsack holds a fixed capacity C_{gt}, the goal of KP01 is to select some possible CRs to be loaded into knapsack so that the total network performance generated by the selected CRs is the optimal while the total cache size of these CRs is not larger than C_{gt}.

IV. HEURISTIC STRATEGY

In this section, we introduce a hybrid heuristic strategy to solve the converted KP01. To be specific, ACO [38], [39] is used to select some initial CRs which are removed from knapsack no longer and the improved expectation efficiency model is used to determine which CRs can be loaded into knapsack in terms of the remaining CRs.

A. PRELIMINARY

In [10], we have proposed an efficient and effective strategy based on expectation efficiency to solve KP01, including two stages. At first, a greedy degree model (please see equation (2) in [10] for details) inspired by greedy strategy is devised to select Q items (called CRs in this paper) as the initial determination; in particular, the selected items are never removed from the knapsack at the later stage. Then, the expectation efficiency model (please see equation (8) in [10] for details) is devised to select some items from the remaining $n-Q$ items and put them into the knapsack, in which each item owns one expectation efficiency value. In spite of this, the strategy still exits two limitations that have been mentioned. On one hand, the determination of Q is imprecise enough, because some item that belongs to these Q items is not the part of the optimal solution while it is loaded into the knapsack, which can cause the large deviation with the optimal solution. On the other hand, the number of calculations with respect to expectation efficiency model is $n - Q$ which is high (that is to say, computing expectation efficiency value for each item, especially for the last some items is unnecessary), at the same time, the model's formula is a little complex (that is to say, the slightly simpler formula should be devised). As a result, we optimize the two points in this paper.

B. ACO FOR INITIAL CRs

We improve ACO and use it to determine the initial Q CRs. For ICN topology, let r_i denote the influence rate of CR_i (the large r_i means that CR_i has the high probability to be loaded into knapsack), and it is defined as a ratio between the generated performance and the allocated cache size, as follows.

$$r_i = \frac{gp_i}{c_i} \tag{9}$$

Let $cost_{ij}$ denote the weight of e_{ij}, and we have

$$cost_{ij} = \frac{1}{|r_i - r_j|} \tag{10}$$

The updating of pheromone is considerably important for ACO. Let $\tau_{ij}(I)$ denote the total pheromone over e_{ij} after I iterations, according to the discrete updating strategy of pheromone, we have

$$\tau_{ij}(I) = (1-\rho)\tau_{ij}(I-1) + aph_{ij}(I) \tag{11}$$

where ρ is a volatilization coefficient of pheromone, $1-\rho$ is a residual factor of pheromone and $aph_{ij}(I)$ is the pheromone accumulated by a number of ants after the I-th iteration. Assume that there are Na ants, for any ant ant_λ (here $1 \leq \lambda \leq Na$), and $aph_{ij}(I)$ is defined as follows.

$$aph_{ij}(I) = \sum_{\lambda=1}^{Na} aph_{ij}^{\lambda}(I) y_\lambda \tag{12}$$

where $y_\lambda = \{0, 1\}$, here $y_\lambda = 1$ means that ant_λ goes through e_{ij} otherwise not, and $aph_{ij}^{\lambda}(I)$ is the pheromone over e_{ij} left by ant_λ after the I-th iteration. Next, we define $aph_{ij}^{\lambda}(I)$ by considering two perspectives: one is the ant behaviors and the other one is the inherent features of ICN. Let L_λ and hop_j denote the total cost traversed by ant_λ within one iteration and the hop count between interest requester and content provider respectively, and we have

$$aph_{ij}^{\lambda}(I) = \kappa \frac{1}{hop_j L_\lambda I} \tag{13}$$

where κ is a regulatory factor to avoid the situation where $hop_j L_\lambda I$ becomes too large or small.

For the forwarding probability of ant_λ, it is related to the pheromone but not limited to the pheromone. If the forwarding probability only relies on the pheromone, the solution is easy to trap in the local optimum. Given this, we need to find another factor to conduct the possibility of the diverse forwarding. Based on the situation where all CRs are arranged by r_i in descending order, let d_{ij} denote the location distance between CR_i and CR_j, and we have

$$d_{ij} = |d_i - d_j| \tag{14}$$

where d_i and d_j are the location numbers of CR_i and CR_j respectively.

Based on pheromone and location distance, let $fp_{ij}^{\lambda}(I)$ denote the forwarding probability of ant_λ from CR_i and CR_j, and we have

$$fp_{ij}^{\lambda}(I) = \frac{[\tau_{ij}(I)]^{\nu}[d_{ij}]^{\omega}}{\sum_{CR_k \in Aw_i^{\lambda}} [\tau_{ik}(I)]^{\nu}[d_{ik}]^{\omega}} \tag{15}$$

where Aw_i^{λ} is a set of CRs which can be used to receive ant_λ; ν and ω are the inspired factors of pheromone and location distance respectively. In particular, both large pheromone and location distance can further conduct the forwarding of ant.

We introduce the method of determining Q based on ACO as follows. At first, we arrange n CRs by r_i in descending order, in which the location number of CR_i is probably not equal to i. Let c'_i denote the allocated cache size of CR whose location number is i, we have the following constraint condition.

$$C_{gt} - \sum_{i=1}^{\theta} c'_i \geq 0 \tag{16}$$

Then, for each location number, if inequality (16) is satisfied, we can obtain some shortest paths, and each shortest path corresponds to a location number. It is obvious that the number of shortest paths is equal to θ; in other words, ACO is used for θ times. Finally, for θ location numbers, we regard the location number whose shortest path value is the minimal as Q. According to the above statements, the pseudo-code of Q determination is described in **Algorithm 1**, where I_{max} is the maximal number of iterations for each usage of ACO. In particular, the first "**for**" handles each location, the second one handles each iteration of ACO, the third one handles each CR and the last one handles each ant.

Algorithm 1 Q Determination Based on ACO

Input: Na ants, C_{gt}, G
Output: Q

```
for i = 1 to Q, do
    for I = 1 to I_max, do
        for j = 1 to n, do
            for λ = 1 to Na, do
                Compute τ_ij(I) with equation (11);
                Compute d_ij with equation (14);
                Forward ant_λ with equation (15);
            endfor
        endfor
    endfor
    Obtain one shortest path and one location number;
    if inequality (16) is not satisfied, then
        Break;
    endif
    Obtain Q shortest path values;
    Q = the location number whose shortest path value
is smallest;
endfor
```

C. EXPECTATION EFFICIENCY FOR REMAINING CRs

As the above mentioned, to compute expectation efficiency values for the last some CRs is unnecessary. In addition, by reviewing equation (8) in [10], its format is a little complex and also need to be optimized. Thus, we improve expectation efficiency model and use it to determine which remaining CRs can be loaded into the knapsack.

1) COMPUTATION OPTIMIZATION

Given a boundary ψ, if

$$\begin{cases} \sum_{i=1}^{\psi} c'_i \leq C_{gt} \\ \sum_{i=1}^{\psi+1} c'_i > C_{gt} \end{cases}$$

is satisfied, it means that the maximal greedy degree is ψ. On this basis, we scan the following $n - \psi$ CRs, if

$$c'_i > C_{gt} - \sum_{i=1}^{\psi} c'_i, \quad \psi + 1 \leq i \leq n \tag{17}$$

is satisfied, it means that the i-th CR cannot be loaded into knapsack. Suppose that the number of CRs which satisfy constraint condition (17) is Q', and the number of calculations with respect to expectation efficiency model is $n - Q - Q'$ rather than the pervious $n - Q$.

2) FORMULA OPTIMIZATION

Based on the above symbols and computation optimization, the expectation efficiency model in [10] is defined as follows.

$$\begin{aligned} &eef_i \\ &= \frac{r_i}{r_{i-1}} \\ &\quad * \frac{r_{i-1}\left(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k\right) - (n-i+1)gp'_i}{\left(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k\right) - (n-i+1)c'_i}, \\ &\quad i \in [Q+1, n] - A \end{aligned} \tag{18}$$

where A is a set of CRs which satisfy constraint condition (17).

In fact, r_i/r_{i-1} is a ratio of two influence rates and it is used to coordinate the second part of equation (18); $r_{i-1}(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k)$ is used to evaluate the corresponding network performance generated by the rest cache size. Instead, r_i/r_{i-1} is removed and the r_{i-1} of $r_{i-1}(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k)$ is optimized by referring to these CRs which have been loaded into knapsack. Let gfr denote the global influence rate in case of loading the i-th CR into knapsack, and we have

$$gfr_i = \frac{\sum_{k=1}^{Q} gp'_k + \sum_{k=Q+1}^{i} gp'_k x_k}{\sum_{k=1}^{Q} c'_k + \sum_{k=Q+1}^{i} c'_k x_k} \tag{19}$$

Let eef' denote the improved expectation efficiency value of the i-th CR, and the simplified model is defined as follows.

$$\begin{aligned} &eef'_i \\ &= \frac{gfr_{i-1}\left(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k\right) - (n-i+1)gp'_i}{\left(C_{gt} - \sum_{k=1}^{Q} c'_k - \sum_{k=Q+1}^{i-1} c'_k x_k\right) - (n-i+1)c'_i}, \\ &\quad i \in [Q+1, n] - A \end{aligned} \tag{20}$$

Based on computation optimization and formula optimization, the improved expectation efficiency strategy for handling the remaining $n - Q - Q'$ CRs is introduced as follows. At first, $n - Q - Q'$ expectation efficiency values are computed according to equation (20) and are arranged in descending order. Then, the corresponding CRs are put into the knapsack one by one while the total cache size cannot exceed C_{gt}. The pseudo-code of the improved expectation efficiency is described in **Algorithm 2**.

Algorithm 2 The Improved Expectation Efficiency

Input: Q, C_{gt}
Output: Op_{total}

```
for i = Q + 1 to n, do
    if constraint condition (17) is not satisfied, then
        Compute eef'_i with equation (20);
    else
        Continue;
    endif
endfor
n = n − Q − Q';
Arrange these expectation efficiency values;
for j = Q + 1 to n, do
    Put the j-th CR into knapsack;
    if the total cache size exceeds C_gt, then
        Break;
    endif
    Op_total = gp'_j + Σ_{i=1}^{Q} gp'_i;
endfor
```

V. SIMULATION RESULTS

A. SETUP

The proposed Heuristic transferring strategy based on ACO and Expectation Efficiency (HAEE) is implemented via two parts: one is the implementation of CTP based on NS3 [40] and the other one is the implementation of KP01 based on Visual Studio, running on a personal computer with Intel(R) core(TM)i5-6200u, CPU2.92 GHZ, 4GB RAM. The simulation is driven based on the real YouTube dataset [41], of which the collection of trace comes from a campus network measurement, including 18751 user requests, 13764 short videos and 2377 hosts. In particular, the performance evaluation is done over GTS topology (149 nodes and 193 edges) [42] and Deltacom topology (113 nodes and 183 edges) [43], as shown in Figs. 1 and 2 respectively.

Furthermore, we compare the proposed HAEE with three the-state-of-the-art mechanisms, i.e., edge caching [33], core

FIGURE 1. Europe-GTS with 149 nodes and 193 edges.

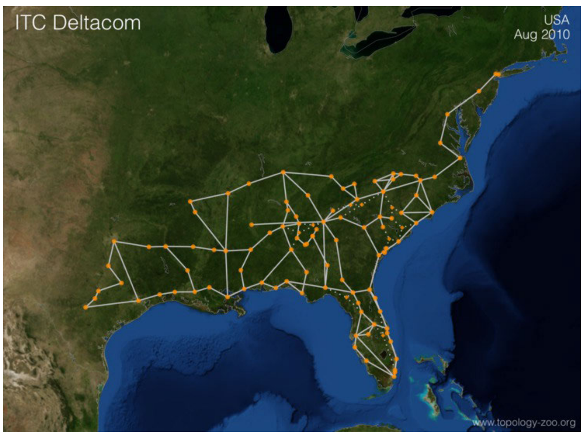


FIGURE 2. USA-Deltacom with 113 nodes and 183 edges.

caching [34] and greedy caching [36], called Baseline in Future Generation Computer Systems (BFGCS), Baseline in Multimedia Tools and Applications (BMTA) and Baseline in Computer Networks (BCN) respectively. In addition, Average Energy Consumption (AEC), Average CRs load (ACL), Average Cache Hit Ratio (ACHR) and Average Throughput (AT) are considered as four evaluation metrics. In terms of comparison experiments, we divide these 18751 interest requests into five intervals in chronological order. For each interval, we extract 400 interest requests, i.e., [1,400], [3751,4150], [7501,7900], [11251,11650] and [15001,15400] and report the corresponding experimental results. As shown in Table 2, we give the settings for the involved simulation parameters.

TABLE 2. Simulation settings.

Parameter	Setting
Collection duration	24h
YouTube dataset size	166GB
Simulation times	30
Network bandwidth	10Gb/s
Na in Algorithm 1	100
I_{max} in Algorithm 1	60
P in equation (3)	7296W
ξ in equation (3)	3×10^{-7}J/bit
α in equation (7)	0.3
β in equation (7)	0.7
ρ in equation (11)	0.45
κ in equation (13)	2
ν in equation (15)	1
ω in equation (15)	1

B. CACHE COMBINATION ANALYSIS

In order to test the proposed HAEE, we give the cache deployment of two topologies at first. To be specific, the nodes are numbered from left to right and top to bottom, and the cache size of each node is allocated varying from 50M to 100M in a random style. In this way, the heterogeneous-cached ICN topologies are completed. In addition, the constrained total cache budget is set as 6809M in GTS topology and 4792M in Deltacom topology. Based on the above, the allocation results and combination results with respect to two topologies are shown as follows.

(i) Cache allocation results over GTS: $C = \{CR_1, CR_2, \cdots, CR_{148}, CR_{149}\} =\{$ 59, 81, 91, 91, 98, 90, 88, 76, 97, 54, 70, 73, 67, 94, 56, 60, 76, 61, 52, 53, 75, 92, 77, 75, 88, 98, 68, 56, 82, 51, 63, 71, 70, 93, 94, 77, 90, 94, 97, 74, 65, 77, 79, 66, 68, 73, 64, 58, 73, 68, 82, 90, 69, 77, 85, 95, 71, 63, 90, 64, 94, 75, 55, 96, 66, 78, 56, 96, 79, 87, 72, 71, 93, 94, 55, 77, 54, 71, 90, 55, 84, 82, 65, 89, 77, 50, 66, 54, 88, 90, 71, 50, 53, 90, 78, 76, 76, 97, 68, 73, 54, 68, 54, 88, 93, 77, 92, 81, 76, 68, 89, 71, 99, 73, 89, 68, 99, 79, 83, 92, 50, 83, 72, 96, 77, 88, 73, 60, 95, 99, 75, 72, 50, 51, 80, 64, 56, 81, 67, 68, 92, 84, 70, 82, 97, 97, 89, 78, 87$\}$.

(ii) Cache allocation results over Deltacom: $C = \{CR_1, CR_2, \cdots, CR_{112}, CR_{113}\} =\{$ 59, 81, 91, 91, 98, 90, 88, 76, 97, 54, 70, 73, 67, 94, 56, 60, 76, 61, 52, 53, 75, 92, 77, 75, 88, 98, 68, 56, 82, 51, 63, 71, 70, 93, 94, 77, 90, 94, 97, 74, 65, 77, 79, 66, 68, 73, 64, 58, 73, 68, 82, 90, 69, 77, 85, 95, 71, 63, 90, 64, 94, 75, 55, 96, 66, 78, 56, 96, 79, 87, 72, 71, 93, 94, 55, 77, 54, 71, 90, 55, 84, 82, 65, 89, 77, 50, 66, 54, 88, 90, 71, 50, 53, 90, 78, 76, 76, 97, 68, 73, 54, 68, 54, 88, 93, 77, 92, 81, 76, 68, 89, 71, 99$\}$.

(iii) Combination results over GTS: $Q = 57$, and the corresponding set of CR numbers is {19, 18, 10, 92, 134, 112, 50, 1, 114, 83, 75, 13, 148, 65, 63, 71, 93, 86, 31, 106, 100, 137, 47, 149, 27, 111, 84, 122, 79, 4, 55, 108, 82, 56, 28, 54, 94, 39, 87, 52, 89, 95, 34, 96, 11, 126, 98, 128, 6, 64, 66, 5, 8, 41, 110, 105, 104}. The number of CRs loaded into knapsack by expectation efficiency model is 31, and the corresponding set of CR numbers is {124, 44, 48, 57, 140, 7, 109, 40, 146, 136, 68, 99, 62, 90, 135, 127, 43, 78, 35, 116, 70, 107, 119, 131, 14, 132, 12, 22, 36, 46, 123}.

(iv) Combination results over Deltacom: $Q = 37$, and the corresponding set of CR numbers is {19, 18, 10, 92, 112,

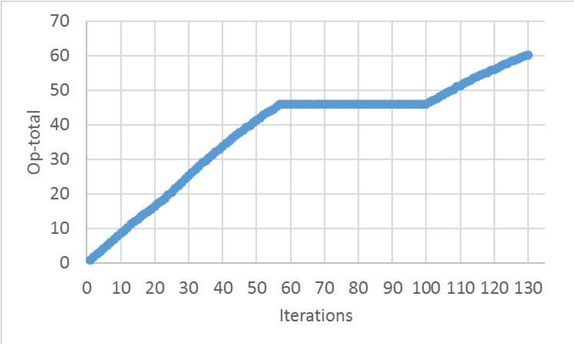


FIGURE 3. The change of Op_{total} over GTS.

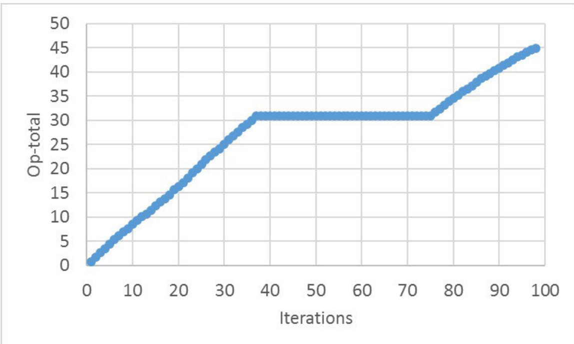


FIGURE 4. The change of Op_{total} over Deltacom.

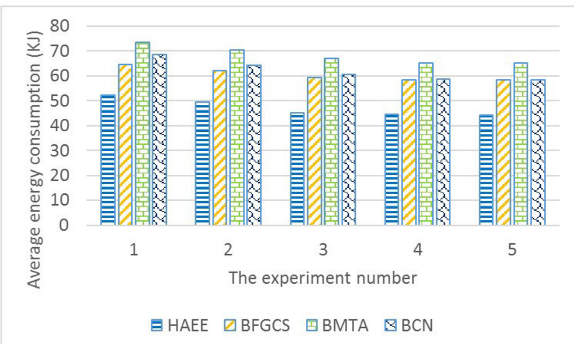


FIGURE 5. Average energy consumption over GTS in terms of different experiment numbers.

50, 1, 83, 75, 13, 71, 63, 86, 31, 106, 100, 93, 65, 111, 47, 27, 4, 79, 108, 84, 55, 82, 54, 28, 56, 39, 87, 89, 94, 95, 96, 52}. The number of CRs loaded into knapsack by expectation efficiency model is 23, and the corresponding set of CR numbers is {34, 11, 64, 5, 8, 41, 98, 110, 66, 6, 105, 44, 57, 104, 48, 7, 109, 40, 68, 99, 90, 62, 78}.

Furthermore, according to the computation results generated from the testing of YouTube dataset and the cache allocation results, we report the corresponding change of Op_{total} in Figs. 3 and 4.

In Fig. 3, Op_{total} of GTS topology is obtained by the combination of 88 CRs, via 131 iterations. Among them, the first 57 iterations mean that $Q = 57$ based on ACO and the corresponding 57 CRs are loaded into knapsack one by one; the following 43 iterations mean that it only requires 43 CRs (smaller than 149-57 = 92 CRs) to be performed by expectation efficiency model; the last 31 iterations means that only 31 CRs can be loaded into knapsack. Similarly, in Fig. 4, Op_{total} of Deltacom topology is obtained by the combination of 60 CRs, via 98 iterations. Among them, the first 37 iterations mean that $Q = 37$ based on ACO and the corresponding 37 CRs are loaded into knapsack one by one; the following 38 iterations mean that it only requires 38 CRs (smaller than 113-37 = 76 CRs) to be performed by expectation efficiency model; the last 23 iterations means that only 23 CRs can be loaded into knapsack.

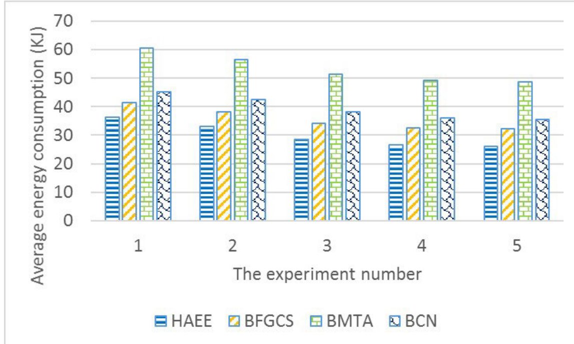


FIGURE 6. Average energy consumption over Deltacom in terms of different experiment numbers.

C. COMPARISON ANALYSIS

1) AVERAGE ENERGY CONSUMPTION

The energy consumption is obtained by equation (3). AECs of HAEE, BFGCS, BMTA and BCN over GTS and Deltacom topologies in terms of five different experiments are reported in Figs. 5 and 6. We observe that the proposed HAEE always has the smallest AEC, followed by BFGCS, BCN and BMTA. To be specific, HAEE uses the smallest number of CRs to reach the optimal network performance. On one hand, the consumed energy due to opening CRs is the smallest; on the other hand, the cache utilization rate in terms of these selected CRs is the highest, and there is no more congested traffic consuming energy. As a result, HAEE has the optimal AEC. For the three baselines, BFGCS only caches the contents at the edge CRs and most CRs are not opened, which saves a lot of energy. Thus, BFGCS has smaller AEC than BMTA and BCN. For the remaining two baselines, BMTA considers centrality measures including closeness centrality, reach centrality, degree centrality and betweeness centrality, which requires to transmit the traffic via most CRs and obtain the measured results. Given this, BMTA has larger AEC than BCN.

2) AVERAGE CRs LOAD

The total CRs load is obtained by equation (4). ACLs of HAEE, BFGCS, BMTA and BCN over GTS and Deltacom topologies in terms of five different experiments are reported in Figs. 7 and 8. We observe that the proposed HAEE always has the smallest ACL, followed by BMTA, BCN and BFGCS. Similar to the pervious section, HAEE aims to facilitate the

TABLE 3. The average reduced rate of traffic over GTS and Deltacom in terms of different experiment numbers (%).

Topology	GTS					Deltacom				
No.	1	2	3	4	5	1	2	3	4	5
HAEE	62.43	66.20	70.75	72.69	**73.40**	69.32	71.49	75.19	76.26	**76.42**
BFGCS	39.50	43.67	48.74	50.97	51.63	44.99	49.22	51.72	52.82	53.11
BMTA	56.12	58.88	61.80	63.66	64.50	63.23	65.92	68.48	69.81	69.98
BCN	52.23	54.50	58.22	60.08	60.97	59.75	61.66	64.14	65.47	66.14

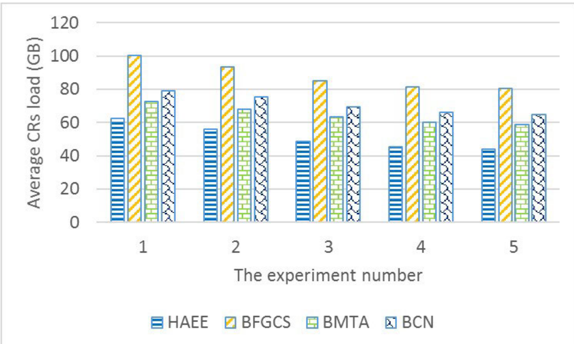


FIGURE 7. Average CRs load over GTS in terms of different experiment numbers.

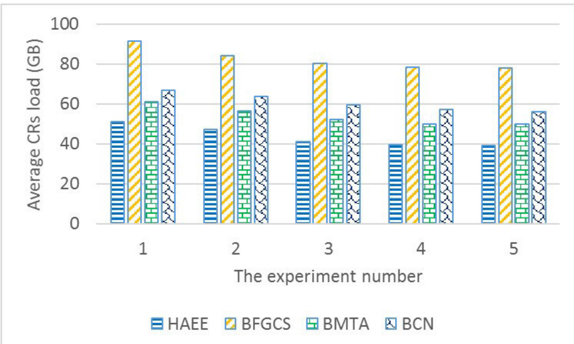


FIGURE 8. Average CRs load over Deltacom in terms of different experiment numbers.

optimal network performance including CRs load which is related to the traffic. Because KP01 is exploited to select the most appropriate combination of CRs, HAEE has the highest ability to handle the transmitted traffic. It is obvious that HAEE has no the load pressure and thus has the optimal ACL. Different from the previous section, BMTA has smaller ACL than BCN and BFGCS, this is because it analyzes the global network traffic by considering closeness centrality, reach centrality, degree centrality and betweeness centrality, which causes that HAEE has higher efficiency to transmit the traffic than BCN and BFGCS. Thus, HAEE faces the lighter load than BCN and BFGCS. For the last two baselines, BCN has higher transmission efficiency for traffic than BFGCS, because it first puts the popular contents at the core CRs and then puts the secondary popular contents at the edge CRs. The relatively comprehensive combination of core caching and edge caching can greatly relieve CRs load, thus BCN has smaller ACL than BFGCS.

Furthermore, Table 3 shows the average reduced rate of traffic which is defined as the difference between 1 and the ratio of ACL and the YouTube dataset size (166GB). We observe that the average reduced rate of HAEE can reach 73.4% over GST and 76.42% over Deltacom in case of the late stage, which further indicates that the proposed HAEE has the satisfactory performance optimization on CRs load.

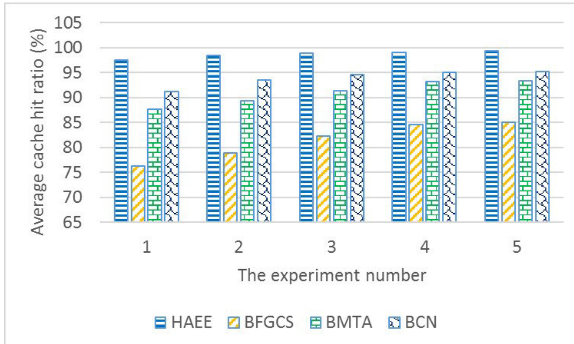


FIGURE 9. Average cache hit ratio over GTS in terms of different experiment numbers.

3) AVERAGE CACHE HIT RATIO

The cache hit ratio is obtained by equation (5). ACHRs of HAEE, BFGCS, BMTA and BCN over GTS and Deltacom topologies in terms of five different experiments are reported in Figs. 9 and 10. We observe that the proposed HAEE always has the highest ACHR, followed by BCN, BMTA and BFGCS. In particular, ACHR of HAEE can reach around 99.43% over GTS and 99.51% over Deltacom, which suggests that these selected CRs based on network analysis, ACO and expectation efficiency can create the satisfactory ACHR. Different from BMTA and BFGCS, BCN leverages the collaboration of edge caching and core caching, of which the distributions of interest requests and user locations are considered. In addition, if the core CRs do not be matched, the edge CRs play the important role to try their best to satisfy these requests, because the edge CRs always cache the secondary popular contents. In other words, BCN almost caches the frequently used contents at the network while the corresponding caching locations are dynamic. Given the above statements, BCN has higher ACHR than BMTA and BFGCS. As the above section mentioned, BMTA analyzes many network factors and it can satisfy interest requests at the most extent. Thus, BMTA has higher ACHR than BFGCS.

4) AVERAGE THROUGHPUT

The throughput is obtained by equation (6). ATs of HAEE, BFGCS, BMTA and BCN over GTS and Deltacom topologies in terms of five different experiments are reported in Figs. 11 and 12. We observe that BFGCS has the largest AT, followed by HAEE, BCN and BMTA, that is to say, the proposed HAEE only has the suboptimal AT. The reasons

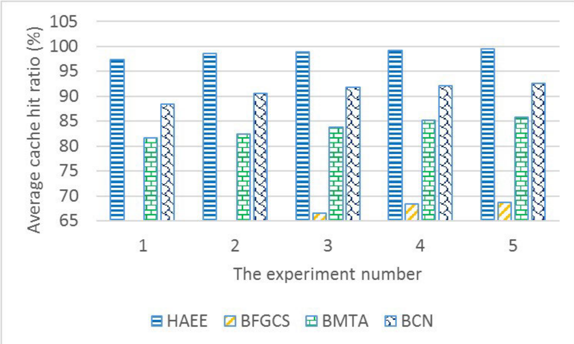


FIGURE 10. Average cache hit ratio over Deltacom in terms of different experiment numbers.

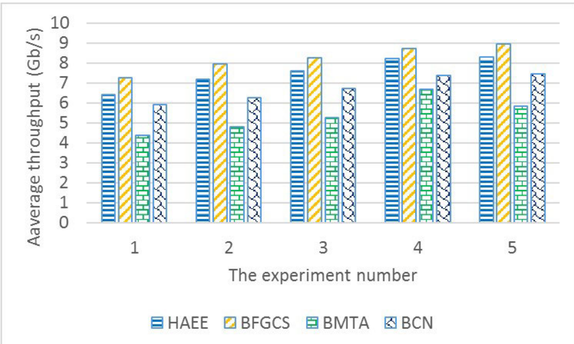


FIGURE 11. Average throughput over GTS in terms of different experiment numbers.

on why BFGCS has the largest AT are concluded as two aspects. On one hand, BFGCS uses cloud computing and 5G techniques to accelerate the transmission of traffic although it faces the most serious CRs load. In other words, the low delay results in the large AT. On the other hand, HAEE introduces ACO to determine the number of CRs loaded into knapsack in advance, which increases the certain network delay. For HAEE, BCN and BMTA, although HAEE introduces ACO, its AT is always larger than BCN and BMTA. Two related reasons are listed as follows. At first, HAEE has smaller ACL than BCN and BMTA, which can be found from Figs. 7 and 8. Then, BCN considers the collaboration of edge caching and core caching while BMTA considers closeness centrality, reach centrality, degree centrality and betweness centrality, both of them analyze the total traffic, which consumes much more time compared to the introduction of ACO. As a result, HAEE has larger AT than them. For the remaining two baselines, BMTA only caches the contents at the core CRs and faces more serious CRs load, thus it has smaller AT than BCN.

D. DISCUSSION

By reviewing the experimental results reported in Section V-C, we know that the proposed HAEE has the optimal network performance on AEC, ACL and ACHR as well as the suboptimal AT. Although HAEE does not reach the optimal network performance on AT, it has the close performance to BFGCS and reaches around 8.3Gb/s over GTS and 8.8Gb/s over Deltacom when the interest requests are stable.

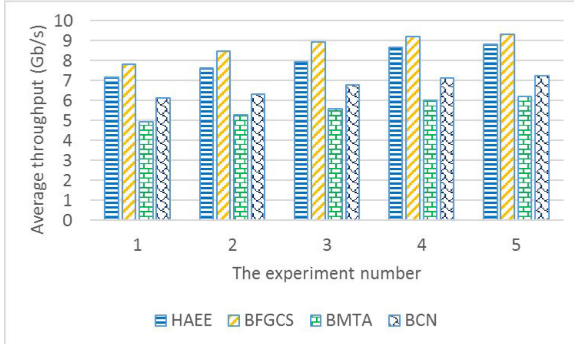


FIGURE 12. Average throughput over Deltacom in terms of different experiment numbers.

However, under the condition where the network bandwidth is 10Gb/s, the obtained AT by HAEE is considerably acceptable. In fact, we also can exploit other techniques to accelerate the transmission of traffic similar to BFGCS that uses cloud computing and 5G, such as DPDK (Data Plane Development Kit) [44] by bypassing the kernel and network coding by improving cache utilization rate. In summary, by evaluating the above four metrics, we think that the proposed HAEE can facilitate the optimal network performance.

VI. CONCLUSION

This paper investigates CTP of ICN to facilitate the optimal network performance, i.e., obtaining the minimal energy consumption and CR load as well as the maximal cache hit ratio and throughput. In terms of the multi-objective optimization problem, we convert it into KP01 based on the given network constraints and four objects. Furthermore, we propose a hybrid heuristic strategy to solve KP01 including two parts. At first, we improve ACO and use it to determine the number of CRs loaded into knapsack in advance. Then, we improve the expectation efficiency model via computation optimization and formula optimization to select how many and which CRs can be loaded into knapsack from the remaining CRs. The proposed heuristic transferring strategy is simulated based on the real YouTube dataset over GTS and Deltacom topologies, and the comparison experiments reveal that the proposed strategy outperforms three the-state-of-the-art benchmarks in terms of energy consumption, CR load, cache hit ratio and throughput.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. ACM CoNEXT*, 2009, pp. 1–12.

[2] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2014.

[3] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 62–67, Jun. 2012.

[4] G. Panwar, R. Tourani, T. Mick, A. Mtibaa, and S. Misra, "DICE: Dynamic multi-RAT selection in the ICN-enabled wireless edge," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 5, pp. 67–72, Oct. 2017.

[5] P. T. Dat, A. Kanno, N. Yamamoto, and T. Kawanishi, "5G transport networks: The need for new technologies and standards," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 18–26, Sep. 2016.

[6] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5g mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May 2015.

[7] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[8] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw. (ICN)*, 2011, pp. 44–49.

[9] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 5, pp. 525–530, Sep. 1978.

[10] J. Lv, X. Wang, M. Huang, H. Cheng, and F. Li, "Solving 0-1 knapsack problem by greedy degree and expectation efficiency," *Appl. Soft Comput.*, vol. 41, pp. 94–103, Apr. 2016.

[11] S. Jun and L. Jian, "Solving 0-1 knapsack problems via a hybrid differential evolution," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, 2009, pp. 134–137.

[12] S. Zhang and S. Liu, "A discrete improved artificial bee colony algorithm for 0-1 knapsack problem," *IEEE Access*, vol. 7, pp. 104982–104991, 2019.

[13] Y. Feng, G.-G. Wang, W. Li, and N. Li, "Multi-strategy monarch butterfly optimization algorithm for discounted 0-1 knapsack problem," *Neural Comput. Appl.*, vol. 30, pp. 3019–3036, Nov. 2018.

[14] F.-H. Tseng, W.-C. Chien, S.-J. Wang, C. F. Lai, and H.-C. Chao, "A novel cache scheme based on content popularity and user locality for future Internet," in *Proc. 27th Int. Conf. Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2018, pp. 1–5.

[15] X. Zheng, G. Wang, and Q. Zhao, "A cache placement strategy with energy consumption optimization in information-centric networking," *Future Internet*, vol. 11, no. 64, pp. 1–16, 2019.

[16] F. Wu, W. Yang, G. He, "FDCP: Cache placement model for information-centric networking using fluid dynamics theory," *Peer-Peer Netw. Appl.*, vol. 12, no. 4, pp. 789–808, 2019.

[17] Y. Thomas, G. Xylomenos, C. Tsilopoulos, and G. C. Polyzos, "Object-oriented packet caching for ICN," in *Proc. 2nd Int. Conf. Inf.-Centric Netw.*, 2015, pp. 89–97.

[18] S. Saha, M. Chauhan, and A. Lukyanenko, "Beyond the limits: Maximization of ICN caching capabilities with global detour algorithm," in *Proc. IEEE Symp. Comput. Commun.*, Jul. 2015, pp. 846–853.

[19] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, "Design and evaluation of probabilistic caching in information-centric networking," *IEEE Access*, vol. 6, pp. 32754–32768, 2018.

[20] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, "Proactive selective neighbor caching for enhancing mobility support in information-centric networks," in *Proc. ACM SIGCOMM*, 2012, pp. 61–66.

[21] S. Wang, J. Bi, J. Wu, "Collaborative caching based on hash-routing for information-centric networking," in *Proc. ACM SIGCOMM*, 2013, pp. 535–536.

[22] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *Proc. ACM SIGCOMM*, 2013, pp. 61–66.

[23] H. K. Rath, B. Panigrahi, and A. Simha, "On cooperative on-path and off-path caching policy for information centric networks (ICN)," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2016, pp. 842–849.

[24] Y. Yang, T. Song, and B. Zhang, "OpenCache: A lightweight regional cache collaboration approach in hierarchical-named ICN," *Comput. Commun.*, vol. 144, pp. 89–99, Aug. 2019.

[25] R. Hou, L. Zhang, T. Wu, T. Mao, and J. Luo, "Bloom-filter-based request node collaboration caching for named data networking," *Cluster Comput.*, vol. 22, no. S3, pp. 6681–6692, May 2019.

[26] B. Azimdoost, G. Farhadi, N. Abani, and A. Ito, "Optimal in-network cache allocation and content placement," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 263–268.

[27] S. Hoteit, M. El Chamie, D. Saucez, and S. Secci, "On fair network cache allocation to content providers," *Comput. Netw.*, vol. 103, pp. 129–142, Jul. 2016.

[28] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 95–107, Jan. 2016.

[29] W. Chu, M. Dehghan, D. Towsley, and Z.-L. Zhang, "On allocating cache resources to content providers," in *Proc. Conf. 3rd ACM Conf. Inf.-Centric Netw.*, 2016, pp. 154–159.

[30] Z. Zhang, H. Ma, Y. Xue, and L. Liu, "Fair video caching for named data networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[31] D. Yuan, Y. Xu, J. Ran, H. Hu, Y. Liu, and X. Li, "An optimal fair resource allocation strategy for a lightweight content-centric networking architecture," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 573–577.

[32] V. S. Mai, S. Ioannidis, D. Pesavento, and L. Benmohamed, "Optimal cache allocation under network-wide capacity constraint," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 816–820.

[33] A. S. Gomes, B. Sousa, D. Palma, V. Fonseca, Z. Zhao, E. Monteiro, T. Braun, P. Simoes, and L. Cordeiro, "Edge caching with mobility prediction in virtualized LTE mobile networks," *Future Gener. Comput. Syst.*, vol. 70, pp. 148–162, May 2017.

[34] K. N. Lal and A. Kumar, "A centrality-measures based caching scheme for content-centric networking (CCN)," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 17625–17642, Jul. 2018.

[35] J. Lv, X. Wang, and M. Huang, "A smart ACO-inspired named data networking forwarding scheme with clustering analysis," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 3, p. e3279, Mar. 2018.

[36] B. Banerjee, A. Kulkarni, and A. Seetharam, "Greedy caching: An optimized content placement strategy for information-centric networks," *Comput. Netw.*, vol. 140, pp. 78–91, Jul. 2018.

[37] P. Mahadevan, P. Sharma, and S. Banerjee, "A power benchmarking framework for network devices," in *Proc. IFIP Int. Conf. Netw.*, 2009, pp. 795–808.

[38] J. Lv, X. Wang, and M. Huang, "ACO-inspired ICN routing mechanism with mobility support," *Appl. Soft Comput.*, vol. 58, pp. 427–440, Sep. 2017.

[39] J. Lv, X. Wang, K. Ren, M. Huang, and K. Li, "ACO-inspired information-centric networking routing mechanism," *Comput. Netw.*, vol. 126, pp. 200–217, Oct. 2017.

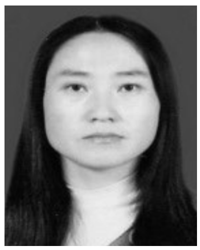
[40] *NS3*. [Online]. Available: https://www.nsnam.org

[41] *YouTube Traces*. [Online]. Available: http://traces.cs.umass.edu/index.php/Network/Network

[42] *GTS-CE*. [Online]. Available: www.topology-zoo.org

[43] *ITC-Deltacom*. [Online]. Available: www.topology-zoo.org

[44] *DPDK*. [Online]. Available: https://www.dpdk.org



YULAN ZHAO received the M.S. degree from Northeast Electric Power University, Jilin, China, in 2009. She is currently an Associate Professor with Jilin Agricultural Science and Technology University. Her research interests include future Internet, pattern recognition, and image fusion.

JIANHUI LV (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from the Jilin Institute of Chemical Technology, Jilin, China, in 2012, and the M.S. and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China, in 2014 and 2018, respectively. He worked with the Network Technology Laboratory, Central Research Institute, Huawei Technologies Company, Ltd., Shenzhen, China, as a Senior Engineer, from January 2018 to July 2019. He is currently an Assistant Research Fellow with the International Graduate School at Shenzhen, Tsinghua University. He has published more than 20 journals (such as *COMNET* (Elsevier), *ASOC* (Elsevier), *JNCA* (Elsevier), and the IEEE COML) and conference papers (such as the IEEE International Conference on Computer Communications (INFOCOM), the IEEE/ACM International Symposium on Quality of Service (IWQoS), and the IEEE International Conference on Parallel and Distributed Systems (ICPADS). His research interests include ICN, in-network caching enabled networks, the IoT, and edge computing. He has served as a Guest Editor (Leader) for *Internet Technology Letters*.