Noname manuscript No.
(will be inserted by the editor)

Deep Learning Based Edge Caching for Multi-cluster Heterogeneous Networks

Jiachen Yang[1] · Chaofan Ma[1] · Bin Jiang[1] · Huihui Wang[2] · Juping Zhang[3] · Gan Zheng[4]

the date of receipt and acceptance should be inserted later

Abstract In this work, we consider a time and space evolution cache refreshing in multicluster heterogeneous networks. We consider a two-step content placement probability optimization. At the initial complete cache refreshing optimization, the joint optimization of the activated base station density and the content placement probability is considered. And we transform this optimization problem into a GP problem. At the following partial cache refreshing optimization, we take the time-space-evolution into consideration and derive a convex optimization problem subjected to the cache capacity constraint and the backhaul limit constraint. We exploit the redundant information in different content popularity using the Deep Neural Network to avoid the repeated calculation because of the change of content popularity distribution at different time slot. Trained DNN can provide online response to content placement in a multi-cluster HetNets model instantaneously. Numerical results demonstrate the great approximation to the optimum and generalization ability.

Keywords DNN · HetNets · joint optimization · user cluster · content placement

1 Introduction

The explosive data growth in online social network has brought us a serious challenge to the design of next generation network architecture [1], [2]. HetNets and edge caching are two key technologies to meet the ever growing wireless data demand by increasing the regional spectral efficiency, decreasing the transmission delay and avoiding the use of the limited backhaul capacity [3]-[7]. The study about the content popularity distribution shows that files requested by the users tend to have a heavy-tailed distribution, i.e., Zipf distribution [8], which means the few very popular files dominate the requests of the users. Inspired by this fact, cache is introduced to reduce the duplicate file transmission. Beside the optimal cache strategy to improve the system performance, we can also apply the BS sleeping technology to improve the overall network energy efficiency [9], [10]. Studies have shown that BSs are largely under-utilized especially at weekends [11].

Inspired by the memory hierarchy in the computer science which introduces the main memory between the CPU and hard disk to reduce the communication delay, we try to realize the cache-enabled base station to bring the file closer to the user requests [12]-[15]. When we bring cache in the base station, we introduce the cache and backhaul resource allocation and scheduling in the wireless network edge. The content placement in cache

Bin Jiang
E-mail: jiangbin@tju.edu.cn

Jiachen Yang
E-mail: yangjiachen@tju.edu.cn

Chaofan Ma
E-mail: machaofan@tju.edu.cn

Huihui Wang
E-mail: hwang1@ju.edu

Juping Zhang
E-mail: jupingnku@gmail.com

Gan Zheng
E-mail: g.zheng@lboro.ac.uk

[1] School of Electrical and Information Engineering, Tianjin University, China · [2] Department of Engineering at Jacksonville University, Jacksonville, FL, 32211, USA · [3] Nankai University, 94 Weijin Rd, Nankai Qu, China, 300071 · [4] Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, UK

refreshing becomes the main concern of our edge cache problem.

Intensive researches about the optimal cache allocation strategies has been conducted. In [13], a simple single tier cache-enabled network is considered to minimize the cache missing probability, and the optimization problem is efficiently solved using dynamic programming. While in [16], [17], channel selection diversity and network interference are taken into consideration, which provides us more specific cache optimization in the wireless caching helper networks. A more complicated heterogeneous network system model can be found in [18], [19], where stochastic geometry is applied to model the HetNets. In [20]-[22], a cache-enabled device-to-device(D2D) communication network is discussed to leverage the spatio-temporal correlation in the different users' data demand. In [23], caching and multicasting are jointly studied over the wireless networks to support massive content delivery, especially, joint consideration of caching and multicasting is extended to a large-scale cache-enabled HetNet with backhaul constraints in [24]. In [25], the previous assumption about the nearest small base station among all that have cached the desired content connecting to the typical user is abandoned, and a user-centric SBS clustering model with two beamforming schemes is studied.

Most of the above works consider the cache optimization as a static optimization problem with a static file library, which usually can not describe the physical world precisely, since the content popularity evolves with time and space. Till now, there exists many works that consider the cache resource allocation as a dynamic optimization problem and apply different kinds of learning methods into this field. In [27], a social-aware networking caching Framework is considered, and the advantages of the big data have been explored to cope with the social dynamics. In [28], a dynamic file library with evolved content popularity and changing files is considered instead of the static file library while preserving the simplicity and computational efficiency of models developed under stationary popularity conditions. In [29], a novel cache replacement method, Trend-Caching, explicitly learns the trend of video content and is more responsive to continuously changing trends of videos. Similar works can still be found in [30].

The main contributions of this paper are summarized as follows:

– We consider the time-space-evolution content popularity instead of the static file library, which makes our system model dynamic.
– Corresponding to the dynamic file library, we adopt the two-step dynamic cache refreshing optimization. The initial complete cache refreshing is the joint optimization of the activated base station density and cache placement probability. The initial cache refreshing optimization provides the initial content placement probability and activated base station densities for the following partial cache refreshing optimization.
– After the initial optimization, we try to conduct partial cache refreshing at each time slot between two adjacent initial optimization using the very limited backhaul.
– We apply the DNN method to learn the map between the input and output for the partial cache refreshing optimization. Simulation results show the great approximation ability.

The remainder of this paper is organized as follows. In section II, we summarized the related works. Section III presents the system model used in this paper. Section IV presents the problem formulation and DNN method. Intensive numerical simulations are presented at section V. Finally, section VI concludes this paper.

2 Related works

Some of the existing works have paid their attention to exploiting the rich contextual information from the device-to-device interaction to learn the content popularity evolution, such as in [31]-[33]. Further more, some of the existing works distinguish content popularity with user preference, and provide a more specific system model by exploiting individual user behavior such as in[34]. In [35], a novel reinforcement learning(RL) framework is put forth for finding the optimal caching policy with unknown popularity profiles, as well as the space-time popularity dynamics of user file requests.

All this existing works about spatio-temporal dynamic content popularity usually consider a relatively easy network topology. The complexity and nonconvexity of the caching optimization problem make it difficult to consider the optimization problem in a multi-cluster heterogeneous network model, which is the main concern in this paper. In [36], [37], Poisson Cluster Process(PCP) is introduced to satisfy the need to consider variety of user and base station(BS) configurations for realistic performance evaluation and system design. This model will help to capture both nonuniformity and coupling in the user and BS locations, which inspires us to consider a PCP-based system model to formulate our optimization problems.

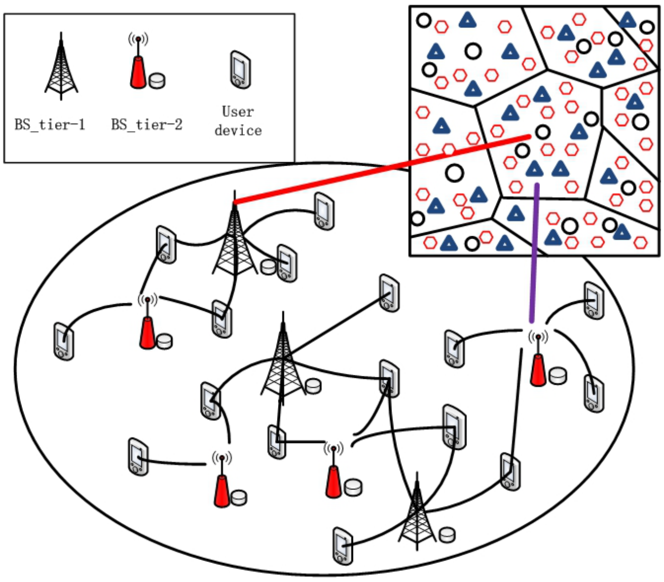


Fig. 1 A multi-cluster $2-tier$ HetNets model, with user and BSs distributed as independent homogeneous PPP in each cluster.

3 System model and problem description

3.1 Multi-cluster heterogeneous network model

We consider a $E-cluster\ K-tier$ HetNets model as in Fig. 1. The heterogeneous network comprsing of K different types of BSs in each cluster is considered in this work with its storage capacity denoted by C_k, and effective radius denoted by r_k, $1 \leq k \leq K$, where the effective radius means that the user can connect to the $k-tier$ BS if and only if the distance between them is less than or equal to the effective radius r_k. And this $K-tier$ HetNets in the cluster e are modeled as K independent homogeneous Poisson Point Process(PPP) with their deployment density denoted as $\lambda_{ke}^{total}, 1 \leq k \leq K, 1 \leq e \leq E$. Again, user distribution on the $2D$ plane is assumed to be homogeneous PPP with its density denoted by $\lambda_{u_e}, 1 \leq e \leq E$.

We assumed that different clusters have different user preference and are independent of each other, while for users in the same cluster, they share the same user preference, hence the same content popularity distribution.

3.2 Base station sleeping

Because of the spatial and temporal evolution of the user density, here we introduce the base station sleeping to adapt to the time-space-varying user requests and avoid the base station resource waste at the idle time and the base station resource deficiency at the rush time.

We consider our BSs running on two modes: activated model and sleep model. The $k-tier$ BSs in the cluster e running on the activated model have the distribution density denoted as λ_{ke}, which is the variable we should optimize. Hence the base station running on the sleeping mode have the distribution density denoted by $\lambda_{ke}^{total} - \lambda_{ke}$. This BS distribution density is closely related to the user density in each cluster. The energy consumption costs of the $k-tier$ BSs running on the activated model and sleep model are denoted as α_k and β_k. We define $t_k = \alpha_k - \beta_k$, for $1 \leq k \leq K$.

3.3 Dynamic file library and cache refreshing model

We assume there is a dynamic file library which contains F files with normalized unit size with its content popularity distribution evolves through time. We denote the $f-th$ most popular file at time slot t in the cluster e as $c_f^e(t)$, and its popularity as $q_f^e(t)$ that follows a general distribution. Files in the library are sorted according to a descending order of popularity and $\sum_{f=1}^{F} q_f^e(t) = 1$.

We adopt a probabilistic caching model. We assume the BSs in the same tier of HetNets share the same content placement probabilities. So the optimal content placement probabilities can be denoted as follows:

$$\mathrm{P}_{K\times F}^{e}(t) = \begin{pmatrix} p_{11}^e(t) & \dots & p_{1F}^e(t) \\ \vdots & \ddots & \vdots \\ p_{K1}^e(t) & \cdots & p_{KF}^e(t) \end{pmatrix}, \tag{1}$$

where $p_{kf}^e(t)$ denotes the content placement probability of file $c_f^e(t)$ in $k-tier$.

3.4 Long term evolution cache model

We divide our optimization process into two continuous step: the initial complete cache refreshing and the following partial cache refreshing.

The initial complete cache refreshing include the joint optimization of the base station density and the cache content placement probability. And this complete cache resource refreshing usually happens at the off-peak time with unlimited backhaul resource. So the complete cache refreshing is available. At the initial optimization process, we are given the initial content popularity $q_f^e(0)$ to optimize the activated base station density λ_{ke} and the initial content placement probability $p_{kf}^e(0)$.

Because of the time-space-evolution of the content popularity in each cluster, the initial content placement probability optimization will soon not satisfy the user requests well as the content popularity changes. To meet this need, we introduce the following partial cache refreshing optimization to provide partial cache refreshing to follow the evolution of the content popularity, and using a very limited backhaul bandwidth D(since we only refresh the partial cache). So when considering the following partial cache refreshing, we are given the current content popularity $q_f^e(t)$ and the last optimal content placement probability $p_{kf}^e(t-1)$, and seek the current content placement probability $p_{kf}^e(t)$. In this stag, we do not optimize the activated base station densities and the activated base station densities are set the same as the initial joint optimization.

Now we can summarize our two-step continuous optimization method as follows: the initial optimization: $\{q_f^e(0)\} \rightarrow \{\lambda_{ke}, p_{kf}^e(0)\}$; the following optimization: $\{q_f^e(t), p_{kf}^e(t-1)\} \rightarrow \{p_{kf}^e(t)\}, \forall t = 1, \cdots, N$, where N denotes the time of partial optimization between two initial complete optimization.

4 Problem development and Deep Neural Network Method

4.1 Problem development using optimization method

We adopt the total missing probability as our system performance metric, which denotes the average probability that a typical user cannot find the requested file in the cache of each BSs within their effective radius. For the ease of analysis, we assume a typical user located at origin o. We define K point sets for tiers in HetNets as $B_k = \{\mathbf{v} | \, \|\mathbf{v}\| \leqslant r_k\}$, $1 \leq k \leq K$, where $\mathbf{v}$ denotes the location of the BS, $\|.\|$ denotes the Euclidean norm. The typical user can get the file cached in the BS located at $\mathbf{v}$ belonging to the $k-tier$ if and only if $\mathbf{v} \subseteqq B_k$.

We first derive the total missing probability in a single cluster e. The distributions of the BSs in the HetNets are assumed to be K independent homogeneous PPP. For a typical file f, $p_{kf}^e(t)$ denotes the content placement probability of this file cached in $k-tier$ in the cluster e at the time slot t. According to the Thin Property [39] of homogeneous PPP, the density of $k-tier$ BSs in the cluster e cached the typical file f is $\lambda_{ke} p_{kf}^e(t)$.

We assume that the typical user can access n_k $k-tier$ BSs cached the typical file f, where n_k satisfies the discrete Poisson distribution with density $\lambda_{ke} p_{kf}^e(t) \pi r_k^2$ as follows:

$$P\{\Phi_k(B_k) = n_k\} = \\ \exp(-\lambda_{ke} p_{kf}^e(t) \pi r_k^2) \frac{(\lambda_{ke} p_{kf}^e(t) \pi r_k^2)^{n_k}}{n_k!}, \tag{2}$$

where Φ_k denotes the homogeneous PPP for $k-tier$. So $n_k = 0$ means there is no such BSs for typical file f at $k-tier$. And $k-tier$ BSs miss the request of typical user for file f with the probability:

$$P\{\Phi_k(B_k) = 0\} = \exp(-\lambda_{ke} p_{kf}^e(t) \pi r_k^2), \tag{3}$$

According to the independence of each tier in HetNets, we derive the missing probability for file f as follows:

$$\begin{aligned} E_f(t) &= \prod\nolimits_{k=1}^{K} P\{\Phi_k(B_k) = 0\} \\ &= \exp(-\sum\nolimits_{k=1}^{K} \lambda_{ke} p_{kf}^e(t) \pi r_k^2). \end{aligned} \tag{4}$$

So the total missing probability for the cluster e at the time slot t equals to

$$\begin{aligned} f^e(t) &= \sum\nolimits_{f=1}^{F} q_f^e(t) E_f(t) \\ &= \sum\nolimits_{f=1}^{F} q_f^e(t) \exp(-\sum\nolimits_{k=1}^{K} \lambda_{ke} p_{kf}^e(t) \pi r_k^2). \end{aligned} \tag{5}$$

4.1.1 The initial complete cache refreshing optimization

According to our assumption that any two adjacent clusters are independent or orthogonal resources are used. For our multi-cluster model, we use the overall total missing probability as our objective function subjected to the cache capacity constraints and energy consumption constraints. We denote the overall total missing probability as the weighted sum of the total missing probability of each cluster and the weights is in proportion to the number of users in each cluster. Now, we can derive our optimization problem as follows:

$$\mathbb{P}_0: \min_{\{p_{kf}^e, \lambda_{ke}\}} \sum_{e=1}^{E} \lambda_{u_e} S_e \frac{f^e(0)}{\sum\limits_{e=1}^{E} \lambda_{u_e} S_e}$$

s.t.

$$\sum\nolimits_{f=1}^{F} p_{kf}^e(0) \leq C_k, \tag{6}$$

$$\begin{aligned} \sum_{e=1}^{E} \sum_{k=1}^{K} t_k \lambda_{ke} \pi r_k^2 &\leq T(\sum_{e=1}^{E} \lambda_{u_e} S_e) \\ &- \sum\nolimits_{k=1}^{K} \sum\nolimits_{e=1}^{E} \beta_k \lambda_{ke}^{total} \pi r_k^2, \end{aligned} \tag{7}$$

$$0 \leq \lambda_{ke} \leq \lambda_{ke}^{total},$$

$$0 \leq p_{kf}^e(0) \leq 1,$$

where (6) denotes the cache capacity constraints, and (7) denotes the energy consumption cost constraints, which comes from

$$\begin{aligned} &\sum\nolimits_{e=1}^{E} \sum\nolimits_{k=1}^{K} \beta_k (\lambda_{ke}^{total} - \lambda_{ke}) \pi r_k^2 \\ &+ \sum\nolimits_{e=1}^{E} \sum\nolimits_{k=1}^{K} \alpha_k \lambda_{ke} \pi r_k^2 \leqslant T(\sum\nolimits_{e=1}^{E} \lambda_{u_e} S_e), \end{aligned} \tag{8}$$

where T denotes the energy consumption efficient. We promise the total energy consumption of both the activated base station and the sleeping base station is less than or equal to the T times of the total average number of the user in all cluster.

Because of the coupling of different clusters in the constraints (7), the optimization problem $\mathbb{P}_0$ is not convex and very difficult to solve. We reformulate the optimization problem $\mathbb{P}_0$ into a geometric program(GP) (9) via the transformations $h_{ke} = \exp(\lambda_{ke})$ and $y^e_{(f-1)\times K+k} = \exp(\lambda_{ke}\pi r_k^2 p^e_{kf}(t))$. Now the GP (9) can be ready to solve using the generic algorithm [40].

$$
\begin{aligned}
&\min \sum\nolimits_{e=1}^{E} \lambda_{u_e} S_e \frac{\sum_{f=1}^{F} q_f \left(\prod_{k=1}^{K} y^e_{(f-1)\times K+k}\right)^{-1}}{\sum_{e=1}^{E} \lambda_{u_e} S_e} \\
&\text{s.t.} \\
&\prod\nolimits_{k=1}^{K} y^e_{(f-1)\times K+k} \le (h_{ke})^{\pi r_k^2 C_k}, \forall e = 1, \cdots, E, \\
&\prod\nolimits_{e=1}^{E} \prod\nolimits_{k=1}^{K} (h_{ke})^{t_k \pi r_k^2} \le \\
&\exp\left[T\left(\sum\nolimits_{e=1}^{E} \lambda_{u_e} S_e\right) - \sum\nolimits_{k=1}^{K} \sum\nolimits_{e=1}^{E} \beta_k \lambda_{ke}^{total} \pi r_k^2\right], \\
&h_{ke} \ge 1, \\
&h_{ke} \le exp(\lambda_{ke}^{total}), \\
&y^e_{(f-1)\times K+k} \le (h_{ke})^{\pi r_k^2}, \\
&y^e_{(f-1)\times K+k} \ge 1.
\end{aligned}
\tag{9}
$$

4.1.2 The following partial cache refreshing optimization

After we accomplish the initial complete joint optimization of the activated base station optimization and initial content placement probability $p^e_{kf}(0)$, we can continue to the following partial cache refreshing optimization using very limited backhaul bandwidth. We still use the same total missing probability as the system performance metric, but the constraints are different.

$$
\begin{aligned}
&\mathbb{P}_1 : \min_{\{p^e_{kf}, \lambda_{ke}\}} \sum_{e=1}^{E} \lambda_{u_e} S_e \frac{f^e(t)}{\sum\limits_{e=1}^{E} \lambda_{u_e} S_e} \\
&\text{s.t.} \\
&\frac{1}{2} \sum\nolimits_{f=1}^{F} \left| p^e_{kf}(t) - p^e_{kf}(t-1) \right| \leqslant D \qquad (10) \\
&\sum\nolimits_{f=1}^{F} p^e_{kf}(t) \le C_k, \\
&0 \le p^e_{kf}(t) \le 1,
\end{aligned}
$$

where the constraint (10) denotes the total cache refreshing of each tier between the t time slot and $t-1$ time slot at each cluster. Because of very limited backhaul bandwidth D is used, we constraint the total cache refreshing less than or equal to the limited backhaul bandwidth D.

The convexity of the problem $\mathbb{P}_1$ is not difficult to derive. The constraint (10) is the absolute function, hence is convex. The rest two constraints are linear with the variable $p^e_{kf}(t)$, hence are convex. The convexity of the objective function is equivalent to the simplified function $g = \exp(-\sum_{l=1}^{L} a_l x_l)$. Since the Hessian matrix of the function g is semi-definite, our original objective function is also convex. So we conclude now the optimization problem $\mathbb{P}_1$ is convex.

4.2 Deep Neural Network method

Due to the quick time and space evolution of the content popularity, we must conduct the partial cache refreshing optimization again and again. And this is not only the waste of the computer resource, but also lead to useless redundant calculation, since the output of our optimization problem $\mathbb{P}_1$ usually shows some patterns. Each time we solve the partial cache refreshing optimization using CVX, we ignore these patterns and just do it from scratch. Besides, a full-trained DNN can provide online output each time as the content popularity evolves. This is very useful for continuous cache refreshing optimization. Deep Neural Network provides us a method to recognize the special patterns. Inspired by this, we consider to use a multilayer perceptron as approximation realization of this continuous mapping from the input space to output space according to Universal Approximation Theorem [38].

We construct our multilayer perceptron architecture for each cluster. This construction helps us ease the complexity of the DNN architecture by reducing the dimension of input and output. Besides, this construction will helps when one cluster's content popularity changes while the rest clusters remain relatively static. At the following partial cache refreshing step, we take the current content popularity and the last content placement probability as input to generate the current content placement probability as output. The input and output of our DNN structure are the same as the optimization.

In our network architecture for each cluster, we adopt 4 hidden layers, one input layer and one output layer with sigmoid function $f(v) = \frac{1}{1+\exp(-v)}$ as activation function to constrain each output node to the range $[0, 1]$.

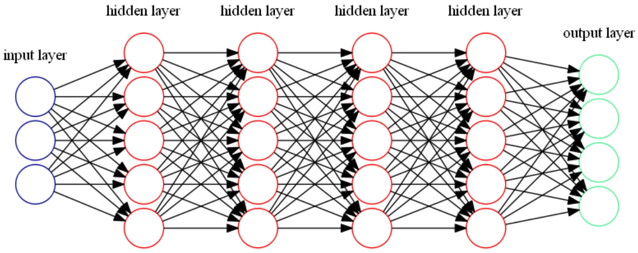


Fig. 2 A full-connected multilayer neural network with 4 hidden layers, 1 input layer and 1 output layer.

In general, the optimization procedure of the DNN is divided into training phase and test phase. During the training phase, we adopt the on-line method to learn the adjustment of the weights of the DNN on an example-by-example basis with each example generated at one specific time slot. This map from the input to the output should have the form: $\{q_f^e(t), p_{kf}^e(t-1)\} \rightarrow \{p_{kf}^e(t)\}$.

In each train step n, we minimize the loss function $\zeta(n)$:

$$\zeta(n) = \frac{1}{2}\sum\nolimits_{k=1}^{K}\sum\nolimits_{f=1}^{F}|p_{kf}(n) - y_{kf}(n)|, \tag{11}$$

where $p_{kf}(n)$ denotes the training sample, and $y_{kf}(n)$ denotes the function signal produced at the output of the neuron at output layer.

After trained with enough training samples, our DNN should acquire the ability to recognize the special content popularity as the input layer and generate the content placement probability as the output in real time. When we test the accuracy of the output of the DNN, it is possible that DNN's output may violate the constraints in optimization problem $\mathbb{P}_1$. So the postprocessing may be needed for the DNN's output:

- for the cache capacity constraint: we first normalized the $f-th$ most popular file's content placement probability $p_{kf}^e(t)$, then multiply the cache capacity $C(k)$, i.e., $p_{kf}^e(t) \triangleq \frac{p_{kf}^e(t)}{\sum_{f=1}^{F} p_{kf}^e(t)} C(k)$;
- for the backhaul limit constraint (10), similar normalized method is applied, i.e., $p_{kf}^e(t) \triangleq \frac{p_{kf}^e(t) - p_{kf}^e(t-1)}{\sum_{f=1}^{F}|p_{kf}^e(t) - p_{kf}^e(t-1)|} D + p_{kf}^e(t-1)$.

5 performance evaluation

5.1 Solving the optimization problem using CVX

We assume the content popularity of our file library satisfies the Zipf distribution, i.e. $q_f = \frac{f^{-\gamma}}{\sum_{f=1}^{F} f^{-\gamma}}$, with Zipf parameter γ randomly generated. Some other system parameters are set as in TABLE 1. We consider a $4-cluster$ $3-tier$ HetNets. We assume that we continue the same two-step optimization day by day, i.e., $M = 110$, which means we need 110 initial complete cache refreshing optimizations. During each day, we conduct $N = 23$ partial cache refreshing optimizations.

We first evaluate the content placement probability evolution from the initial complete cache refreshing optimization to the following partial cache refreshing optimization in Fig. 3. Because of the backhaul limit constraint (10), the optimal partial cache refreshing optimization evolves from the initial complete caches refreshing optimization. In Fig. 3(b), when the Zipf distribution parameter γ evolves from 1.41 to 0.60, i.e., from heavy-tailed distribution to a more uniform distribution, the system tends to cache the most popular files with a reduced probability and increase the cache probability of the less popular files. This optimal partial cache refreshing pattern coincides with our intuition.

5.2 Solving the optimization problem using DNN

We formulate our neural network architecture based on Tensorflow, and the parameter settings for the DNN structure are summarized in TABLE 2.

Table 2 DNN parameters based on Tensorflow

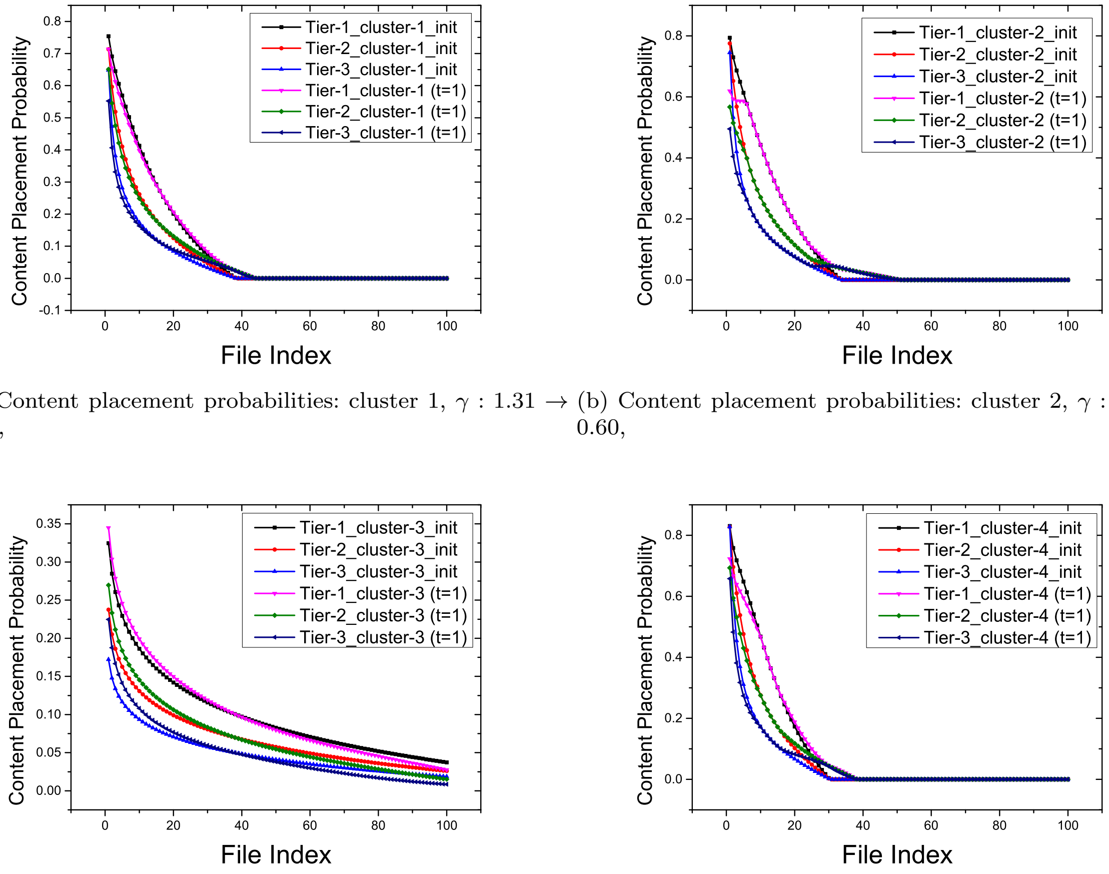
parameters	values
nodes of input layer	500
nodes of hidden layer 1	100
nodes of hidden layer 2	100
nodes of hidden layer 3	100
nodes of hidden layer 4	50
nodes of output layer	400
learning algorithm	AdamOptimizer
learning rate	0.001

5.2.1 Loss function $\zeta(n)$

We first evaluate the loss function $\zeta(n)$, defined in (11), which indicates the statistical average of the error between the neural network's output and the optimal content placement probability. So, $\zeta(n)$ represents the accuracy the of our DNN architecture. As shown in Fig. 5, the loss function $\zeta(n)$ decreases sharply at first 200 generations. Usually, we can promise the final loss function approximately 0.01 for both train samples and test samples. The trend of train loss curve and test loss curve show the high consistency, which shows so designed and trained DNN generalizes well.

Table 1 System parameters

parameters	values
Heterogeneous tiers: K	3
Clusters: L	4
File library: F	100
Cache capacity: $\mathbf{C}$	[10,7,5]
Limited backhaul bandwidth: D	0.5
Zipf exponent	0.5-1.5, random generated
User density $\boldsymbol{\lambda}_u$	[25,28,32,30]
Partial cache refreshing times: N	23
Repeat times: M	110
Effective radius: $\mathbf{r}$ (KM)	[1, 0.8, 0.4]
Cluster radius: s (KM)	[10, 9.6, 8.4, 8.7]
energy consumption cost:$\boldsymbol{\beta}$, $\mathbf{t}$ (KW)	[2, 1.4, 0.8], [10, 7, 5]
energy consumption cost parameter: T	0.006
Total base station deployment density: $\boldsymbol{\lambda}^{\boldsymbol{total}}(/KM^2)$	[1,1.12,1.28,1.6;2,2.24,2.56,2.13;13,11.2,12.8,11.1]



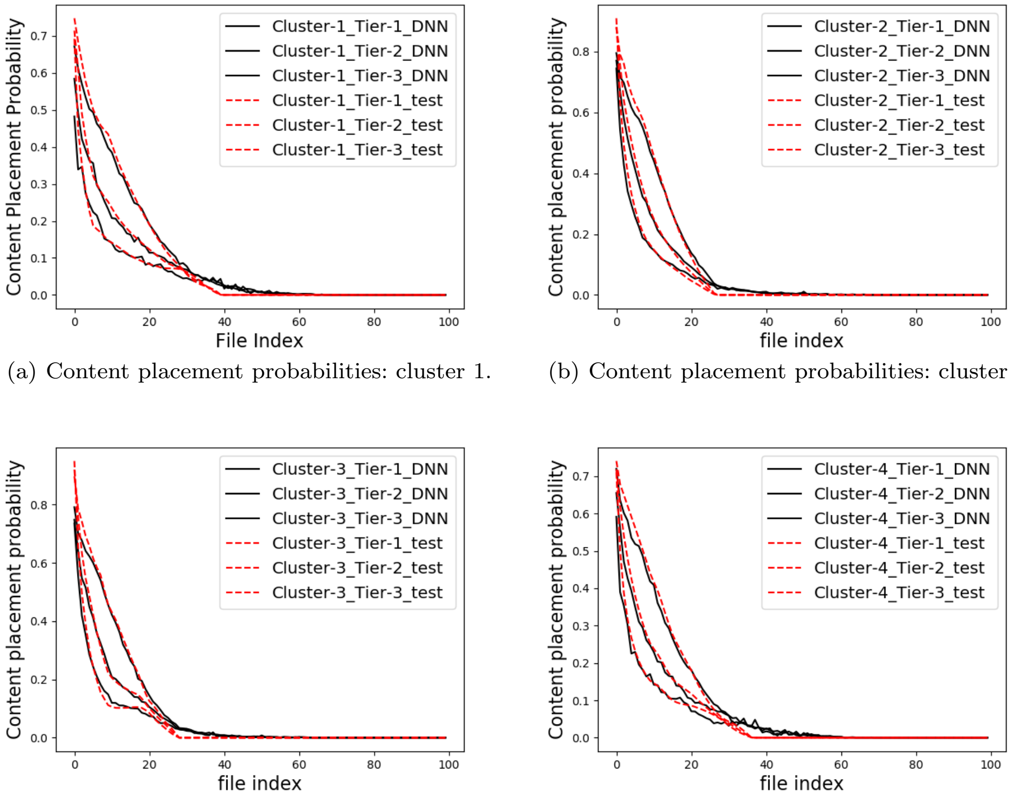
(a) Content placement probabilities: cluster 1, γ : 1.31 → 1.13,

(b) Content placement probabilities: cluster 2, γ : 1.41 → 0.60,

(c) Content placement probabilities: cluster 3, γ : 0.63 → 0.78,

(d) Content placement probabilities: cluster 4, γ : 1.41 → 1.05,

Fig. 3 Comparisons between the initial complete cache refreshing optimization and the following partial cache refreshing optimization at time slot $t = 1$, $\gamma_{init} = [1.31, 1.41, 0.63, 1.41] \rightarrow \gamma(1) = [1.13, 0.60, 0.78, 1.05]$.



(a) Content placement probabilities: cluster 1.

(b) Content placement probabilities: cluster 2.

(c) Content placement probabilities: cluster 3.

(d) Content placement probabilities: cluster 4.

Fig. 4 Space evolution Comparisons between the output of the CVX and DNN.

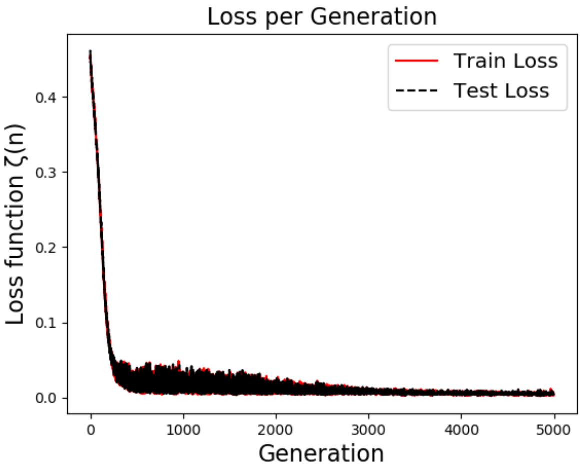


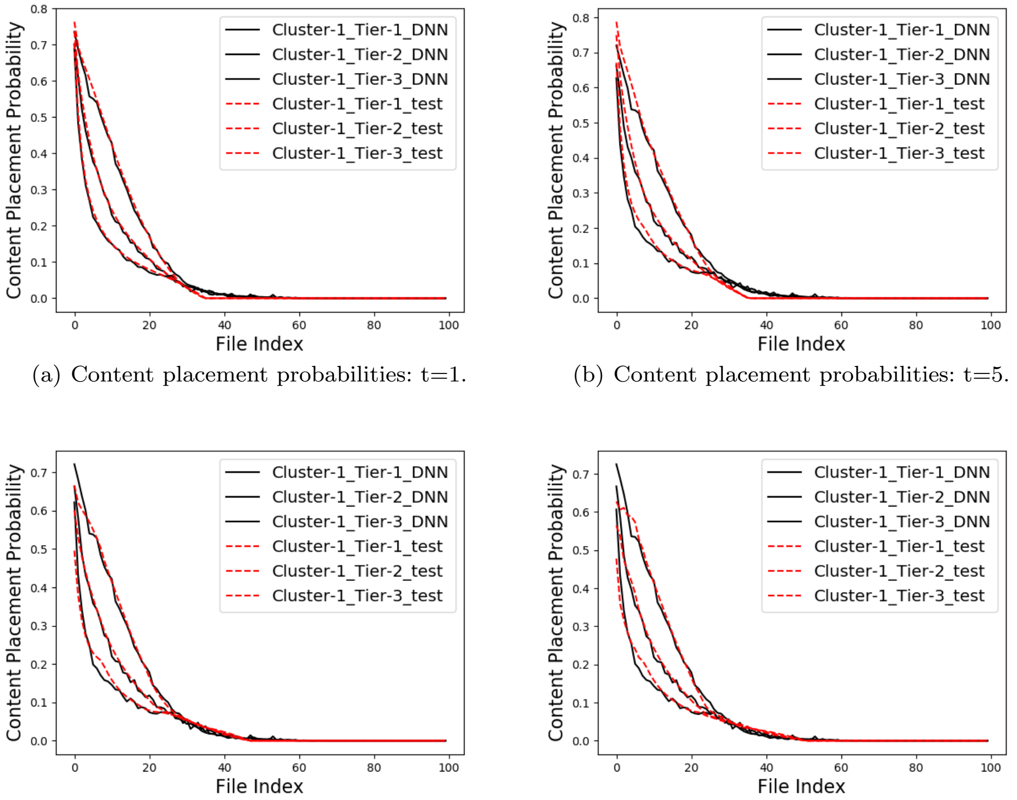
Fig. 5 Loss function $\zeta(n)$ per generation.

5.2.2 Content placement probabilities comparison between the output of the CVX and DNN

In Fig. 4, we evaluate the approximation ability of our DNN structure with the test data generated by the CVX with space-variant content popularity. We consider a space-evolution Zipf distribution with each cluster having its own randomly generated Zipf distribution parameter γ. Simulation results show that our DNN method provide a great approximation ability to test data in each cluster.

In Fig. 6, we evaluate the approximation ability of our DNN structure with the test data generated by the CVX with time-variant content popularity. We consider a time-variant Zipf parameter γ in the $cluster - 1$. We consider the time slot $\mathbf{t} = [1, 5, 9, 13]$. Again, the DNN method provides great approximation ability.

In a word, our DNN method provides not only the space evolution cache refreshing but also the time evolution cache refreshing. Traditional method, such as G-P, usually needs massive computation time, which can not be promised during the partial cache refreshing for the quick content popularity evolution. While for a full-trained DNN, online responding to new cache refreshing is available.



(a) Content placement probabilities: t=1.

(b) Content placement probabilities: t=5.

(c) Content placement probabilities: t=9.

(d) Content placement probabilities: t=13.

Fig. 6 Time evolution comparisons between the output of DNN and CVX, with time slot $\mathbf{t} = [1, 5, 9, 13]$.

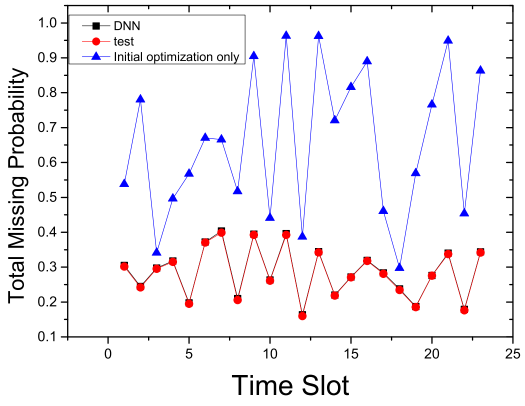


Fig. 7 Total missing probability comparison of DNN and the generated test data at different time slot.

5.2.3 System performance comparison between the CVX and DNN

In Fig. 7, we test the accuracy of the output of the DNN method as the time evolves with content popularity changing all the time. Simulation results show that our DNN method provides an accurate approximation to test data generated by the CVX with only slightly performance degeneration. Besides, for the spatio-temporal evolution content popularity, our following partial optimization provides a far better system performance than the initial optimization only.

6 Conclusion

In this work, we consider a time-space-evolution cache refreshing in multicluster heterogeneous networks. We consider a two-step content placement probability optimization. At the initial complete cache refreshing optimization, the joint optimization of the activated base station density and the content placement probability is considered. And we transform this optimization problem into a GP problem. At the following partial cache refreshing optimization, we take the time-space-evolution into consideration and derive a convex optimization problem subjected to the cache capacity constraint and the backhaul limit constraint. Besides, we construct the DNN structure in each cluster to learn the content placement patterns. A full trained DNN can provide us online respond to each input, which will save the computation resource and satisfy the fast evolution of the content placement probabilities. Simulation results show that our DNN method provides a great approximation ability to the test data.

References

1. Cisco, "Cisco visual networking index: Global mobile data traffic forecast update: 2016–2021 white paper," Feb. 2017.
2. G. Araniti, A. Orsino, L. Militano, "Context-aware Information Diffusion for Alerting Messages in 5G Mobile Social Networks," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 427-436, Apr. 2017.
3. L. Zhou, D. Wu, Z. Dong and X. Li, "When Collaboration Hugs Intelligence: Content Delivery over Ultra-Dense Networks," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 91-95, Dec. 2017.
4. J. An, K. Yang, and J. Wu, "Achieve Sustainable Ultra-Dense Heterogeneous Networks for 5G," *IEEE Commun. Mag.*, vol 55, no. 12, pp. 84-90, 2017.
5. L. Zhou, D. Wu, and Z. Dong, "When Collaboration Hugs Intelligence: Content Delivery over Ultra-Dense Networks" , *IEEE Commun. Mag.*, vol 55, no. 12, pp. 91-95, 2017.
6. X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, Cache in the air: Exploiting content caching and delivery techniques for 5G systems, IEEE Commun. Mag., vol. 52, no. 2, pp. 131C139, Feb. 2014.
7. F. Cicirelli, A. Guerrieri, and G. Spezzano, "Edge Computing and Social Internet of Things for large-scale smart environments development," *IEEE Internet of Things Journal*, Nov. 2017.
8. A. Tatar, M. D. D. Amorim, and S. Fdida, A survey on predicting the popularity of web content, *Journal of Internet Services and Applications*, vol. 5, no. 1, pp. 8-28, 2014
9. Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, Probabilistic small-cell caching: Performance analysis and optimization, IEEE Trans. Veh. Technol., vol. 66, no. 5, pp. 4341C4354, May. 2017.
10. D. Zhai, R. Zhang, L. Cai, B. Li, Y. Jiang, "Energy-Efficient User Scheduling and Power Allocation for NOMA based Wireless Networks with Massive IoT Devices," IEEE Internet of Things Journal, accepted, Mar. 2018.
11. E. Oh, B. Krishnamachari, and X. Liu, Toward dynamic energy-efficient operation of cellular network infrastructure, *IEEE Communications Magazine*, vol. 49, no. 6, pp. 56-61, 2011.
12. R. E. Bryant and D. R. O'Hallaron, Computer Systems: A Programmer's Perspective, Carnegie Mellon University, 2015.
13. K. Avrachenkov, X. Bai, and J. Goseling, "Optimization of caching devices with geometric constraints," *Performance Evaluation*, vol. 113, pp. 68-82, Aug. 2017.
14. C. A. Weng, and K. Psounis, "Distributed Caching and Small Cell Cooperation for Fast Content Delivery," *The, ACM International Symposium. ACM*, pp. 127-136, 2015
15. H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305-1314, 2015.
16. S. H.Chae, C. Wan, "Caching Placement in Stochastic Wireless Caching Helper Networks: Channel Selection Diversity via Caching," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6626-6637, 2016.
17. C. Yang, Y. Yao, B. Xia, K. Huang, W. Xie, and Y. Zhao, "Interference Cancellation at Receivers in Cache-Enabled Wireless Networks", IEEE Transactions on Vehicular Technology, vol. PP, no. 99, 2017.
18. C. Yang, Y. Yao, and Z. Chen, "Analysis on Cache-Enabled Wireless Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 131-145, 2016.
19. B. Serbetci, and J. Goseling, "On Optimal Geographical Caching in Heterogeneous Cellular Networks," *Wireless Communications and NETWORKING Conference, IEEE*, San Francisco, CA, USA, Mar. 2017.
20. S. Krishnan, and H. S. Dhillon, "Distributed caching in device-to-device networks: A stochastic geometry perspective," *Signals, Systems and Computers, 2015, Asilomar Conference on. IEEE*, pp. 1280-1284, 2016.
21. Y. Wang, X. Tao, and X. Zhang," Cooperative Caching Placement in Cache-Enabled D2D Underlaid Cellular Network," *IEEE Communications Letters*, vol. 5, no. 5, pp. 1151-1154, 2017.
22. J. Rao, H. Feng, C. Yang, Z. Chen, B. Xia, Optimal Caching Placement for D2D Assisted Wireless Caching Networks, IEEE International Conference on Communications (ICC), 2016.
23. Y. Cui, D. Jiang, and Y. Wu, Analysis and optimization of caching and multicasting in large-scale cache-enabled wireless networks, IEEE Trans. Wireless Commun., vol. 15, no. 7, pp. 5101-5112, Jul. 2016.
24. Y. Cui and D. Jiang, Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks, IEEE Trans. Wire-

less Commun., vol. 16, no. 1, pp. 250-264, Jan. 2017.

25. X. Xu and M. Tao, Analysis and optimization of probabilistic caching in multi-antenna small-cell networks. [Online]. Available: https://arxiv.org/pdf/1709.00664.pdf
26. Wen J., K. Huang, and S. Yang, et al, Cache-Enabled Heterogeneous Cellular Networks: Optimal Tier-Level Content Placement, IEEE Trans. Wireless Commun., vol. 16, no. 9, Sept. 2017, pp. 5939-5952.
27. K. Machado, A. Boukerche, and E. Cerqueira, et al, A Socially-Aware In-Network Caching Framework for the Next Generation of Wireless Networks, IEEE Communication Magazine, vol. 55, no. 12, pp. 38-43, 2017.
28. M.Garetto, E. Leonardi, and S. Traverso, Efficient analysis of caching strategies under dynamic content popularity, Computer Communications (INFOCOM), 2015 IEEE Conference on, Aug. 2015, pp. 2263-2271.
29. S. Li, J. Xu, and M. V. D. Schaar, "Trend-Aware Video Caching Through Online Learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503-2516, 2016.
30. M. Leconte, G. Paschos, L. Gkatzikis, et al, "Placing Dynamic Content in Caches with Small Population," *IEEE INFOCOM 2016*, pp. 1-9, 2016.
31. E. Bastug, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, Mumbai, India, May 2015.
32. A. Sengupta, S. D. Amuru, and R. Tandon, "Learning distributed caching strategies in small cell networks," *International Symposium on Wireless Communications Systems, IEEE*, pp. 917-921, 2014.
33. L. Zhou, D. Wu, and J. Chen, "When Computation Hugs Intelligence: Content-Aware Data Processing for Industrial IoT", *IEEE Internet of Things Journal*, Dec. 2017.
34. B. Chen, C. Yang, "Caching Policy for Cache-enabled D2D Communications by Learning User Preference," *Information Theory*, pp. 1-32, Jan. 2018.
35. A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-time Popularities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 180-190, Dec. 2017.
36. C. Saha, M. Afshang, and H. S. Dhillon, "Poisson cluster process: Bridging the gap between PPP and 3GPP HetNet models" *Information Theory and Applications Workshop(ITA)*, San Diego, CA, USA, Feb. 2017.
37. Wen J., K. Huang, and S. Yang, et al, Cache-Enabled Heterogeneous Cellular Networks: Optimal Tier-Level Content Placement, IEEE Trans. Wireless Commun., vol. 16, no. 9, Sept. 2017, pp. 5939-5952.
38. S. Haykin, Neural Networks and Learning Machines, Prentice Hall, Jan. 2008.
39. B. Francois and B. Blaszczyszyn, Stochastic Geometry and Wireless Networks: Volume I Theory, Now Publishers Inc., Hanover, MA, USA, Mar. 2009.
40. S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge, U.K.: Cambridge University Press, 2004.