Cooperative Edge Caching in Fog Radio Access Networks: A Pigeon Inspired Optimization Approach

Chengyu Xia[1,2,3], Yanxiang Jiang[1,2,3,*], Mugen Peng[4], Fu-Chun Zheng[1,5], Mehdi Bennis[6], and Xiaohu You[1]
[1]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China.
[2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[3]Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, 865 Changning Road, Shanghai 200050, China
[4]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[5]School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China
[6]Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland
E-mail: {cyxia, yxjiang, fzheng, xhyu}@seu.edu.cn; pmg@bupt.edu.cn; mehdi.bennis@oulu.fi

***Abstract*—In this paper, the cooperative edge caching problem in fog radio access networks (F-RANs) is investigated to minimize the average download delay. Considering the non-linear and coupled multi-variable nature of the original optimizing problem, we transform it into an equivalent integer linear programming problem with decoupled variables. Then, we decomposed the transformed problem into two subproblems which can be solved separately by each fog access point (F-AP). Considering the non-deterministic polynomial hard (NP-hard) nature of the two decomposed subproblems, we propose an improved pigeon inspired optimization (PIO) based cooperative edge caching scheme, which utilizes Cauchy perturbation and self-adaptive factor to avoid pre-mature convergence and achieve a better search performance, respectively. Our proposed scheme not only allows F-APs to make cache decisions with low computational complexity, but also has very low message passing overhead. Simulation results show that our proposed scheme can greatly decrease the average download delay.**

***Index Terms*—Fog radio access networks, cooperative edge caching, pigeon inspired optimization, average download delay.**

I. Introduction

With the explosive and continuous growth of various multimedia service, a huge number of user equipments (UEs) are expected to be connected to wireless networks in the coming future. This has triggered increasing pressure on capacity-limited backhaul links in wireless networks [1]. In order to deal with this challenge, fog radio access network (F-RAN) has emerged as a promising architecture for future wireless networks. To decrease the average download delay, fog access points (F-APs) with limited cache capacity can cache popular contents so that UEs can download popular contents from network edge [2]–[5]. Moreover, F-APs can conduct cooperative edge caching to further decrease the average download delay.

Recently, there have been many researches concerning cooperative edge caching. In [6], a centralized cache placement algorithm was proposed for small cell networks to minimize the average download delay, in which a central coordinator solves the cache optimizing problem and makes cache decisions for each base station (BS). In [7], a distributed algorithm based on belief propagation (BP) was proposed to minimize the average download delay, which allows each BS to make its caching decision via local information. In [8], the authors proposed an efficient cooperative caching scheme in hierarchical networks by considering the cooperation between radio remote heads and base band units. Another cooperative caching scheme was proposed in [9] for device to device integrated cellular networks by considering the cooperation between device and BS. However, the aforementioned caching schemes might not be as efficient in F-RANs since F-APs have relatively small cache capacity [10] and the number of UEs associated with each F-AP is often too small to reveal the content aggregation effect [11].

Motivated by the aforementioned discussions, we propose an improved pigeon inspired optimization (PIO) based cooperative edge caching scheme in F-RANs. Our proposed scheme allows F-APs to make their cache decisions based on local information. Cauchy perturbation and self-adaptive factor are adopted to avoid pre-mature convergence and achieve a better global and local search ability. Our proposed scheme can achieve a low average download delay and offer a low computational complexity simultaneously. Moreover, it has lighter message passing overhead than the baseline. To the best of our knowledge, no existing research has applied PIO in solving the edge caching problem in wireless networks.

The rest of this paper is organized as follows. In Section II, the system model is briefly described. In Section III, the cooperative edge caching problem is formulated. The proposed PIO based cooperative edge caching scheme is presented in Section IV. Simulation results are shown in Section V. Final conclusions are drawn in Section VI.

II. System Model

An illustration of F-RANs is shown in Fig. 1. For the sake of simplicity, it is assumed that each UE can only be served by one F-AP (serving F-AP) at the same time and F-APs from different clusters cannot cooperate with each other. Therefore, we only have to investigate the issues in one cluster. Assume

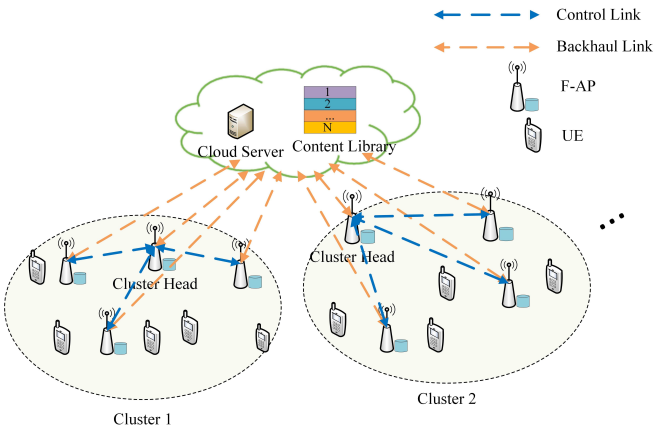

Fig. 1. Illustration of F-RANs.

there are M F-APs and K UEs in the considered cluster. Let $\mathcal{A} = \{a_1, a_2, a_3, \ldots, a_m, \ldots, a_M\}$ denote the F-AP set and $\mathcal{U} = \{u_1, u_2, u_3, \ldots, u_k, \ldots, u_K\}$ denote the UE set. Let $\boldsymbol{L}$ denote an $M\times K$ binary matrix and l_{mk} denote the element of the mth row and the kth column in $\boldsymbol{L}$, which indicates whether UE u_k is associated with F-AP a_m or not. That is, $l_{mk} = 1$ means UE u_k is associated with F-AP a_m and $l_{mk} = 0$ otherwise. Thus, the group of UEs associated with F-AP a_m can be denoted as $\mathcal{U}_m = \{u_k \in \mathcal{U} | l_{mk} = 1\}$.

Let Q denote the cache capacity of each F-AP. Let $\mathcal{C} = \{c_1, c_2, c_3, \ldots, c_n, \ldots, c_N\}$ denote the content library located in the cloud server. Without loss of generality, we assume that all the contents have the same size as in [6], i.e., $|c_n| = |c|$, $\forall c_n \in \mathcal{C}$. UE u_k will request content c_n from its associated F-AP with a probability of p_{nk}. Let $\boldsymbol{X}$ denote an $M\times N$ cache placement matrix and x_{mn} denote the element of the mth row and the nth column in $\boldsymbol{X}$, which indicates whether content c_n is cached by F-AP a_m or not. That is, $x_{mn} = 1$ means content c_n is cached by F-AP a_m and $x_{mn} = 0$ otherwise. Therefore, the cache placement matrix $\boldsymbol{X}$ indicates the caching strategy of the considered F-AP cluster. The objective of this paper is to find the optimal cache placement matrix to minimize the average download delay.

III. Problem Formulation

In this Section, we firstly formulate the cooperative edge caching problem in F-RANs and then transform it into an integer linear programming problem with decoupled variables. Finally, we decompose the integer linear programming problem into two subproblems which can be solved by each F-AP separately.

A. Problem Formulation

If the requested content is cached in the serving F-AP, the requesting UE can download it directly. Let t_{nm} denote the local request count, i.e., the number of times content c_n is requested by the UEs associated with F-AP a_m. Let q_{mn} denote the normalized local content popularity of c_n at F-AP a_m. Then, we have:

$$q_{mn} = \frac{t_{nm}}{\sum_{n=1}^{N} t_{nm}}. \tag{1}$$

Let $\bar{D}_1$ denote the average download delay from a serving F-AP to the corresponding requesting UE. Then, it can be expressed as follows:

$$\bar{D}_1 = \frac{1}{K} \sum_{n=1}^{N} \sum_{m=1}^{M} x_{mn} q_{mn} d_{FU}. \tag{2}$$

where d_{FU} denotes the download delay from a serving F-AP to the requesting UE when the requested content is cached in the serving F-AP.

If the requested content is not cached in the serving F-AP, it will transmit the request to the cluster head, which maintains a cache list for each F-AP in this cluster. If the requested content has been cached by another F-AP in the same cluster, the cluster head will identify it and coordinate the cooperation between two F-APs. It should be emphasized that the job of the cluster head is only to identify the F-AP which stores the requested content and conduct the cooperation with simple control signals but is not responsible for making caching decision for each F-AP. When the requested content c_n is disseminated from F-AP a_l to F-AP a_m, its local request count t_{nl} will also be transmitted to F-AP a_m. Then, F-AP a_m will update its local request count t_{nm} as follows:

$$t_{nm} = \delta t_{nl} + (1-\delta) t_{nm}, \quad \delta \in [0, 1], \tag{3}$$

where δ is a parameter used to indicate the influence of the estimated content popularity from other F-APs on that of the local F-AP. With updated t_{nm}, F-AP a_m can obtain the updated q_{mn} according to (3). Let $\{y_{lm}^n\} \in \{0, 1\}$ denote whether content c_n is disseminated from F-AP a_l to F-AP a_m or not. Here, we set $y_{mm}^n = 0$. Let $\bar{D}_2$ denote the average download delay via F-AP cooperation. Then, it can be expressed as follows:

$$\bar{D}_2 = \frac{1}{K} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{l=1}^{M} y_{lm}^n q_{mn} (d_{FF} + d_{FU}), \tag{4}$$

where d_{FF} denotes the transmit delay between the two cooperative F-APs.

If the requested content is not cached in the cluster, it has to be fetched from the cloud server. Let $\bar{D}_3$ denote the average download delay from the cloud server. Then, it can be expressed as follows:

$$\bar{D}_3 = \frac{1}{K} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{l=1}^{M} (1 - y_{lm}^n)(1 - x_{mn}) q_{mn} (d_{CF} + d_{FU}), \tag{5}$$

where d_{CF} denotes the transmit delay from the cloud server to the serving F-AP. Since the channel condition between the cloud server and F-APs are generally consistent, d_{FF} and d_{CF} can be considered as functions of content size $|c|$ and can be

$$\min_{x_{mn},y_{lm}^n}\left\{\bar{D}_1+\bar{D}_2+\bar{D}_3\right\}=\min_{x_{mn},y_{lm}^n}\{\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}x_{mn}q_{mn}d_{FU}+\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M}y_{lm}^n q_{mn}\left(d_{FF}+d_{FU}\right)$$
$$+\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M}\left(1-y_{lm}^n\right)\left(1-x_{mn}\right)q_{mn}\left(d_{CF}+d_{FU}\right)\},\tag{6}$$

$$\text{s.t.}\quad \sum_{n=1}^{N}x_{mn}\le Q,\tag{6a}$$
$$y_{lm}^n\le x_{ln}\vee x_{mn}-x_{mn},\tag{6b}$$
$$\sum_{l=1}^{M}y_{lm}^n\le\bigcup_{l=1}^{M}x_{ln}-x_{mn},\tag{6c}$$
$$x_{mn},y_{lm}^n\in\{0,1\}.\tag{6d}$$

easily obtained in practice.

With $\bar{D}_1$, $\bar{D}_2$ and $\bar{D}_3$, the cooperative edge caching problem of the considered cluster can be formulated in (6) as shown at the top of this page, where $\vee$ denotes logic calculation 'or', $\bigcup_{l=1}^{M}x_{ln}=x_{1n}\oplus x_{2n}\oplus\cdots\oplus x_{Mn}\in\{0,1\}$, and $\oplus$ denotes logic calculation 'exclusive or'. (6a) denotes the constraint of cache capacity, (6b) denotes the constraint that cooperation will not happen if the serving F-AP has cached the requested content, (6c) guarantees that there will only be one F-AP pairing with the serving F-AP during each cooperation, and (6d) constrains that x_{mn} and y_{lm}^n are all binary variables.

B. Problem Transformation

The problem in (6) requires great computational burden because it is a multi-variable optimizing problem with coupled variables x_{mn} and y_{lm}^n, and involves non-linear constraints in (6b) and (6c). Therefore, we propose to transform the problem in (6) to an integer linear programming problem and decouple the variables for higher computational efficiency in this subsection.

Theorem 1: The constraints in (6b) and (6c) are equivalent to the following two linear constraints in (7b) and (7c).

$$y_{lm}^n\le x_{ln},\quad\forall m,l\in\{1,2,3,\ldots,M\},\forall n\in\{1,2,3,\ldots,N\}.\tag{7b}$$

$$\sum_{l=1}^{M}y_{lm}^n\le 1-x_{mn},\quad\forall m\in\{1,2,3,\ldots,M\},\forall n\in\{1,2,3,\ldots,N\}.\tag{7c}$$

Proof: To prove (6b) and (6c) are equivalent to (7b) and (7c), we only need to prove (7b) and (7c) can be derived from (6b) and (6c) and vice versa. If $x_{mn}=1$ or $x_{mn}=0$ and $x_{ln}=0$, we can get $y_{lm}^n=0$ from (6b), thus (7b) and (7c) hold. If $x_{mn}=0$ and $x_{ln}=1$, we can get $y_{lm}^n\le 1$ from (6b) and (6c), thus (7b) and (7c) also hold.

Similarly, if $x_{mn}=1$ or $x_{ln}=0$, we can get $y_{lm}^n=0$ from (7b) and (7c), thus (6b) and (6c) hold. Otherwise, if $x_{mn}=0$ and $x_{ln}=1$, we can get $y_{lm}^n\le 1$ from (7b) and (7c), thus (6b) and (6c) also hold. This completes the proof. ■

According to *Theorem 1*, the problem in (6) can be transformed into an integer linear programming problem. However, it is still difficult to find the optimal solution for this problem since the variables x_{mn} and y_{lm}^n are coupled.

Theorem 2: The average download delay from the cloud sever to the requesting UE in (6), i.e., the third item in (6), $\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M}\left(1-y_{lm}^n\right)\left(1-x_{mn}\right)q_{mn}\left(d_{CF}+d_{FU}\right)$, is equivalent to $\frac{1}{K}\left(1-\bigcup_{m=1}^{M}x_{mn}\right)\sum_{n=1}^{N}\sum_{m=1}^{M}q_{mn}\left(d_{CF}+d_{FU}\right)$, where $\bigcup_{m=1}^{M}x_{mn}$ denotes whether content c_n is cached in the considered cluster or not.

Proof: $\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M}\left(1-y_{lm}^n\right)\left(1-x_{mn}\right)q_{mn}\left(d_{CF}+d_{FU}\right)\neq 0$ only when $y_{lm}^n=0$ and $x_{mn}=0$ for all l, m, n, which exactly means that content c_n is not cached in the considered cluster. That is, $\left(1-y_{lm}^n\right)\left(1-x_{mn}\right)$ is equivalent to $\left(1-\bigcup_{m=1}^{M}x_{mn}\right)$. This completes the proof. ■

According to *Theorem 1* and *Theorem 2*, the problem in (6) can be transformed into the integer linear programming problem in (7) with decoupled variables as shown at the top of the next page.

C. Problem Decomposition

With variables decoupled, the optimizing problem in (7) can be further decomposed into the following two subproblems.

(1) *Subproblem 1*: Optimize x_{mn}, which can be formulated as follows:

$$\min_{x_{mn}}\left\{\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}x_{mn}q_{mn}d_{FU}+\frac{1}{K}\left(1-\bigcup_{m=1}^{M}x_{mn}\right)\sum_{n=1}^{N}\sum_{m=1}^{M}q_{mn}\left(d_{CF}+d_{FU}\right)\right\}.\tag{8}$$
$$\text{s.t. } (7a),\ (7d)$$

$$\min_{x_{mn},y_{lmn}^n} \{\frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M} x_{mn}q_{mn}d_{FU} + \frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M} y_{lm}^n q_{mn}\left(d_{FF}+d_{FU}\right) + \frac{1}{K}\left(1-\bigcup_{m=1}^{M} x_{mn}\right)\sum_{n=1}^{N}\sum_{m=1}^{M} q_{mn}\left(d_{CF}+d_{FU}\right)\}, \tag{7}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} x_{mn} \le Q, \tag{7a}$$

$$y_{lm}^n \le x_{ln}, \tag{7b}$$

$$\sum_{l=1}^{M} y_{lm}^n \le 1 - x_{mn}, \tag{7c}$$

$$x_{mn}, y_{lm}^n \in \{0,1\}. \tag{7d}$$

(2) *Subproblem 2*: Optimize y_{lm}^n, which can be formulated as follows:

$$\min_{y_{lm}^n} \left\{ \frac{1}{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{M} y_{lm}^n q_{mn}\left(d_{FF}+d_{FU}\right) \right\}. \tag{9}$$
$$\text{s.t. } (7b),\ (7c),\ (7d)$$

Subproblem 1 is about the caching decision of each F-AP. *Subproblem 2* is about the cooperation decision of the F-APs which can be solved at the cluster head.

IV. Proposed PIO Based Cooperative Edge Caching

The subproblems in (8) and (9) are non-deterministic polynomial hard (NP-hard), which require exponential computational complexity if traditional greedy search algorithm is adopted. On the other side, swarm intelligence algorithms do not depend on the analyticity of the optimization problem and are suitable for dealing with complex and large-scale problems which are difficult to solve by traditional approaches. As a novel swarm intelligence algorithm, PIO is inspired by the homing behavior of pigeons. In [12] and [13], it was shown that PIO has lower computational complexity than other baseline swarm intelligence algorithms and has stronger robustness since the optimizing process is not affected by changes in one or several individuals. Therefore, we propose an improved PIO based cooperative edge caching scheme, which can obtain the globally optimal solution yet with relatively low computational complexity.

In PIO, there are two operators: the map and compass operator and the landmark operator, which take the role of the magnetic field, sunlight and landmark in real world. At the beginning, a group of pigeons will be initialized for the optimizing target, with the position of each pigeon as a potential solution to the optimizing problem. The position and velocity of each pigeon will be initialized randomly. The maximum iteration numbers for the two operators will be set respectively. Then, the pigeons will firstly update their positions and velocities in the map and compass operator. When reaching the maximum iteration number, the pigeons will jump into the landmark operator. In each iteration, the global best pigeon is obtained by the fitness function. If the position of the global best pigeon converges or the iteration number reaches the maximum value, the algorithm will output the position of the current global best pigeon as the final result.

To solve the subproblems in (8) and (9) for each F-AP, a group of pigeons will be initialized to be its caching status vectors, and the position of each pigeon denotes a potential cache placement. The fitness functions are the objective functions in (8) and (9), respectively. The position of the global best pigeon is output as the optimal cache placement. The details of the proposed improved PIO based scheme are presented below.

A. Map and Compass Operator

Let $\boldsymbol{x}_i = [X_{i1}, X_{i2}, \ldots, X_{iq}, \ldots, X_{iQ}]^T$ denote the position vector of the ith pigeon, where Q denotes the dimension of the solution space, and X_{iq} can be set by the index of the possible cached content. Let $\boldsymbol{v}_i = [V_{i1}, V_{i2}, \ldots, V_{iq}, \ldots, V_{iQ}]^T$ denote the velocity vector of the ith pigeon. Then, $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ can be initialized randomly and updated as follows:

$$\begin{cases} \boldsymbol{v}_i^{(t)} = \boldsymbol{v}_i^{(t-1)} \cdot e^{-\vartheta t} + \xi \cdot \left(\boldsymbol{x}_g - \boldsymbol{x}_i^{(t-1)}\right), \\ \boldsymbol{x}_i^{(t)} = \boldsymbol{x}_i^{(t-1)} + \boldsymbol{v}_i^{(t-1)}. \end{cases} \tag{10}$$

where ϑ is a factor used to control the search speed, ξ is a random number between 0 and 1, and $\boldsymbol{x}_g$ is the position of the global best pigeon in the $t-1$th iteration.

B. Landmark Operator

In the tth iteration, each pigeon will fly to the Euclidean center of all pigeons, which is assumed to be the final destination. Therefore, the position of each pigeon can be updated as follows:

$$\boldsymbol{x}_i^{(t)} = \boldsymbol{x}_i^{(t-1)} + \xi \cdot \left(\boldsymbol{x}_c^{(t)} - \boldsymbol{x}_i^{(t-1)}\right), \tag{11}$$

where $\boldsymbol{x}_c^{(t)}$ is the Euclidean center of the pigeons in the tth iteration. Let $N_p^{(t)}$ denote the generation size in the tth iteration, $\boldsymbol{x}_c^{(t)}$ can be obtained as follows:

$$\boldsymbol{x}_c^{(t)} = \frac{\sum_{i=1}^{N_p^{(t)}} \boldsymbol{x}_i^{(t)} \cdot \text{ fitness } \left(\boldsymbol{x}_i^{(t)}\right)}{\sum_{i=1}^{N_p^{(t)}} \text{fitness } \left(\boldsymbol{x}_i^{(t)}\right)}. \tag{12}$$

In the landmark operator, the number of pigeons will be halved in each iteration and only the half closer to $\boldsymbol{x}_c^{(t)}$ will remain. This is to simulate the fact that pigeons far away from the destination will follow other pigeons, thus they will lose their searching ability. The number of pigeons in the tth iteration can be expressed as follows:

$$N_p^{(t)} = \text{ceil}\left(\frac{N_p^{(t-1)}}{2}\right). \tag{13}$$

C. Improved PIO

The original PIO suffers from pre-mature convergence. Moreover, the map factor ϑ is a constant value, which makes it difficult to achieve a balance between global search and local search. By considering the aforementioned disadvantages, an improved PIO is proposed by using Cauchy perturbation and a self-adaptive ϑ.

In the map and compass operator of PIO, the algorithm is highly likely to fall into pre-mature convergence if $\boldsymbol{x}_g$ has not changed for multiple iterations. To avoid this, we propose to add a Cauchy perturbation to $\boldsymbol{x}_g$. According to [14], Cauchy distribution can describe many non-linear phenomena better than uniform distribution and Gaussian distribution. We firstly define a factor ω. If $\boldsymbol{x}_g$ has not changed in recent ω iterations, a Cauchy perturbation will be added to $\boldsymbol{x}_g$ as follows:

$$\boldsymbol{x}_g' = \boldsymbol{x}_g + C(x_0, \gamma) \cdot (u_b - l_b), \tag{14}$$

where u_b and l_b are the upper and lower bounds of the solution space and $C\left(x_0, \gamma\right)$ is a random number generated by Cauchy probability density function as follows:

$$f\left(x; x_0, \gamma\right) = \frac{1}{\pi}\left[\frac{\gamma}{\left(x - x_0\right)^2 + \gamma^2}\right], \tag{15}$$

where x_0 and γ respectively denote the location parameter and scale parameter in the Cauchy distribution. If $\boldsymbol{x}_g'$ is within the boundary of the solution space, then $\boldsymbol{x}_g = \boldsymbol{x}_g'$. Otherwise, $\boldsymbol{x}_g$ will not change. If the perturbated $\boldsymbol{x}_g$ is better than the original one, the pigeons will be reinitialized and converge again to jump out of the local optimum.

Self-adaptive factor is adopted in our modified PIO to achieve a better global and local search performance. When ϑ is small, each pigeon has a large speed, which is beneficial for global search. When ϑ is large, each pigeon has a small speed, which is beneficial for local search. So the ideal pattern for ϑ is to be smaller in the early iterations for a better global search and larger in the latter iterations for a better local search. According to this requirement, we propose to set ϑ to a self-adaptive pattern as follows:

$$\begin{cases} \vartheta = \frac{1}{1+\alpha \cdot e^{-x}}, \\ x = -10 + \frac{20t}{N_{c1}+N_{c2}}, \end{cases} \tag{16}$$

where t denotes the current iteration number, α decides the minimum value of ϑ, N_{c1} and N_{c2} denote the maximum iteration numbers in the map and compass operator and the landmark operator, respectively.

D. Computational Complexity

Let N_p denote the initial generation size. The computational complexity of the proposed scheme can be calculated to be $O(N_m N_p Q)$. In comparison, the computational complexity of the traditional greedy search algorithm is $O(NM^2)$. Generally, we have $N_m N_p Q \ll NM^2$. Therefore, the proposed scheme has a much lower computational complexity than the traditional greedy search algorithm.

V. Simulation Results

We conduct simulations to evaluate the performance of the proposed scheme. In our simulations, each UE requests content from its associated F-AP at a random rate. Each F-AP collects requests for 10 hours so that enough requests can be collected to estimate content popularity. As in [8], we assume that there is no inter-cluster interference by adopting proper scheduling policies. Without loss of generality, the content size $|c|$ is set to 100 Mbits. We set $M = 3, K = 50, N = 100$, $d_{\text{CF}} = 40$ ms, $d_{\text{FF}} = d_{\text{FU}} = 5$ ms and $N_p = 300$. Both cooperative and non-cooperative strategies are studied for our proposed scheme and the baselines. Message passing overhead is measured by all information transmitted in the networks, including control signals, contents and other messages.

In Fig. 2, we show the average download delay of the proposed scheme with different cache capacities. Also included are the centralized greedy algorithm which can find the optimal solution and the BP algorithm [7] as the baselines for performance comparison. It can be observed that the average download delay decreases with the cache capacity. It can also be observed that the delay performance is significantly improved with cooperative caching strategy. This is due to the fact that local content popularity will be updated according to that of neighbors during the cooperation, which can better reveal the content aggregation effect. Moreover, the proposed scheme can achieve a nearly ideal delay performance and is apparently superior to the BP algorithm. This is because the improved PIO algorithm maintains a better balance between global search and local search.

In Fig. 3, the iterative procedure of the average download delay of different F-APs in the considered cluster is plotted with the cache capacity Q set to 10 contents. It can be observed that it generally takes more iterations to converge when cooperative caching strategy is adopted. This is because the estimated local content popularity will be updated for each cooperation so the algorithm has to converge again. It can also be observed that a significantly smaller download delay is obtained when cooperative strategy is adopted. Moreover,

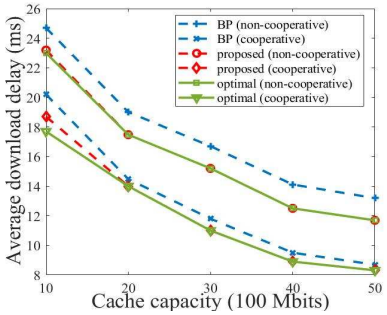

Fig. 2. Average download delay vs cache capacity

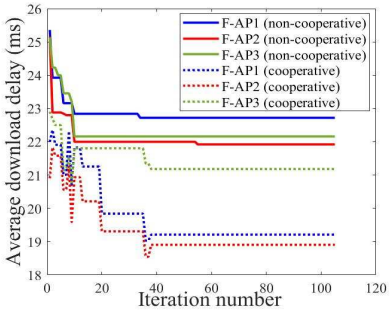

Fig. 3. Average download delay vs iteration number

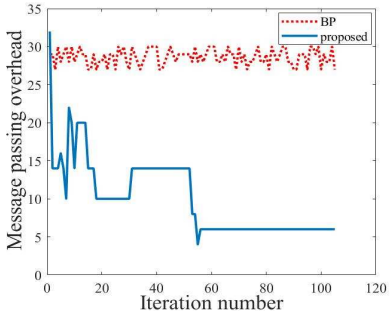

Fig. 4. Message passing overhead vs iteration number

the proposed algorithm converges within 40 iterations for both cooperative strategy and non-cooperative strategies. This is because a better performance in global search and local search is achieved by adopting the self-adaptive factor in our proposed scheme.

In Fig. 4, the message passing overhead of the proposed scheme is plotted in comparison with that of the BP algorithm. It can be observed that the proposed cooperative edge caching scheme has a much lighter message passing overhead than the baseline. This is because the BP algorithm requires belief to be transmitted between all the neighbouring nodes in the networks in each iteration, which results in a relatively heavy message passing overhead. However, there are only control signals and contents transmitted between the cooperative F-APs in the proposed scheme.

VI. Conclusions

In this paper, we have investigated the cooperative edge caching problem in F-RANs to minimize the average download delay. We have transformed and decomposed the original complex optimizing problem into two integer linear programming subproblems. We have proposed a cooperative edge caching scheme based on the improved PIO. Specially, the proposed scheme can avoid pre-mature convergence and achieve a better global and local search performance by adopting Cauchy perturbation and self-adaptive factor. Simulation results have demonstrated that our proposed scheme achieves a better delay performance than the baselines.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 61971129, the Natural Science Foundation of Jiangsu Province under Grant BK20181264, the National Key R&D Program of China under Grant 2018YFB1801103, the Research Fund of the State Key Laboratory of Integrated Services Networks (Xidian University) under Grant ISN19-10, the Research Fund of the Key Laboratory of Wireless Sensor Network & Communication (Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences) under Grant 2017002 and the Research Fund of the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen) under Grant HITSZ20190631.

References

[1] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, Jul. 2016.

[2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[3] X. Cui, Y. Jiang, X. Chen, F. Zhengy, and X. You, "Graph-based cooperative caching in Fog-RAN," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, Mar. 2018, pp. 166–171.

[4] Y. Jiang, W. Huang, M. Bennis, and F. Zheng, "Decentralized asynchronous coded caching design and performance analysis in fog radio access networks," *IEEE Transaction on Mobile Computing (Early Access)*, pp. 1–11, Jan. 2019.

[5] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[6] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in Fog-RANs: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7039–7051, Nov. 2017.

[7] ——, "Content caching at the wireless network edge: A distributed algorithm via belief propagation," in *2016 IEEE International Conference on Communications (ICC)*, May. 2016, pp. 1–6.

[8] T. X. Tran, A. Hajisami, and D. Pompili, "Cooperative hierarchical caching in 5G cloud radio access networks," *IEEE Network*, vol. 31, no. 4, pp. 35–41, Jul. 2017.

[9] P. Lin, Q. Song, Y. Yu, and A. Jamalipour, "Extensive cooperative caching in D2D integrated cellular networks," *IEEE Communications Letters*, vol. 21, no. 9, pp. 2101–2104, Sep. 2017.

[10] F. Xu and M. Tao, "Fundamental limits of decentralized caching in Fog-RANs with wireless fronthaul," in *2018 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 1430–1434.

[11] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[12] L. Gan and H. Duan, "Robust binocular pose estimation based on pigeon-inspired optimization," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, Jun. 2015, pp. 1043–1048.

[13] X. Li, X. Wang, S. Xiao, and V. C. M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," in *2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 5652–5657.

[14] J. Tang and X. Zhao, "Particle swarm optimization with adaptive mutation," in *2009 WASE International Conference on Information Engineering*, vol. 2, Jul. 2009, pp. 234–237.