Computer Networks 126 (2017) 200–217

Contents lists available at ScienceDirect

Computer Networks

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

ACO-inspired Information-Centric Networking routing mechanism

Jianhui Lv[a], Xingwei Wang[b,*], Kexin Ren[a], Min Huang[c], Keqin Li[d]

[a] College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[b] College of Software, Northeastern University, Shenyang, 110169, China
[c] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[d] Department of Computer Science, State University of New York, New York 12561, USA

ARTICLE INFO

Article history:
Received 19 December 2016
Revised 2 May 2017
Accepted 12 July 2017
Available online 13 July 2017

Keywords:
Information-centric networking
Bio-inspired networking
Routing
Ant colony optimization
Content concentration

ABSTRACT

In recent years, the bio-inspired solution has been employed to address routing optimization issue intelligently without manual intervention. In this paper, we propose a novel Ant Colony Optimization (ACO)-inspired Information-Centric Networking (ICN) Routing mechanism (ACOIR) by mapping ACO into ICN. At first, we devise a content management strategy based on the storage of name prefix to help conveniently and effectively manage and provide contents. Secondly, we propose a continuous model for content concentration by considering dynamic environment to conduct interest forwarding. Thirdly, we give a computation scheme about forwarding probability with physical distance and content concentration considered to determine the forwardable outgoing interface. Finally, we propose a comprehensive routing mechanism based on probabilistic forwarding to retrieve the most suitable content copy. We evaluate the proposed ACOIR, and the experimental results demonstrate that ACOIR can obtain the optimal solution and has better performance than other methods.

1. Introduction

Nowadays, Information-Centric Networking (ICN) has been accepted as a new paradigm to indicate that information objects are more important than IP addresses. The profound fruit brought by the change of communication mode can effectively achieve content distribution and support mobility [1]. However, moving the focus from IP addresses to information objects raises ICN routing scalability issue, because the amount of contents in network is considerably enormous. Especially when non-aggregatable flat names and even hierarchical names are employed, the routing table size increases explosively [2]. Besides, it is difficult for interest request to retrieve the content copy optimally, because the routing (table) in ICN is stateless and has no adaptability brought by the usage of distributed forwarding strategy. In addition, users in ICN usually need to send new interest requests when the content cannot be retrieved. In particular, when nodes or links cannot work effectively, the re-routing problem is very difficult to be solved. In that case, the network has to need manual intervention in order to make it work normally. Many kinds of ICN routing schemes have been proposed to solve the above mentioned routing problems, for example, forwarding interest request via the so-called best outgoing interface, exploiting in-network caching capability, and even devising other routing styles based on new architectures. However, the corresponding results are not adequately effective. Given this consideration, the effective ICN routing scheme should be further designed.

Recently, the bio-inspired solution has been investigated to solve the routing optimization problem [3–5], and its research can usually be divided into three fields, i.e., system, networking and computing. Among them, the bio-inspired system is capable of adapting and learning how to react to unforeseen scenarios with emergent properties. The bio-inspired networking is capable of providing new services and applications by considering networking features. The bio-inspired computing is capable of doing some operations according to the inherent computing rules and behaviors of biology. In fact, the bio-inspired solution can overcome the above mentioned three limitations in ICN routing due to its self-evolution, self-organization and survivability [6]. Naturally, the bio-inspired ICN routing is promising and feasible. As we know, most researches on bio-inspired ICN routing focus on Ant Colony Optimization (ACO) [7]. The principle of ACO is derived from the natural ant behaviors when searching for the shortest path between nest and food source. Meanwhile, ants communicate indirectly by laying the corresponding pheromone and following the trail with high pheromone, as a result, the pheromone accumulates on the shortest path. Although ACO-inspired ICN routing has been

* Corresponding author.
E-mail addresses: lvjianhui2012@163.com (J. Lv), wangxw@mail.neu.edu.cn (X. Wang), rkx51005@163.com (K. Ren), mhuang@mail.neu.edu.cn (M. Huang), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.comnet.2017.07.004
1389-1286/

proposed, its feasibility has not been analyzed in the other literatures. Then, we summarize five comprehensive explanations to illustrate the feasibility of ACO-inspired ICN routing as follows.

- "What" not "where": ICN pays attention to the content rather than IP-address, in which interest packet is used to retrieve the content irrespective of its physical location. ACO concentrates on what food is rather than where food is, in which ant is used to find the unknown food.
- Naming: ICN relies on the name-based routing, where content name is persistent, available and authentic. The food name in ACO is also unique and it exists in the natural world, thus ACO relies on food name to find the route for ant. In addition, both content types and food odors are diverse.
- Consumer-driven: In ICN, content provider does not provide the content before sending content request from interest requester; when the content is obtained, it is returned to interest requester no matter which content provider it comes from; it is obvious that ICN is the interest-driven mode. In ACO, food is not likely to be provided for ant before food request is sent, and food only needs to meet the ant's requirements regardless of which ant wants; in other words, ACO belongs to the ant-driven mode.
- Mobility: ICN supports mobility of interest requester excluding that of content; in other words, the content can be returned to interest requester no matter where interest requester moves, however, interest requester cannot obtain the content effectively when the content moves. In ACO, ants can find food by their cooperation and organization no matter where food moves, and food can be also carried to nest no matter where nest moves, which displays that ACO supports mobility of food and nest.
- Multiple resources and most suitable resource: In ICN, Content Routers (CRs) cache the multiple content copies, and they always provide the most suitable resource (*e.g.*, the closest content copy). In ACO, there are many same food sources in the natural world, and ants find the most suitable food along the shortest path in a distributed and parallel manner.

Furthermore, existent ACO-inspired ICN routing proposals only adopt computing rules of ant (bio-computing) regardless of considering ICN features (bio-system and bio-networking). In other words, they only pay attention to design and update pheromone, without considering Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB) together. Besides, their corresponding updating strategies of pheromone are discrete, which is against the actual ant behaviors. Therefore, it is considerably essential to design a comprehensive mechanism to solve the ICN routing optimization problem from the perspective of system, networking and computing in order to make modelling process accord with the actual ant behaviors better.

In this paper, we propose a novel ACO-inspired ICN Routing mechanism (ACOIR), which maps the system models of ACO into ICN, and the major contributions are summarized as follows. (i) We map ACO into ICN with environment constraint and routing scenario, and propose the system framework of ACOIR by simulating ant behaviors to retrieve the most suitable content. (ii) To help manage contents in CS conveniently and provide them effectively, we devise a content management strategy based on the storage of Name Prefix Trie (NPT). (iii) To self-adaptively conduct interest forwarding, we propose a continuous content concentration model under dynamic environment in PIT. (iv) In order to determine the suitable outgoing interface in FIB to forward interest request, we propose a computation scheme about forwarding probability based on physical distance and content concentration. (v) To retrieve the most suitable content copy with good performance, we devise the comprehensive mechanism of ACOIR based on probabilistic forwarding and further prove its covergence.

The rest of this paper is structured as follows. The related work is reviewed in Section 2. Section 3 maps ACO into ICN. Section 4 presents the proposed ACOIR mechanism. The performance evaluation is done in Section 5. Finally, Section 6 concludes this paper.

2. Related work

2.1. ACO-inspired routing solutions in non-ICN

There are some ACO-inspired routing researches in traditional networks other than ICN-alike. In [8], an intelligent routing scheme based on ACO in peer-to-peer networks was proposed. It regarded the message which was forwarded successfully as agent and further used biological procedure to forward the following packets for resource discovery. In [9], an adaptive ACO-based pheromone diffusion routing framework was proposed by introducing network information region, in which spatial and temporal network information were exchanged among adjacent routers. In [10], a clustering-based ACO scheme was proposed to handle vehicle routing problem in dynamic environment covering both random and cyclic traffic situations. In [11], an ACO-based routing scheme in optical networks was proposed to deal with Byzantine failures. It used a crankback re-routing mechanism to circumvent congestion during the light-path establishment process.

Furthermore, ACO-inspired solutions have been used to solve the energy-efficient routing problem. In [12], a self-adaptive energy saving routing mechanism based on ACO was proposed to make the Internet more energy-efficient. It heuristically solved the formulated NP-hard routing problem without any supervised control by allowing incoming flows to be autonomously aggregated on specific heavy-loaded links and switching off light-loaded links. In [13], an ACO-based approach was used to compute routing table in a decentralized manner in order to minimize the global energy consumption.

Moreover, ant-like methods have been suitable for dynamic networks such as Mobile Ad hoc NETworks (MANETs) due to the adaptability to the changing environment. In [14], an improved ACO-based dynamic source routing scheme in MANETs was proposed. It focused primarily on efficient routing by avoiding congestion and link breakage phenomena to produce high data packet delivery ratio with low end-to-end delay, routing overhead and energy consumption. In [15], an ACO-based power saving routing algorithm in MANETs was proposed to enhance adaptability and stability in reacting to node movement. According to the above, ACO has been applied to various kinds of networks.

Although [8–15] showed remarkable results, they were not designed for ICN routing optimization. In addition, their corresponding pheromone models were discrete; instead, we devise a continuous pheromone model which is closer to the ant foraging behaviors.

2.2. ACO-inspired ICN routing solutions

There have been a lot of ACO-inspired ICN routing proposals. In [16], Content-Centric Networking (CCN) architecture was extended to support service routing decision. It used a distributed scheme to gather service information based on ACO in order to deliver a service request to the best service instance. However, [16] assumed that the best service instance was known, which was against the fact that content/service provider was unknown in ICN routing. Besides, [16] neither considered content feature (type) nor had enough controllers on ants. In [17], an improved CCN routing to [16] was proposed based on the combination of Genetic Algorithm (GA) and ACO. It used the fast random global search capability of GA and the distributed parallel search capacity of ACO to obtain high routing success rate with low delay. However, [17] only used positive feedback feature to seek the exact solution regardless of diversity feature. In addition, the proposed routing scheme in [17] had not been verified. In [18], an ACO-based routing was proposed to enable the optimal forwarding decision for delay-sensitive traffic, and the forwarding interface with large pheromone value had high priority. However, [18] only forwarded interest request via outgoing interface with the highest priority, which caused serious load in terms of some links. In fact, [18] also played positive feedback feature excessively. In [19], a multipath transmission routing mechanism was proposed with probabilistic ACO approach. It supported data stream transmission through multiple links to achieve high throughput. However, [19] forwarded interest request to all outgoing interfaces; in other words, it played diversity feature excessively, which generated redundant packets and reduced network performance. In [20], a greedy ant colony forwarding algorithm was proposed to adaptively decrease influence caused by the inherent problems, such as link failure, network congestion and dynamic topology. It used two kinds of ants to complete routing optimization process, i.e., one was to discover all possible paths and optimize them, the other was to reinforce path optimization with Quality of Service (QoS) considered, such as bandwidth, delay, delay jitter and cost. However, the two-stage routing strategy in [20] increased content retrieval time and resource consumption. In [21], a QoS-supported routing strategy was proposed. It had ability to consistently select a route to meet the user's QoS requirements with low packet loss rate. However, [21] consumed a large amount of time to capture the link state information such as delay, cost and error rate. In [22], an adaptive forwarding strategy was proposed. It used ACO to compute forwarding probability of outgoing interface in order to reduce transmission delay and balance network overhead. However, [22] did not consider trade-off between positive feedback feature and diversity feature.

Furthermore, [16–22] did not take mobility into full consideration. In [23], a dynamic ACO-based routing scheme was proposed to cope with content movement. However, it could not obtain the shortest path from interest requester to content provider under the given situation because interest request had no ability to detect content provider after content movement when the pheromone was not enough.

Moreover, [16–23] all presented discrete pheromone models regardless of continuous foraging behaviors during the ant routing process. Therefore, we proposed a continuous pheromone model in order to make modelling process better fit the actual ant behaviors. For example, in [6], we simulated the situation where ACO was used to solve Traveling Salesman Problem (TSP) to design forwarding probability with content concentration and similarity relation considered. In addition, [6,16–23] always had some limitations to be improved. At first, they did not provide sufficient evidences on why ACO could be applied to ICN routing. In this paper, we illustrate the feasibility of ACO-inspired ICN under environment constraint and routing scenario. Secondly, they only leveraged ant behaviors to compute forwarding probability and did not consider the inherent ant features from the perspective of system and networking. In this paper, we consider CS, PIT and FIB comprehensively, including content management, content concentration design, forwarding probability computation and routing decision. Thirdly, they did not provide theoretical analysis for their designed schemes. In this paper, we present some mathematical proof to show the feasibility of the proposed routing mechanism.

3. Mapping ACO into ICN

In this section, we map ACO into ICN under environment constraint and routing scenario, which indicates the feasibility of ACO-inspired ICN. Furthermore, we present the system framework of ACOIR by simulating ant behaviors.

3.1. Environment constraint

As depicted in Fig. 1, ACO has three main roles, i.e., ant, food and pheromone. It is easy to understand that ant is regarded as interest packet (inp) or data packet (dap), and food is regarded as content. For convenience, in this paper, we call inp and dap as inp-ant and dap-ant respectively. The pheromone is key factor to connect ant with food, at the same time, content concentration is key factor to support content retrieval, thus pheromone is regarded as content concentration. We present a definition about content concentration as follows.

Definition 1 (content concentration)**.** A group of inp-ants send their same content request with a lot of iterations from one CR to another CR, and the requested content information is laid, dispersed and accumulated over link or path between two CRs. The accumulated amount of information by this group of inp-ants is regarded as content concentration.

At first, we discuss the differences among content concentration, content popularity and pheromone as follows.

- Content concentration & content popularity: (i) The initial content concentration is zero rather than the accumulated value, that is, content concentration for each inp-ant routing is always generated starting from zero. Instead, content popularity is the consistent accumulation process in terms of all same request contents, that is, the current content popularity is influenced by the previous one. (ii) Content popularity reflects the requested number of contents within a certain time period, and it is not dispersed over link or path. Instead, content concentration reflects the number of inp-ants over link or path, and it is dispersed.
- Content concentration & pheromone: Although pheromone and content concentration have the wonder of using different approaches to achieve equally satisfactory results, ants in ACO release the same kind of pheromone no matter which kind of foods are requested; instead, content concentration in ACO-inspired ICN can reflect different kinds of content requests for different inp-ants. Thus, content concentration in ACO-inspired ICN reflects more fine-grained content type than pheromone in ACO.

Then, we present three illustrations about ACO and ACO-inspired ICN as follows.

- The purpose of ACO is to find food, and ants travel based on pheromone in an interactional, cooperative and self-organizing way; inp-ants retrieve their interested content based on content concentration in a parallel and interactional way rather than individual action.
- The natural environment makes pheromone fade and even disappear as time elapses whilst the network environment makes content concentration weaken and even be null; meanwhile, the change is gradual at a certain rate.
- Each ant lays pheromone into environment and communicates with others indirectly based on pheromone when it finds food, as a result, other ants will be attracted. Each inp-ant lays the requested name information into the network and communicates with others indirectly based on content concentration

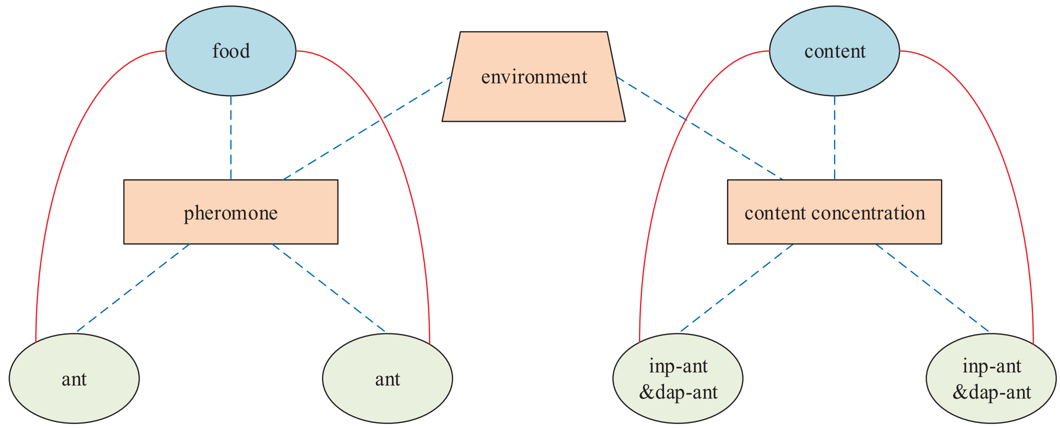


Fig. 1. ACO and ACO-inspired ICN.

when it finds the content, and inp-ants draw around the content.

3.2. Routing scenario

In this paper, we consider the ant foraging process as the routing process of which inp-ant retrieves the content in ICN. Each location in ACO consists of three components, i.e., Food Warehouse (FW) to store foods, Pheromone Matrix (PM) to percept pheromone, and Tabu Search Table (TST) to conduct the travel direction for ant. Each CR in ICN consists of three components, i.e., CS to store contents to be provided for inp-ants, PIT to percept content concentration, and FIB to record the forwardable outgoing interface for inp-ant. In fact, CS, PIT and FIB correspond to FW, PM and TST respectively. The routing scenarios of ACO and ACOIR are shown in Fig. 2.

For an ant foraging process, it can be simply described as follows. Each ant starts its travel to find food from nest. When arriving at a location, it searches FW to see whether the interested food can be found. If yes, it accomplishes foraging and goes back to nest; otherwise, it percepts pheromone by PM and begins the following travel by TST. Similarly, we map ant behaviors into inp and dap, and each inp-ant retrieves the content starting from interest requester. When arriving at one CR, it checks CS to see whether the content exists. If yes, dap-ant goes back to interest requester and content retrieval is finished; otherwise, it percepts content concentration by PIT and begins the following forwarding by FIB.

We give some special illustrations about ACOIR as follows.

- ACOIR is different from that using ACO to solve some other classical problems. For the former, inp-ants are to retrieve the unknown content copy (i.e., destination is unknown). However, ants are usually to find some interested things at the known location (i.e., destination is known) for the latter, such as TSP [7] and the shortest path problem in network [8]. In short, the former is much closer to the actual ant foraging behaviors than the latter.
- The foraging process in ACO is unrestricted under the natural environment, that is, each ant can crawl to the interested food freely. The content retrieval in ACOIR is restricted under the undirected connect graph, thus the latter is easier to be done than the former.
- ACOIR is different from the vanilla ICN routing scenario, especially during the process of checking PIT. The former simulates ant behaviors and further percepts content concentration while without spending much time waiting for dap-ant. However, the latter needs to spend much time dealing with PIT, for example, checking whether the similar or even same interest requests have arrived.

3.3. ACOIR framework

As shown in Fig. 3, ACOIR is composed of the following four modules, i.e., Content Management Module (CMM), Content Concentration Module (CCM), Forwarding Probability Module (FPM) and Routing Decision Module (RDM). Among them, CMM is used to manage contents in CS and provide them for inp-ants; CCM is used to percept content concentration laid by inp-ant; FPM is used to record outgoing interface and compute the corresponding forwarding probability. In fact, CMM, CCM and FPM are used for the local inp-ant forwarding in terms of one CR. RDM is used to retrieve the most suitable content copy for inp-ant, and it is used for the global inp-ant routing among CRs.

4. The Proposed ACOIR Mechanism

4.1. Content management (CS/CMM)

CS is responsible for storing contents, and the speed of content retrieval depends on storage approach and lookup way. Thus, an effective content management strategy is needed.

4.1.1. Storage

In ACO, when an ant arrives at FW which is considered as a small world, instead of looking up food blindly, it finds food by the corresponding smell since different foods have different smells. This enlightens us to store contents according to content type (*e.g.*, sport, travel and shopping) which represents a class of interests extracted from content name. In this way, the content can be looked up according to the corresponding content type instead of searching from the first content item to the last one orderly, thus lookup efficiency is improved. In this paper, we use NPT [24] to store contents, and each content type is considered as a subtree.

The content name consists of length-variable strings, and it is separated by dots and/or slashes, at the same time, each string is usually composed of “a–z”, “A–Z”, “0–9” and “_” [25]. Suppose that CS stores N content items, and each one is denoted as cn_p, $1 \leq p \leq N$. We present two definitions about content name length and NPT height as follows.

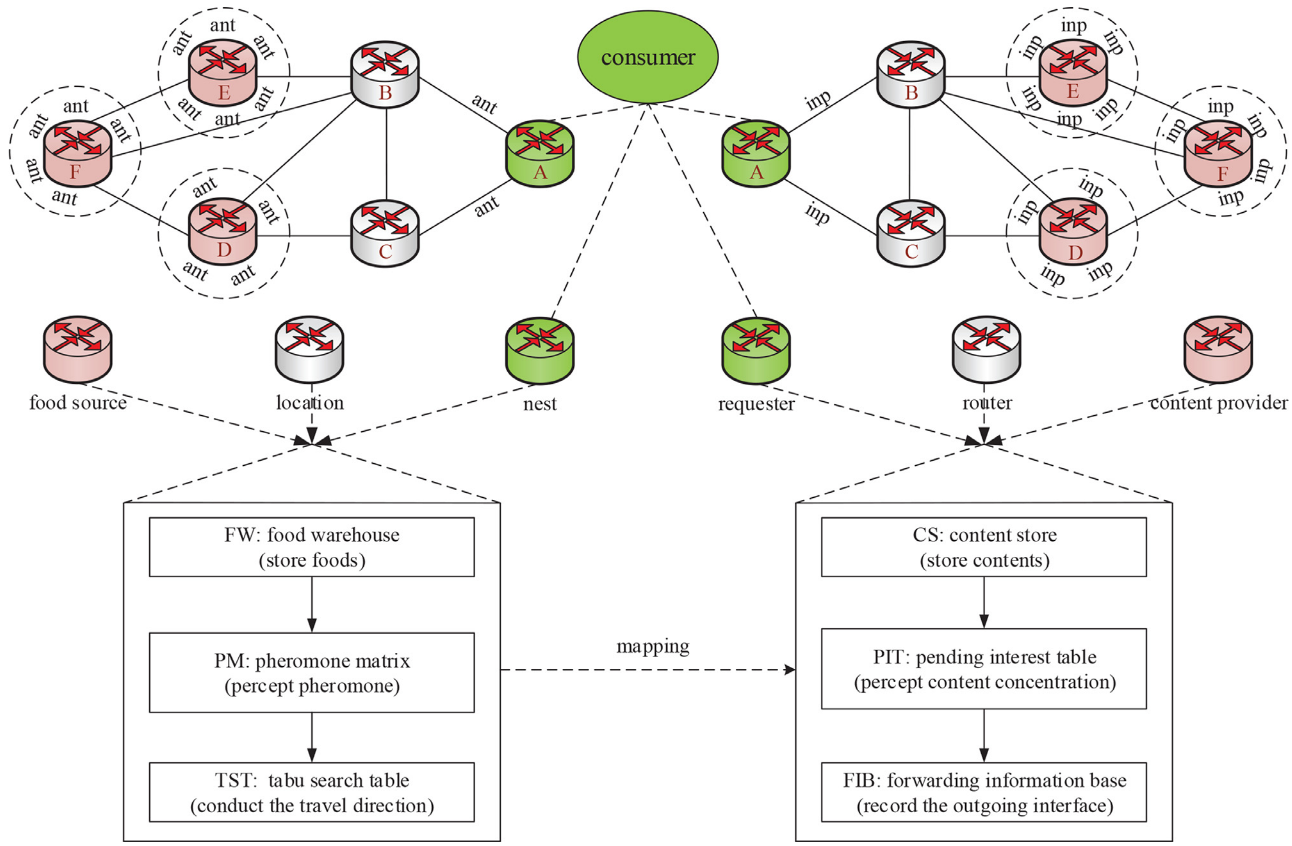


Fig. 2. Routing scenarios of ACO and ACOIR.

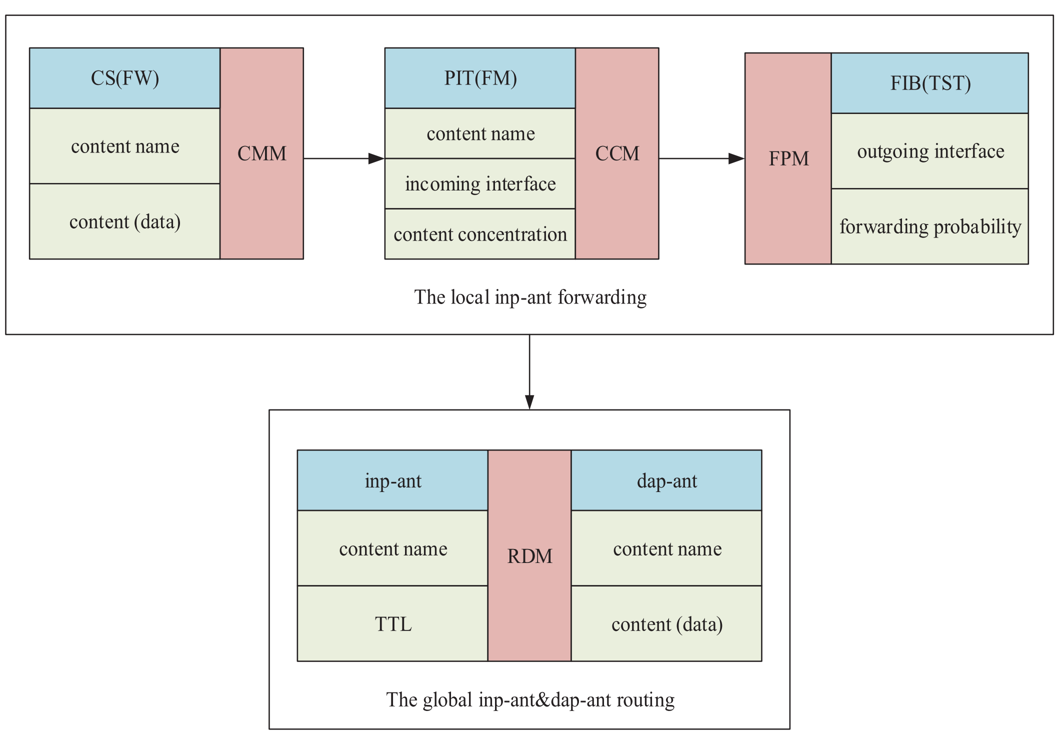


Fig. 3. The system framework of ACOIR.

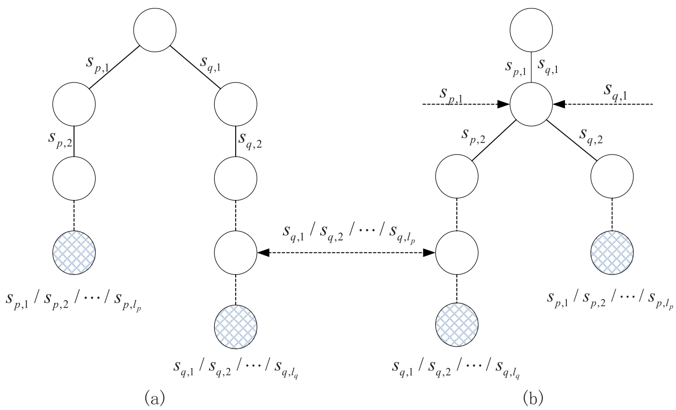


Fig. 4. NPT of cn_p and cn_q.

Definition 2 (content name length)**.** $\forall$ cn_p, if it is converted to $s_{p,1}/s_{p,2}/\cdots/s_{p,l_p}$, here $s_{p,\,k}$ $(1 \le k \le l_p)$ is an independent string excluding both dot and slash, l_p is the length of cn_p.

Definition 3 (NPT height)**.** Let $h_{\max}$ represent the height of NPT generated from these content names in CS, and

$$h_{max} = \max_{p=1}^{N} l_p. \tag{1}$$

$\forall$ cn_p and cn_q, $p \ne q$, $1 \le q \le N$, on condition of $l_p < l_q$, if $s_{p,1} \ne s_{q,1}$, NPT of cn_p and cn_q is presented in Fig. 4(a); if $s_{p,1} = s_{q,1}$ and $s_{p,2} \ne s_{q,2}$, NPT of cn_p and cn_q is presented in Fig. 4(b).

In Fig. 4, the gray node stores a content item including its name and the corresponding content. The white node does not include the content but it can store a content item. For example, $s_{p,\,1}$ $(s_{q,\,1})$ can be stored at the white node to which two dashed and one-way arrows point in Fig. 4(b).

How to add a new content item cn_x into NPT is introduced as follows. We present two different conditions: one is that the first independent string of cn_x is beyond $s_{1,1}$, $s_{2,1}$, $\cdots$, $s_{N,1}$, mathematically,

$$\left\{s_{x,1}\right\} \bigcap \left\{s_{p,1} \middle| 1 \le p \le N\right\} = \Phi; \tag{2}$$

the other is that $s_{x,\,1}/s_{x,\,2}/\cdots/s_{x,\,u}$ is found from NPT, while $s_{x,u+1}$ is beyond $s_{1,u+1}, s_{2,u+1}, \cdots, s_{N,u+1}$. Mathematically, the following (3) and (4) always hold.

$$\left\{s_{x,1}/s_{x,2}/\cdots/s_{x,u}\right\} \bigcap \left\{cn_p \middle| 1 \le p \le N\right\} = \left\{s_{x,1}/s_{x,2}/\cdots/s_{x,u}\right\}, \tag{3}$$

$$\left\{s_{x,u+1}\right\} \bigcap \left\{s_{p,u+1} \middle| 1 \le p \le N\right\} = \Phi. \tag{4}$$

In terms of the above two conditions where cn_x is added to NPT, they are similar to the condition where cn_q is added to NPT of cn_p, as shown in Fig. 4(a) and (b) respectively. However, cn_x has to be looked up before being added into NPT so that whether it exists in NPT can be known, which indicates the importance of lookup way.

4.1.2. Lookup

When an inp-ant arrives at one CR, it looks up CS to see whether the content exists. Meanwhile, NPT should be looked up quickly and effectively. In terms of these strings at the same level in NPT, they are looked up in sequential way. Suppose that the requested content name is denoted as cn_r, and the lookup process can be described as follows and in Algorithm 1: search $s_{r,\,k}$ at the level k in NPT from left to right starting from $k = 1$ until $s_{r,\,k}$ cannot be found or $k = l_r$.

Algorithm 1 Lookup algorithm.

Input: cn_r
Output: cn_r or failure
Begin
  for $k = 1$ to l_r, **do**
    Search $s_{r,k}$ in sequential way;
    if $s_{r,k}$ is not found, **then**
      Return failure;
  end for
  if $k == l_r$, **then**
    Return cn_r;
End

The following theorem shows the time complexity of Algorithm 1.

Theorem 1. *Algorithm 1 runs in O(N). The related proof is found in Appendix A.*

Corollary 1. *Consider the storage management with NPT as M1 and that without NPT as M2, and M1 has better performance than M2. The related proof is found in Appendix A.*

4.2. Content concentration (PIT/CCM)

PIT is responsible for doing perception of content concentration, and the convergence speed of ACOIR depends on the formulation of content concentration. Thus, an effective content concentration model is needed.

4.2.1. Thought

Suppose that there are m inp-ants and each one is denoted as ia_λ, here $1 \le \lambda \le m$. When ia_λ traverses from CR_i to CR_j, $1 \le i, j \le n$, here n is the number of CRs in network topology, it lays content concentration over its traversed trail. As a matter of fact, content concentration is dynamically changing. On one hand, the environment makes it weaken as time goes; on the other hand, the short distance between current location and interest requester means high content concentration. Thus, different locations at different times have different content concentrations. In Fig. 5, we model CR_i as the origin in a three-dimensional coordinate. Let $\tau^\lambda(t,\,d)$ represent content concentration at the location d and at the time t in terms of ia_λ, where d is distance-axis and t is time-axis, and we have

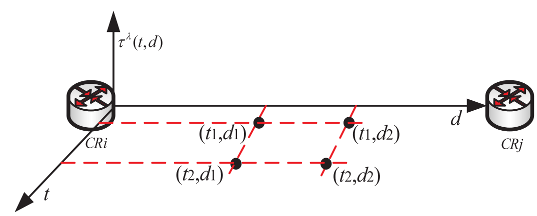


Fig. 5. The three-dimensional coordinate among $\tau^\lambda(t,\,d)$, t and d.

$$\tau^\lambda(t_1, d_1) \ne \tau^\lambda(t_1, d_2) \ne \tau^\lambda(t_2, d_1) \ne \tau^\lambda(t_2, d_2). \tag{5}$$

As a matter of fact, $\tau^\lambda(t,\,d)$ decreases with the increasing of t, *e.g.*, $\tau^\lambda(t_1,\,d_1) > \tau^\lambda(t_2,\,d_1)$ in case of $t_1 < t_2$, and it increases with the increasing of d, *e.g.*, $\tau^\lambda(t_1,\,d_1) < \tau^\lambda(t_1,\,d_2)$ in case of $d_1 < d_2$.

In order to build the model for $\tau^\lambda(t,\,d)$ conveniently, we have the following assumption.

Hypothesis 1. Assume that inp-ant only has ability to percept content concentration within one hop. In other words, when ia_λ is beyond the edge between CR_i and CR_j (denoted as $e_{i,\,j}$), it cannot percept content concentration over $e_{i,\,j}$.

4.2.2. Verification

At first, we discuss the relationship between $\tau^{\lambda}(t,\,d)$ and t. Since $\tau^{\lambda}(t,\,d)$ is a continuously differentiable function in terms of $t\in[0,+\infty)$, consider $d=0$, and we have

$$\begin{cases}\frac{\partial\tau^{\lambda}(t,0)}{\partial t}=-\theta\cdot\tau^{\lambda}(t,0)\\ \tau^{\lambda}(0,0)=\tau_0^{\lambda}.\end{cases} \tag{6}$$

Among them, the first one means that the change rate of content concentration is proportional to content concentration; the second one means that the initial content concentration is τ_0^{λ}; θ is a positive constant. It is necessary to point out that the change rate is equivalent to evaporation rate (denoted by er). The former is the mathematical gradient meaning while the latter is the physical interpretation regarding the former.

By solving the differential equation (6), we have

$$\tau^{\lambda}(t,0)=\tau_0^{\lambda}\cdot e^{-\theta\cdot t}. \tag{7}$$

According to (6) and (7), we have

$$er=\left|\frac{\partial\tau^{\lambda}(t,0)}{\partial t}\right|=\theta\cdot\tau_0^{\lambda}\cdot e^{-\theta\cdot t}, \tag{8}$$

which is a variable. This is different from that in the other literatures, where it is assumed as a constant. In comparison, our designed content concentration model is much closer to the actual ant foraging scenario from the perspective of evaporation rate.

In fact, if ia_{λ} does not arrive at a new location, the current content concentration is meaningless since a complete edge is not traversed. Suppose that ia_{λ} arrives at CR_j at t_{thr}, and we have $\tau^{\lambda}(t,d_{i,j})=0$ in case of $t<t_{thr}$, here $d_{i,\,j}$ is the distance between CR_i and CR_j. Furthermore, $\tau^{\lambda}(t_{thr},d_{i,j})=\tau_0^{\lambda}$.

Then, we discuss the relationship between $\tau^{\lambda}(t,\,d)$ and d. Since $\tau^{\lambda}(t,\,d)$ is a continuously differentiable function in terms of $d\in[0,\,d_{i,\,j}]$, we divide $[0,\,d_{i,\,j}]$ into ξ parts, $[0,\,d_1]$, $(d_1,\,d_2]$, $\cdots$, $(d_{\xi-1},\,d_{\xi}]$, here $d_{\xi}=d_{i,j}$. Suppose that ia_{λ} arrives at d_k at t_k, here $1\le k\le\xi$ and $t_{\xi}=t_{thr}$, because the initial content concentration is τ_0^{λ} when ia_{λ} arrives at a new location, we have

$$\tau^{\lambda}(t,d_k)=\tau_0^{\lambda}\cdot e^{-\theta(t-t_k)}, \tag{9}$$

especially when $d_k=0$, (9) is same as (7).

In order to build their relationship model conveniently, we have the following assumption.

Hypothesis 2. Assume that the rate of ia_{λ} at any time t remains constant, denoted as v_2.

Let ϖ and v_1 represent the size of inp-ant and the transmission rate of link respectively, based on Hypothesis 2, and we have

$$t_k=\frac{d_k}{v_2}+\frac{\varpi}{v_1}. \tag{10}$$

Putting (10) into (9), and we have

$$\tau^{\lambda}(t,d_k)=\tau_0^{\lambda}\cdot e^{-\theta\left(t-\frac{d_k}{v_2}-\frac{\varpi}{v_1}\right)}. \tag{11}$$

However, (11) is a special function in terms of d_k. By generalizing (11), i.e. considering d_k as d, we have

$$\tau^{\lambda}(t,d)=\tau_0^{\lambda}\cdot e^{-\theta\left(t-\frac{d}{v_2}-\frac{\varpi}{v_1}\right)}. \tag{12}$$

Furthermore,

$$\lim_{t\to+\infty}\tau^{\lambda}(t,d)=\lim_{t\to+\infty}\tau_0^{\lambda}\cdot e^{-\theta\left(t-\frac{d}{v_2}-\frac{\varpi}{v_1}\right)}=0, \tag{13}$$

which indicates that content concentration at d will disappear when t reaches the unlimited level.

4.2.3. Modelling

Let $cc_{i,j}^{\lambda}(t)$ represent content concentration over $e_{i,\,j}$ at the time t for ia_{λ}, according to the above, $cc_{i,j}^{\lambda}(t)$ is defined as follows.

$$\begin{aligned}cc_{i,j}^{\lambda}(t)&=\sum_{d\in[0,d_{i,j}]}\tau^{\lambda}(t,d)=\int_0^{d_{i,j}}\tau^{\lambda}(t,d)\,dd\\&=\frac{v_2\cdot\tau_0^{\lambda}}{\theta}\cdot e^{-\theta\left(t-\frac{\varpi}{v_1}\right)}\left(e^{\frac{\theta\cdot d_{i,j}}{v_2}}-1\right).\end{aligned} \tag{14}$$

According to (14), we know that large $d_{i,\,j}$ has high content concentration. Thus, we have to modify (14) to make it satisfy that small $d_{i,\,j}$ has high content concentration. Let $mcc_{i,j}^{\lambda}(t)$ represent the modified $cc_{i,j}^{\lambda}(t)$, and we have

$$mcc_{i,j}^{\lambda}(t)=\frac{L_{\lambda}}{cc_{i,j}^{\lambda}(t)}, \tag{15}$$

where L_{λ} is the total distance traversed by ia_{λ} within one iteration.

As a matter of fact, $\tau^{\lambda}(t,\,d)$ is a decreasing function in terms of t, then $cc_{i,j}^{\lambda}(t)$ should also be a decreasing function in terms of t. Thus, the inequality (16) always holds.

$$t-\frac{d_k}{v_2}-\frac{\varpi}{v_1}\ge 0. \tag{16}$$

Namely,

$$t\ge\max_{k=1}^{\xi}\left(\frac{d_k}{v_2}+\frac{\varpi}{v_1}\right)=t_{thr}. \tag{17}$$

According to (17), the definition domain of (14) is $t\in[t_{thr},+\infty)$, which is consistent with the practical situation where m inp-ants accomplish one iteration with t_{thr} at least. Then, for ia_{λ} within one iteration, let $mcc_{i,\,j}(t,\,I)$ represent content concentration over $e_{i,\,j}$ at the Ith iteration, and we have

$$mcc_{i,j}(t,I)=\sum_{\lambda=1}^{m}mcc_{i,j}^{\lambda}(t)\cdot x_{\lambda}, \tag{18}$$

where

$$x_{\lambda}=\begin{cases}1,\ ia_{\lambda}\ traverses\ e_{i,j}\\ 0,\ otherwise.\end{cases} \tag{19}$$

Suppose that m inp-ants need Δt_I to accomplish the Ith iteration, and the total content concentration over $e_{i,\,j}$ after I iterations is defined as follows.

$$\begin{aligned}Tcc_{i,j}(t,I)=\ &mcc_{i,j}(t-\Delta t_{I-1}-\Delta t_{I-2}-\cdots-\Delta t_1,I)\\&+mcc_{i,j}(t-\Delta t_{I-2}-\cdots-\Delta t_1,I-1)\\&+\cdots\\&+mcc_{i,j}(t-\Delta t_1,2)\\&+mcc_{i,j}(t,1).\end{aligned} \tag{20}$$

Furthermore,

$$Tcc_{i,j}(t+\Delta t_I,I)=Tcc_{i,j}(t+\Delta t_I,I-1)+mcc_{i,j}(\Delta t_I,I). \tag{21}$$

where $Tcc_{i,j}(t+\Delta t_I,\,I-1)$ and $mcc_{i,\,j}(\Delta t_I,\,I)$ represent the remaining and increased content concentration respectively.

For example, assume that $\Delta t_1=1\,\text{ms}$, $\Delta t_2=2\,\text{ms}$, $\Delta t_3=3\,\text{ms}$, $\Delta t_4=4\,\text{ms}$, $\Delta t_5=5\,\text{ms}$ and $\Delta t_6=6\,\text{ms}$. When $t=17\ \text{ms}\in[15\ \text{ms},21\ \text{ms}]$, the total content concentration over $e_{i,\,j}$ after 5 iterations is denoted as

$$\begin{aligned}Tcc_{i,j}(17,5)=\ &mcc_{i,j}(17,1)+mcc_{i,j}(16,2)+mcc_{i,j}(14,3)\\&+mcc_{i,j}(11,4)+mcc_{i,j}(7,5).\end{aligned} \tag{22}$$

Table 1
The computing process of Fw_i^λ in terms of the network topology in Fig. 2.

CR_i	Edge	Ad_i^λ	Re_i^λ	Ut_i^λ	Fw_i^λ
A	$--$	BC	BC	$BCDEF$	BC
B	$A\to B$	$ACDEF$	$CDEF$	$CDEF$	$CDEF$
D	$B\to D$	BCF	CF	CEF	CF
F	$D\to F$	BDE	BE	CE	E
E	$F\to E$	BF	B	C	Φ

4.3. Forwarding probability (FIB/FPM)

FIB is responsible for recording the forwardable outgoing interface and the corresponding forwarding probability. Thus, an effective computation scheme about forwarding probability is needed.

Before forwarding ia_λ from CR_i, the following three issues have to be addressed: (i) which interfaces of CR_i can be regarded as outgoing interfaces and used to forward ia_λ, (ii) how to obtain forwarding probability of outgoing interface at the first iteration for m inp-ants, and (iii) how to obtain forwarding probability of outgoing interface after the first iteration.

4.3.1. Forwardable outgoing interface

When ia_λ arrives at CR_i, not all interfaces can be used to forward ia_λ, that is, not all CRs which are adjacent to CR_i can receive ia_λ. We present four definitions about untraversed CRs, adjacent CRs, remaining CRs and forwardable outgoing interfaces as follows.

Definition 4 (untraversed CRs)**.** If the CRs are not traversed by ia_λ at CR_i, they are regarded as the untraversed CRs from the whole network perspective. The corresponding set is denoted by Ut_i^λ. Consider ia_λ traverses from CR_k to CR_i, and we have

$$Ut_i^\lambda = Ut_k^\lambda - \{CR_i\}. \tag{23}$$

Definition 5 (adjacent CRs)**.** If the CRs are adjacent to CR_i, they are regarded as the adjacent CRs of $CR_{i,}$ which depend on network topology. The corresponding set is denoted by Ad_i^λ.

Definition 6 (remaining CRs)**.** If ia_λ traverses from CR_k to CR_i, the remaining CRs of CR_i are equivalent to the adjacent CRs of CR_i excluding CR_k. The corresponding set is denoted by Re_i^λ. Mathematically,

$$Re_i^\lambda = Ad_i^\lambda - \{CR_k\}, \tag{24}$$

especially when CR_i is interest requester,

$$Re_i^\lambda = Ad_i^\lambda. \tag{25}$$

Definition 7 (forwardable outgoing interfaces)**.** If the interfaces of CR_i are used to forward ia_λ, they are regarded as the forwardable outgoing interfaces. The corresponding set is denoted by Fw_i^λ, and we have

$$Fw_i^\lambda = Re_i^\lambda \bigcap Ut_i^\lambda, \tag{26}$$

especially when $Fw_i^\lambda = \Phi$, CR_i cannot forward ia_λ, that is, the forwarding of ia_λ is over.

We present an example to illustrate the computing process of Fw_i^λ starting from A, the network topology is from Fig. 2, and the results are shown in Table 1.

4.3.2. Forwarding probability at the first iteration

The inp-ant retrieves the content based on content concentration. However, the initial content concentration is zero over all edges due to the fact that there is no inp-ant forwarding. In order to avoid forwarding inp-ant blindly, we consider the distance between CRs to compute forwarding probability at the first iteration. Let $fp1_{i,j}^\lambda(t)$ represent forwarding probability from CR_i to CR_j at the first iteration, and we have

$$fp1_{i,j}^\lambda(t) = \frac{\left(d_{i,j}(t)\right)^{-1}}{\sum\limits_{CR_\varsigma \in Fw_i^\lambda}\left(d_{i,\varsigma}(t)\right)^{-1}}. \tag{27}$$

According to (27), small distance between CR_i and CR_j means that CR_i forwards ia_λ to CR_j with large probability. The distance between CRs depends on network topology, thus $d_{i,j}(t)$ keeps unchanged as time goes on. Mathematically,

$$d_{i,j}(t) = d_{i,j}. \tag{28}$$

Furthermore,

$$fp1_{i,j}^\lambda(t) = \frac{\left(d_{i,j}\right)^{-1}}{\sum\limits_{CR_\varsigma \in Fw_i^\lambda}\left(d_{i,\varsigma}\right)^{-1}}. \tag{29}$$

4.3.3. Forwarding probability after the first iteration

After finishing the first iteration, some edges have content concentration to conduct the inp-ant forwarding. Let $fp2_{i,j}^\lambda(t)$ represent forwarding probability from CR_i to CR_j after the first iteration, and we have

$$fp2_{i,j}^\lambda(t) = \frac{Tcc_{i,j}(t,I)}{\sum\limits_{CR_\varsigma \in Fw_i^\lambda} Tcc_{i,\varsigma}(t,I)}. \tag{30}$$

According to (30), high content concentration over $e_{i,j}$ means that CR_i forwards ia_λ to CR_j with large probability. According to (29) and (30), we can obtain the comprehensive forwarding probability in terms of the Ith iteration, denoted by $fp_{i,j}^\lambda(t,I)$, as follows.

$$fp_{i,j}^\lambda(t,I) = \begin{cases} fp1_{i,j}^\lambda(t), I = 1 \\ fp2_{i,j}^\lambda(t), 1 < I \le I_{max}, \end{cases} \tag{31}$$

where $I_{\max}$ is the maximal number of iterations.

The outgoing interface with the highest $fp_{i,j}^\lambda(t,I)$ does not mean that it must be used to forward ia_λ, instead, the system generates a stochastic value st_i^λ for ia_λ before ia_λ makes forwarding decision at CR_i. In fact, selecting outgoing interface with the highest $fp_{i,j}^\lambda(t,I)$ to forward ia_λ means that the forwarding is greedy, thus it goes against diversity feature in spite of keeping positive feedback feature. The detailed forwarding decision of ia_λ based on st_i^λ is described in Section 4.4.2.

4.4. Routing decision (RDM)

In this section, we design the routing mechanism based on probabilistic forwarding rather than relying on deterministic path. The former can take full advantage of self-organization and cooperation to retrieve the most suitable content copy; meanwhile, the probabilistic routing processes for two different interest requests or two same interest requests at different moments are independent of each other. The latter usually refers to a fixed measure of how many interest requests have been successfully retrieved over a path, which only indicates that the path is empirical and even credible but does not ensure that the path can conduct interest request to retrieve the most suitable content copy; meanwhile, the deterministic routing processes for two same interest requests at different moments are mutual influence. For example, once the content moves from the most suitable source to another place, the deterministic is no longer reliable because ICN only supports mobility of interest requester excluding that of content [26].

4.4.1. Termination condition

We discuss two types of termination conditions, one is for the local forwarding of ia_λ within one iteration and the other is for the global inp-ant routing process. For the first one, it is composed of the following three cases: (i) the content is found, (ii) the survival time of ia_λ expires, i.e., TTL (Time To Live) is over, and (iii) $Fw_i^\lambda = \Phi$; when one of them is satisfied, the forwarding of ia_λ is finished. For the second one, when one CR is drew from a certain amount of inp-ants or $I = I_{\max}$, the global inp-ant routing is finished.

4.4.2. Forwarding decision

We discuss forwarding decision of ia_λ. Suppose that CR_i has w_i forwardable outgoing interfaces to forward ia_λ and their corresponding CRs are denoted by $CR_{i1}, CR_{i2}, \cdots, CR_{iw_i}$, we have

$$Fw_i^\lambda = \left\{ CR_{io} \middle| 1 \le o \le w_i \right\}. \tag{32}$$

Consider $\forall\ st_i^\lambda \in (0, 1)$, if

$$\sum_{o=1}^{\kappa} fp_{i,io}^\lambda(t, I) \ge st_i^\lambda, \tag{33}$$

where

$$\sum_{o=1}^{w_i} fp_{i,io}^\lambda(t, I) = 1, \tag{34}$$

the outgoing interface to which $CR_{i\kappa}$ corresponds is used to forward ia_λ.

In fact, (33) and (34) indicate diversity feature of inp-ant, that is, each outgoing interface of CR_i can be used to forward ia_λ due to the stochastic st_i^λ.

Furthermore, suppose that CR_{io} receives m_{io} inp-ants from CR_i, $m_{io} \le m$, and we have

$$\sum_{o=1}^{w_i} m_{io} = m. \tag{35}$$

In this paper, assume that m_{io} is proportional to $fp_{i,io}^\lambda(t, I)$, mathematically,

$$m_{io} \sim fp_{i,io}^\lambda(t, I). \tag{36}$$

Suppose that $m = 10$, $fp_{i,1}^\lambda(t, I) = 0.65$, $fp_{i,2}^\lambda(t, I) = 0.18$ and $fp_{i,3}^\lambda(t, I) = 0.17$, and we have (i) $m_1 = 6$, $m_2 = 2$, $m_3 = 2$, (ii) $m_1 = 7$, $m_2 = 2$, $m_3 = 1$, or (iii) $m_1 = 7$, $m_2 = 1$, $m_3 = 2$.

In fact, (35) and (36) means that outgoing interface with higher $fp_{i,io}^\lambda(t, I)$ is used to forward more inp-ants, which embodies positive feedback feature of inp-ant.

4.4.3. Routing process

The ACOIR process is composed of the global inp-ant routing and the dap-ant routing. The global inp-ant routing is to find the content by multiple forwarding with m inp-ants, going through I iterations. The dap-ant routing is to return the content to interest requester by generating a certain amount of dap-ants according to the size of content, and it starts when the global inp-ant routing is finished.

- The global inp-ant routing
  For one iteration, when CR_i receives the requested content name cn_r from ia_λ, check whether (i) cn_r can be found from NPT, (ii) TTL is over, or (iii) $Fw_i^\lambda = \Phi$. If one of them is satisfied, the forwarding of ia_λ is finished; otherwise, ia_λ adds cn_r into PIT and percepts content concentration with (20), at the same time, it selects an outgoing interface to forward itself with (33). For m inp-ants, they start next iteration after accomplishing one iteration. In order to accelerate the global inp-ant routing, we consider that, when a certain amount of inp-ants draw around one CR, the global inp-ant routing is finished and the CR holds the closest content copy. Mathematically,

$$m_0 > \frac{1}{2} m, \tag{37}$$

  where m_0 is the number of inp-ants which draw around the CR. Then, consider the worst condition where the number of iterations reaches $I_{\max}$, i.e.,

$$I = I_{max} \tag{38}$$

  holds, the global inp-ant routing is finished as well. However, this case means that the routing is failed.
  In short, either (37) or (38) is satisfied, the global inp-ant routing is finished. Meanwhile, the routing is successful when (37) is satisfied, and it is failed when (38) is satisfied.
- The dap-ant routing
  When the global inp-ant routing is finished, content provider generates some dap-ants. Let Dn and Sc represent the number of dap-ants and the size of content respectively, and we have

$$Dn = \left\lceil \frac{Sc}{2^\chi\ B} \right\rceil, \tag{39}$$

  where χ is a positive integer and B is the unit of byte.
  These Dn dap-ants carry their corresponding contents to interest requester along the path which is reverse with that of inp-ant by looking up the corresponding PITs. Let Sc_1 represent the size of content carried by the first $Dn - 1$ dap-ants and Sc_2 represent that carried by the last dap-ant, and we have

$$Sc_1 = 2^\chi\ B, \tag{40}$$

  and

$$Sc_2 = Sc - (Dn - 1) \cdot 2^\chi\ B. \tag{41}$$

  It is necessary to point out that the FIBs along the optimal path, which is resolved by m inp-ants during the global inp-ant routing, are updated during the dap-ant routing, because the global inp-ant routing is based on probabilistic forwarding. In this way, the proposed ACOIR can support mobility with more flexibility. For example, consider that some FIBs are updated, and the probabilistic forwarding is meaningless; especially when content movement happens before generating dap-ants, the FIBs updating has no help for the subsequent interest requests. In this paper, we do not give the detailed discussion about the dap-ant routing process due to its simpleness.

According to the above statements, the proposed ACOIR is described in Algorithm 2.

The following theorem shows the time complexity of ACOIR algorithm.

Theorem 2. *Algorithm 2 runs in $O\big(I \cdot m(n \cdot N + n^2)\big)$. The related proof is found in Appendix B.*

4.5. Convergence analysis

In order to demonstrate the feasibility of ACOIR theoretically, we give two theorems: if ACOIR can be run with enough time, (i) it can retrieve the content with the probability approaching 1, and

Algorithm 2 ACOIR algorithm.

Input: m inp-ants
Output: the content or failure
Begin
for $I = 1$ to $I_{\max}$, **do**
  for $i = 1$ to n, **do**
    if (37) is satisfied, **then**
      Return the content;
    if the content is found,
    TTL is over or $Fw_i^{\lambda} = \Phi$, **then**
      break;
    else
      Add the requested content name into PIT;
      Percept the content concentration with (20);
      Select the outgoing interface with (33);
    end if
  end for
end for
while (38) is satisfied, **do**
  if all CRs does not satisfy (37), **then**
    Return failure;
end while
End

(ii) it can converge to the optimal solution with the probability approaching 1.

Theorem 3. *Suppose that X represents the event which the content is found with ACOIR and that P(X) represents the corresponding probability. When the content exists,* $\forall\ \varepsilon > 0,\ \varepsilon \to 0,\ I \to +\infty$, *we have*

$$P(X) \geq 1 - \varepsilon, \tag{42}$$

and

$$\lim_{I \to +\infty} P(X) = 1. \tag{43}$$

The related proof is found in Appendix C.

Theorem 4. *Suppose that* $P^*(I)$ *represents the probability which the optimal solution* s^* *is found at the first time by I iterations. When the content exists,* $\forall\ \varepsilon > 0,\ \varepsilon \to 0,\ I \to +\infty$, *we have*

$$\lim_{I \to +\infty} P^*(I) = 1. \tag{44}$$

The related proof is found in Appendix D.

5. Performance evaluation

5.1. Simulation setup

In this section, two real network topologies, i.e., NSFNET [19] and Deltacom [27] are used for performance evaluation, as shown in Figs. 6 and 7 respectively. Meanwhile, NSFNET topology consists of 14 nodes and 21 edges with 1 interest requester and 4 content providers; Deltacom topology consists of 97 nodes and 124 edges with 8 interest requesters and 5 content providers. In addition, we capture data from Sohu website during 1 week and every day for 1 h. For the captured data, we extract the names of interest requests from the HTTP requests, and each CR stores 10,000 content items by adopting Least Recently Used (LRU) replacement strategy.

In existing literatures, only [6,16–23], introduce ACO into ICN routing, where [6,16,18,19], present the relatively systematic designs. Thus, the methods in [6,16,18,19], i.e., ACO-inspired ICN Routing with Content concentration and Similarity relation (AIRCS), Services over Content-Centric Routing (SoCCeR), QoS-Aware Path

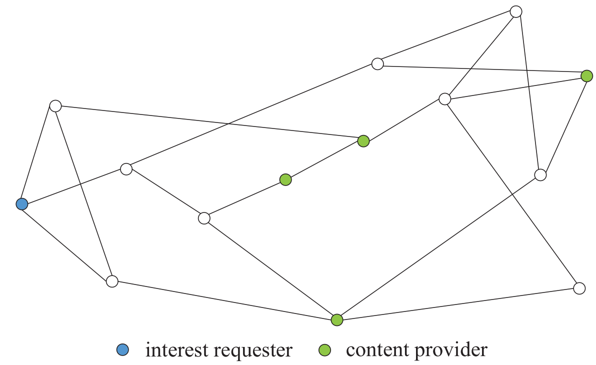


Fig. 6. NSFNET topology with 14 nodes and 21 edges.

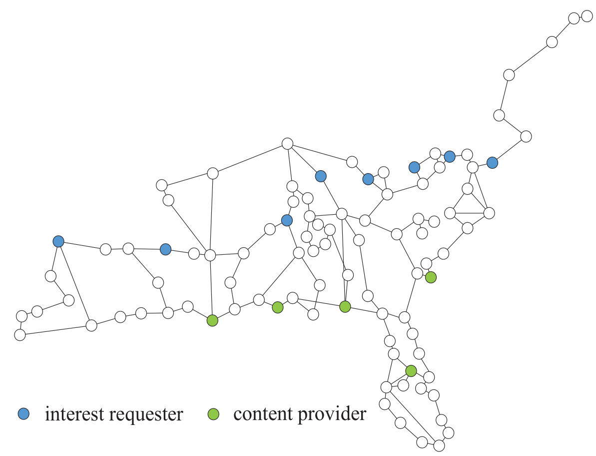


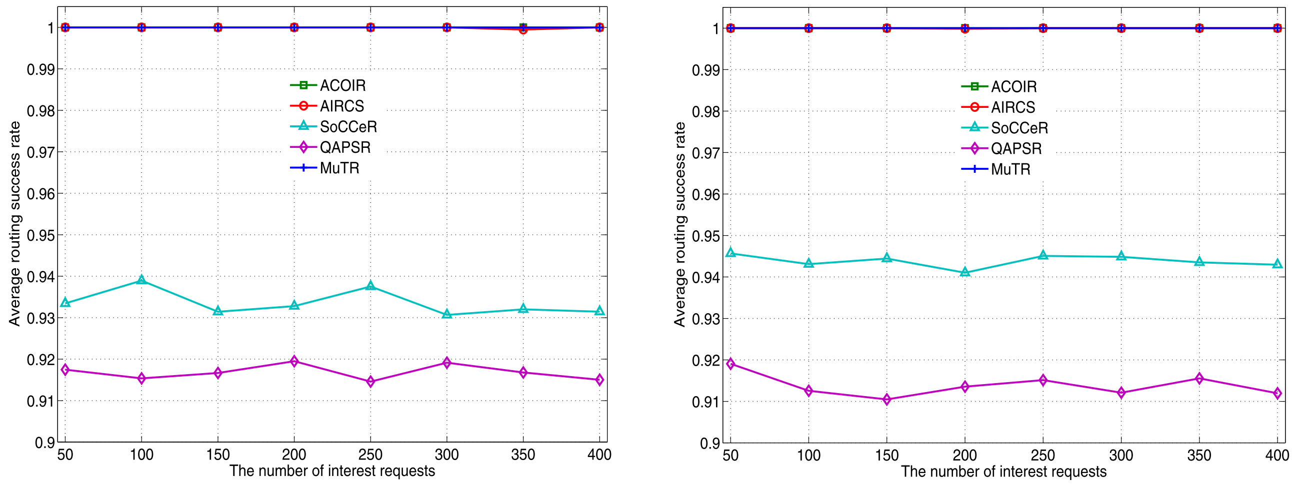
Fig. 7. 2010-Deltacom topology-USA with 97 nodes and 124 edges.

Selection Routing (QAPSR) and Multipath Transmission Routing (MuTR) are compared with ACOIR. In addition, Average Routing Success Rate ($\mathbb{ARSR}$), Average Iteration Times ($\mathbb{AIT}$), Average Routing Hops ($\mathbb{ARH}$), Average Routing Delay ($\mathbb{ARD}$), Average Time Overhead ($\mathbb{ATO}$) and Average Load Balance Degree ($\mathbb{ALBD}$) are considered as six evaluation metrics.

We have implemented ACOIR by C++ programming language on a personal computer with the Intel (R) i5-4590, 3.30 GHZ CPU, 4GB RAM over Windows 7 system, testing 50, 100, 150, 200, 250, 300, 350 and 400 interest requests respectively by 100 times simulations. For these parameters, we make simulations under different settings to find the proper one, i.e., $m = 10$, $\varpi = 256B$, $\theta = 2.5$, $\tau_0^{\lambda} = 2$, $\chi = 8$ and $I_{\max} = 100$.

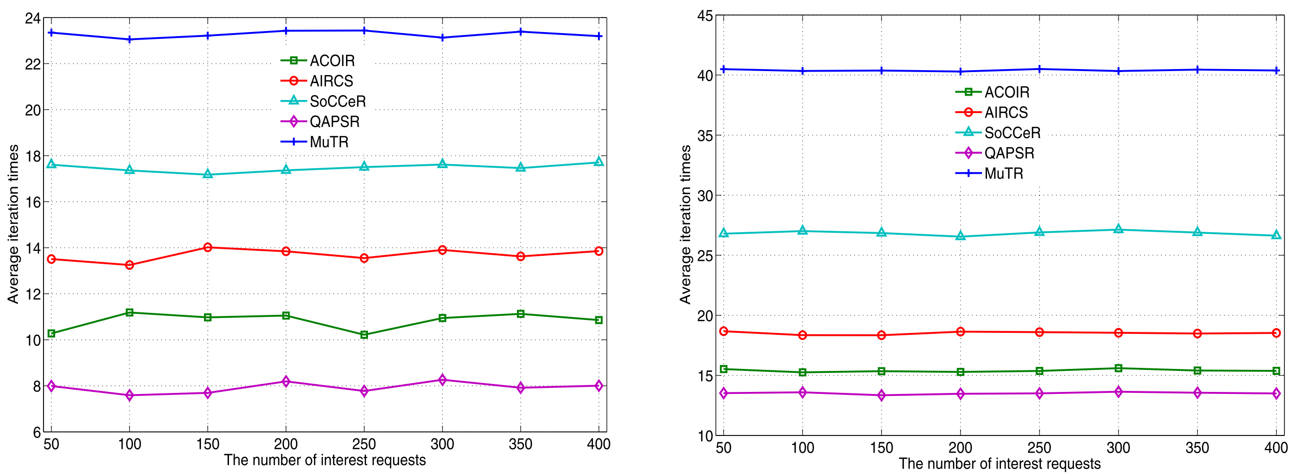
5.2. Average routing success rate

$\mathbb{ARSR}$s for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 8. We observe that ACOIR, AIRCS and MuTR have the highest $\mathbb{ARSR}$, followed by SoCCeR and QAPSR. In particular, $\mathbb{ARSR}$s of ACOIR, AIRCS and MuTR can reach 100% while those of SoCCeR and QAPSR only remain in (94%, 95%) and (91%, 92%) respectively. For ACOIR and AIRCS, when and only when TTLs of all inp-ants are over within I iterations, the content cannot be retrieved; however, its corresponding probability approaches 0, thus the content can be retrieved always (see Theorem 3). When one CR receives inp-ant, MuTR forwards inp-ant via all outgoing interfaces; as a result, the content can be retrieved always. Instead, both SoCCeR and QAPSR forward inp-ant via outgoing interface with the highest forwarding probability. Obviously, SoCCeR and QAPSR do not make full use of diversity feature of inp-ant, thus their $\mathbb{ARSR}$s are reduced. However, SoCCeR randomly selects an outgoing interface to forward



(a) # over NSFNET (b) # over Deltacom

Fig. 8. Average routing success rate for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.



(a) # over NSFNET (b) # over Deltacom

Fig. 9. Average iteration times for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.

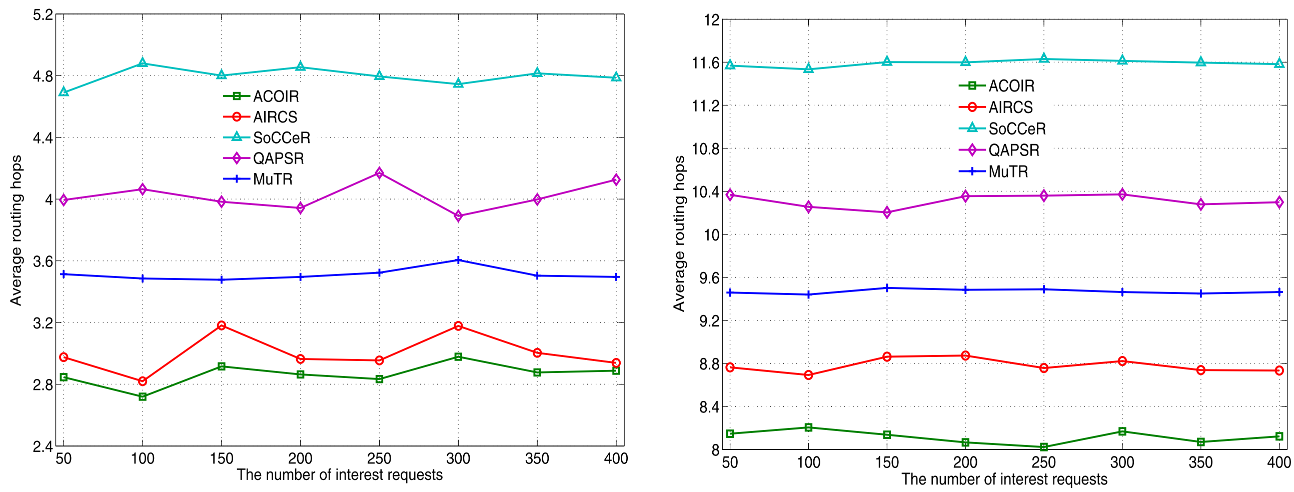
inp-ant, which avoids being trapped into the local optimum. Thus, SoCCeR has higher $\mathbb{ARSR}$ than QAPSR.

5.3. Average iteration times

$\mathbb{AIT}$s for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 9. We observe that QAPSR has the smallest $\mathbb{AIT}$, followed by ACOIR, AIRCS, SoCCeR and MuTR. In fact, $\mathbb{AIT}$ mainly depends on positive feedback feature of inp-ant, and strong positive feedback means small $\mathbb{AIT}$. In terms of these five methods, QAPSR always forwards inp-ant via outgoing interface with the highest forwarding probability, and thus has the strongest positive feedback. MuTR always forwards inp-ant via all outgoing interfaces based on probabilistic forwarding without considering positive feedback feature of inp-ant, and thus has the weakest positive feedback. Furthermore, ACOIR and AIRCS forwards more inp-ants via outgoing interface with higher forwarding probability while SoCCeR forwards some inp-ants based on probabilistic forwarding, thus ACOIR and AIRCS has stronger positive feedback than SoCCeR. In conclusion, QAPSR shows the strongest positive feedback feature, followed by ACOIR/AIRCS, SoCCeR and MuTR. For ACOIR and AIRCS, if a certain amount of inp-ants draw around one CR is larger than half of the number of inp-ants, ACOIR is finished while AIRCS is not finished. Thus, ACOIR has smaller $\mathbb{AIT}$ than AIRCS.

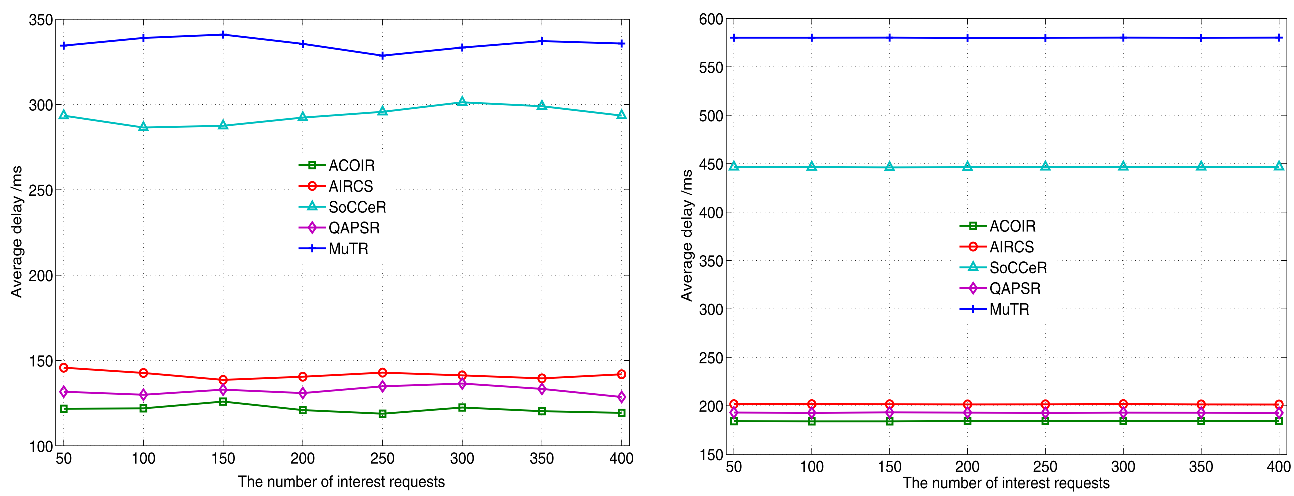
5.4. Average routing hops

$\mathbb{ARH}$s for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 10. We observe that ACOIR has the smallest $\mathbb{ARH}$, followed by AIRCS, MuTR, QAPSR and SoCCeR. Among them, ACOIR, AIRCS and MuTR forward inp-ant via all outgoing interface while SoCCeR and QAPSR forward inp-ant via outgoing interface with the highest forwarding probability, which leads to ACOIR, AIRCS and MuTR being much easier to find the closest content copy than SoCCeR and QAPSR. Thus, ACOIR, AIRCS and MuTR have smaller $\mathbb{ARH}$ than SoCCeR and QAPSR. For ACOIR, AIRCS and MuTR, ACOIR and AIRCS can always retrieve the closest content copy (see Theorem 4) with the help of diversity feature and positive feedback feature of inp-ant; however, MuTR only uses probabilistic forwarding strategy regardless of positive feedback feature of inp-ant. Thus, MuTR has larger $\mathbb{ARH}$ than ACOIR and AIRCS. For ACOIR and AIRCS, ACOIR checks the number of inp-ants drew around each CR, which promotes the routing process and thus has smaller $\mathbb{ARH}$ than AIRCS. For SoCCeR and QAPSR, QAPSR forwards all inp-ants via outgoing interface with the highest forwarding probability, which shows strong positive feedback feature of inp-ant; however, SoCCeR considers the probability-based forwarding to some extent. Thus, QAPSR has smaller $\mathbb{ARH}$ than SoCCeR.



(a) # over NSFNET (b) # over Deltacom

Fig. 10. Average routing hops for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.



(a) # over NSFNET (b) # over Deltacom

Fig. 11. Average delay for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.

5.5. Average routing delay

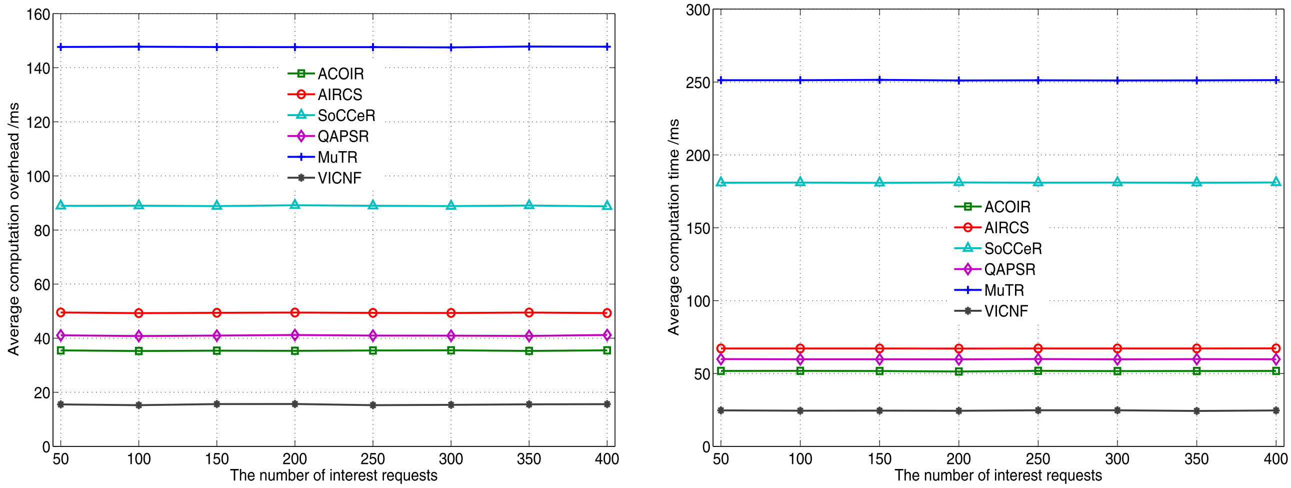
$\mathbb{ARD}$s for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 11. We observe that ACOIR has the smallest $\mathbb{ARD}$, followed by QAPSR, AIRCS, SoCCeR and MuTR. $\mathbb{ARD}$ mainly depends on $\mathbb{AIT}$, $\mathbb{ARH}$ and lookup. (i) In fact, during the forwarding process of inp-ant, the time consumed by lookup occupies large proportion. ACOIR uses a content management strategy based on NPT, which reduces lookup time effectively and thus decreases $\mathbb{ARD}$ to a great extent. Besides, ACOIR has the smallest $\mathbb{ARH}$ and the second smallest $\mathbb{AIT}$. Taken together, ACOIR has the smallest $\mathbb{ARD}$. (ii) For MuTR, it has far larger $\mathbb{AIT}$ than ACOIR, QAPSR, AIRCS and SoCCeR, and has smaller but close $\mathbb{ARH}$ to QAPSR and SoCCeR. Besides, MuTR forwards inp-ant via all outgoing interfaces, which causes huge network load and goes against the forwarding of inp-ant, and thus largely increases transmission delay. Taken together, MuTR has the largest $\mathbb{ARD}$. (iii) For AIRCS and SoCCeR, they do not design the distinct strategy to decrease $\mathbb{ARD}$. However, AIRCS has smaller $\mathbb{AIT}$ and $\mathbb{ARH}$ than SoCCeR, thus it has smaller $\mathbb{ARD}$ than SoCCeR. (iv) For QAPSR and AIRCS, although QAPSR has larger $\mathbb{ARH}$ than AIRCS, but their $\mathbb{ARH}$s are close. Besides, QAPSR has far smaller $\mathbb{AIT}$ than AIRCS. In addition, QAPSR does not search PIT when one CR receives inp-ant, which also saves much time. Taken together, QAPSR has smaller $\mathbb{ARD}$ than AIRCS.

5.6. Average time overhead

The total time overhead mainly consists of computation overhead at CRs and messaging overhead in network.

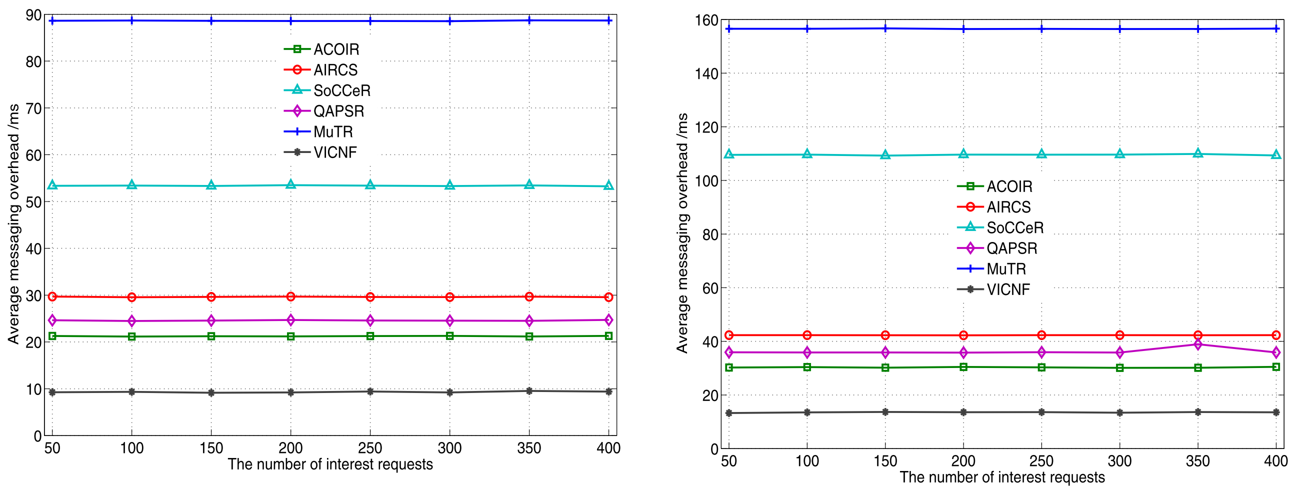
5.6.1. Average computation overhead

Average computation overheads at CRs for ACOIR, AIRCS, SoCCeR, QAPSR, MuTR and Vanilla ICN Forwarding scheme (VICNF) are reported in Fig. 12, where the computation overhead is composed of content lookup of CS, content concentration perception of PIT and forwarding decision of FIB. We observe that VICNF has the smallest average computation overhead, followed by ACOIR, QAPSR, AIRCS, SoCCeR and MuTR, because VICNF uses the deterministic forwarding strategy to forward inp via the so-called best outgoing interface while the other methods uses probabilistic forwarding strategy. It is obvious that the introduction of ACO increases processing time at each CR and further increases routing delay compared to VICNF. In spite of this, VICNF neither addresses ICN routing issue intelligently nor guarantees to retrieve the most suitable



(a) # over NSFNET (b) # over Deltacom

Fig. 12. Average computation overhead at CRs for ACOIR, AIRCS, SoCCeR, QAPSR, MuTR and VNDNF over NSFNET and Deltacom.



(a) # over NSFNET (b) # over Deltacom

Fig. 13. Average messaging overhead for ACOIR, AIRCS, SoCCeR, QAPSR, MuTR and VNDNF over NSFNET and Deltacom.

content copy in a cooperative and self-organizing way. In fact, the related reasons on why the order about average computation overhead at CRs is ACOIR, QAPSR, AIRCS, SoCCeR and MuTR are similar to Section 5.5.

5.6.2. Average messaging overhead

Average messaging overheads in network for ACOIR, AIRCS, SoCCeR, QAPSR, MuTR and VICNF are reported in Fig. 13, where the messaging overhead is generated from that interest requester successively sends inp-ants to the network, and this can cause different degrees on network congestion according to different network status. We observe that VICNF has the smallest average messaging overhead, followed by ACOIR, QAPSR, AIRCS, SoCCeR and MuTR, because VICNF sends an inp while the other methods send a group of inp-ants with small size. From this perspective, the messaging overhead brought by ACO increases routing delay compared to VICNF. However, in fact, the purpose of using ACO is to retrieve the most suitable content copy in the environment of cooperation and self-organization while having small routing delay as much as possible. Due to different messaging and forwarding conditions, network congestion conditions in terms of ACOIR, QAPSR, AIRCS, SoCCeR and MuTR are different. MuTR sends inp-ants to all CRs, which means flooding and thus has the largest average messaging overhead. ACOIR only sends inp-ants to some CRs according to content concentration and it uses trade-off between positive feedback feature and diversity feature, which eases network congestion effectively and thus has the smallest average messaging overhead. Similarly, the results analysis on QAPSR, AIRCS, SoCCeR can be found from Sections 5.3, 5.4 and 5.5.

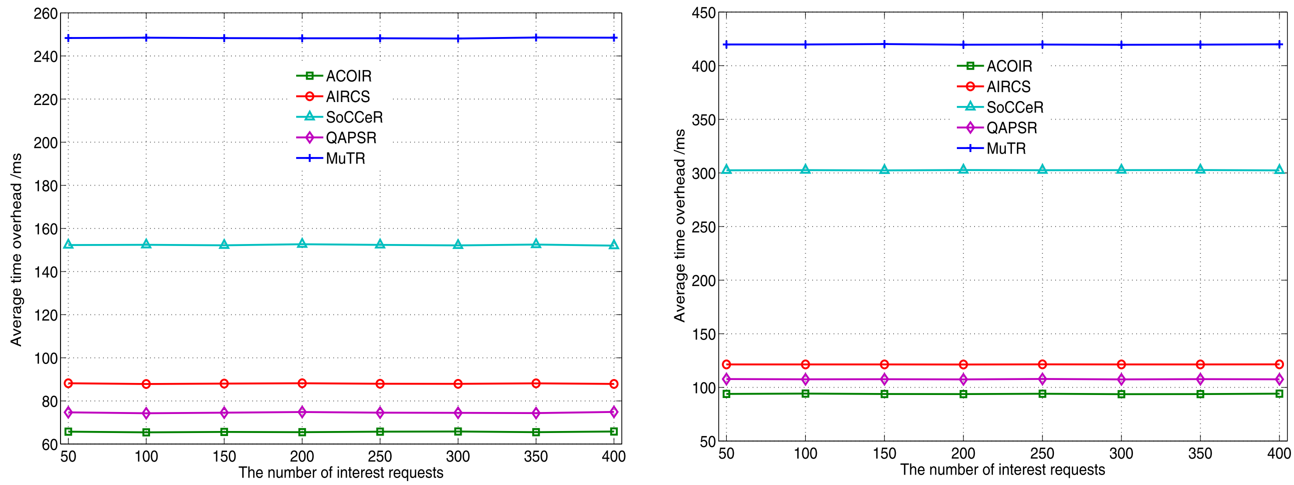
5.6.3. Total average time overhead

$\mathbb{ATO}$s for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 14. We observe that ACOIR has the smallest $\mathbb{ATO}$, followed by QAPSR, AIRCS, SoCCeR and MuTR. In order to avoid lots of repetition on the results analysis, the related reasons are not presented and they can be found from Sections 5.6.1 and 5.6.2.

5.7. Average load balance degree

We use variation coefficient to describe load balance degree, and it is defined as follows.

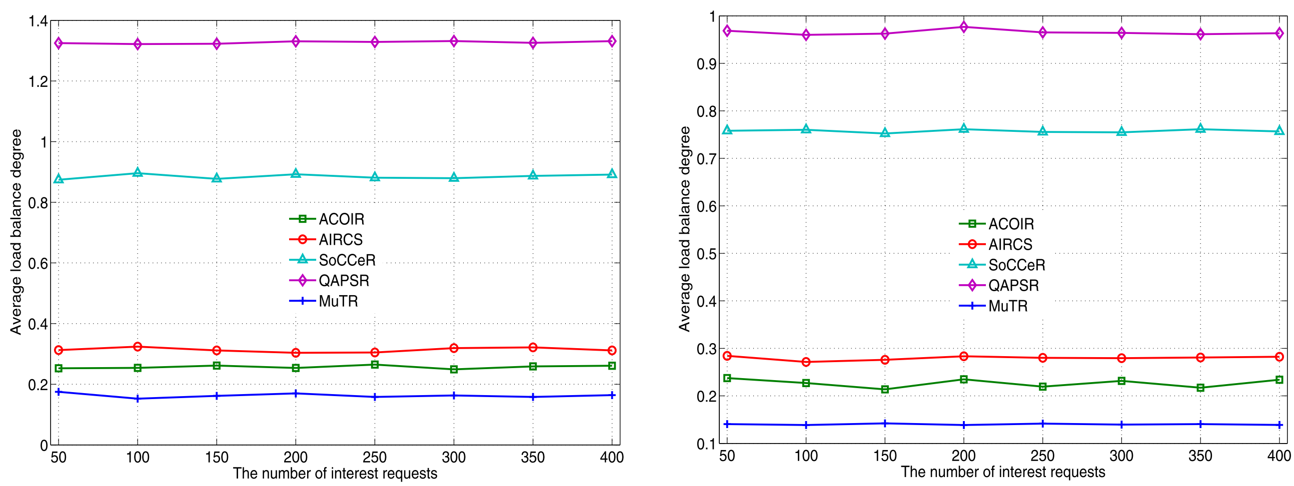
$$\rho = \frac{\sqrt{\sum_{i=1}^{Q}\left((r_i - r_{ave})^2/Q\right)}}{r_{ave}}, \tag{45}$$



(a) # over NSFNET

(b) # over Deltacom

Fig. 14. Total average time overhead for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.



(a) # over NSFNET

(b) # over Deltacom

Fig. 15. Average load balance degree for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR over NSFNET and Deltacom.

$$r_{ave} = \frac{1}{Q}\sum_{i=1}^{Q} r_i, \tag{46}$$

$$r_i = \frac{b(i)_{occ}}{b(i)_{tot}}. \tag{47}$$

Among them, ρ is load balance degree, Q is the total number of links, $b(i)_{occ}$ is the occupied bandwidth of link i, and $b(i)_{tot}$ is the total bandwidth of link i. The low load balance degree indicates good performance.

ALBDs for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR are reported in Fig. 15. We observe that MuTR has the lowest ALBD, followed by ACOIR, AIRCS, SoCCeR and QAPSR. (i) QAPSR always forwards inp-ant via outgoing interface with the highest forwarding probability, which makes the corresponding link heavy-loaded and the other links light-loaded. Thus, QAPSR has the highest ALBD. (ii) SoCCeR uses probabilistic forwarding strategy to some extent, which shares the total load from the link with the highest forwarding probability to these links without it. Thus, SoCCeR has lower ALBD than QAPSR. (iii) Both ACOIR and AIRCS always forward inp-ant based on probabilistic forwarding strategy, and the outgoing interface with higher forwarding probability has more inp-ants to be forwarded. Thus, both ACOIR and AIRCS have lower ALBD than SoCCeR. (iv) For ACOIR, it reflects much stronger diversity feature of inp-ant than AIRCS, which makes more links in ACOIR share load and thus has lower ALBD than AIRCS. (v) However, as routing on, some links with higher forwarding probability have more and more inp-ants while those with lower forwarding probability have fewer and fewer inp-ants, and then the corresponding ALBD becomes higher. Thus, ACOIR has higher ALBD than MuTR.

5.8. Comprehensive performance

The comprehensive performance depends on ARSR, AIT, ARH, ARD, ATO and ALBD, and the mathematical relationship among ACOIR, AIRCS, SoCCeR, QAPSR and MuTR is shown in Table 2. In terms of one metric, a method with good performance is ranked as small number, and the rank begins from 1 with the increment of 1 (*e.g.*, ARH of ACOIR is ranked as 1, while that of SoCCeR is ranked as 5). If some methods have the same performance, their corresponding ranks are same (*e.g.*, ARSRs of ACOIR, AIRCS and MuTR are ranked as 1). According to Table 2, we can get the ranking results of these five methods regarding six metrics, as shown in Table 3.

The Wilcoxon-based testing according to the results in Table 3 is used to evaluate the comprehensive performance, here the level

Table 2
The mathematical relationship among ACOIR, AIRCS, SoCCeR, QAPSR and MuTR.

Metric	Mathematical relationship	Note
$\mathbb{ARSR}$	ACOIR = AIRCS = MuTR > SoCCeR > QAPSR	The higher the better
$\mathbb{AIT}$	QAPSR < ACOIR < AIRCS < SoCCeR < MuTR	The smaller the better
$\mathbb{ARH}$	ACOIR < AIRCS < MuTR < QAPSR < SoCCeR	The smaller the better
$\mathbb{ARD}$	ACOIR < QAPSR < AIRCS < SoCCeR < MuTR	The smaller the better
$\mathbb{ATO}$	ACOIR < QAPSR < AIRCS < SoCCeR < MuTR	The smaller the better
$\mathbb{ALBD}$	MuTR < ACOIR < AIRCS < SoCCeR < QAPSR	The lower the better

Table 3
The ranking results for ACOIR, AIRCS, SoCCeR, QAPSR and MuTR.

Algorithm	$\mathbb{ARSR}$	$\mathbb{AIT}$	$\mathbb{ARH}$	$\mathbb{ARD}$	$\mathbb{ATO}$	$\mathbb{ALBD}$
ACOIR	1	2	1	1	1	2
AIRCS	1	3	2	3	3	3
SoCCeR	4	4	5	4	4	4
QAPSR	5	1	4	2	2	5
MuTR	1	5	3	5	5	1

Table 4
Wilcoxon-based testing results.

Comparison	R^0	R^+	R^-	p-value
AIRCS versus ACOIR	1	5	0	0.038
SoCCeR versus ACOIR	0	6	0	0.026
QAPSR versus ACOIR	0	5	1	0.071
MuTR versus ACOIR	1	4	1	0.078

of significance is set as 0.1, and the testing results are shown in Table 4. We observe that all p-values are smaller than 0.1, that is, ACOIR has the best comprehensive performance. In fact, ACOIR integrates CS, PIT and FIB comprehensively; however, AIRCS, QAPSR, MuTR and SoCCeR only have the corresponding forwarding strategies, i.e., considering FIB regardless of CS and PIT. Therefore, ACOIR has better comprehensive performance than them. Furthermore, ACOIR has the best comprehensive performance over both NSFNET and Deltacom (i.e., two different scale network topologies), which indicates that ACOIR has good scalability. Moreover, ACOIR has same $\mathbb{ARSR}$ over NSFNET and Deltacom, which indicates that ACOIR has feasibility and good stability.

6. Conclusions

Although ICN is a promising networking paradigm, its routing issues are increasingly severe. The bio-inspired solution can effectively solve the above ICN routing issues. Given this consideration, we introduce ACO into ICN and propose an ACO-inspired ICN routing mechanism, called ACOIR. Meanwhile, we propose a content management strategy, a continuous model of content concentration and a computation scheme of forwarding probability. In particular, we present some theoretical analysis to prove the feasibility of ACOIR. In addition, we evaluate the performance of ACOIR by considering routing success rate, iteration times, routing hops, delay, time overhead and load balance. Simulation results have demonstrated that ACOIR has good performance.

However, as a novel bio-inspired ICN routing mechanism, ACOIR also has some limitations, which need to be enhanced and improved. On one hand, the devised content management strategy for CS does not take ant features into adequate consideration. On the other hand, this paper does not leverage the feature of which different ants have different work abilities to retrieve different content types for different kinds of inps. In future, we will improve ACOIR to overcome the above mentioned limitations.

Acknowledgments

We would like to thank the editors and all anonymous reviewers for helpful suggestions, which have considerably improved and enhanced the quality of this paper. This work is supported by the National Natural Science Foundation of China under Grant No. 61572123 and the National Science Foundation for Distinguished Young Scholars of China under Grant No. 71325002.

Appendix A. Theorem 1 and Corollary 1 Proof.

Proof. Theorem 1.

- i) Consider the best condition where N content items have the same content names, that is, $\forall\ 1 \le p,\ q \le N,\ cn_p = cn_q$. Then, the minimal lookup times (denoted by $ln1_{\min}$) is the shortest content name length (denoted by $h_{\min}$), and we have

$$ln1_{min} = h_{min} = \min_{p=1}^{N} l_p. \tag{48}$$

- ii) Consider the worst condition where N content items have different content names, that is, $\forall\ 1 \le p,\ q \le N,\ cn_p \ne cn_q$, here, $s_{p,\,1} \ne s_{q,\,1}$; in other words, the root node in NPT has N subtrees and the height of subtree p is $l_p - 1$. At first, N subtrees are searched in sequential way, and the correspondingly maximal lookup times is N. Then, only one subtree is searched, and the correspondingly maximal lookup times is $h_{\max} - 1$. Thus, the totally maximal lookup times (denoted by $ln1_{\max}$) is defined as follows.

$$ln1_{max} = N + h_{max} - 1. \tag{49}$$

Consider $h_{\max} \ll N$, and the time complexity of Algorithm 1 is $O(N)$. To sum up i) and ii), Theorem 1 is proved. □

Proof. Corollary 1. Let $ln2_{\max}$ and $ln2_{\min}$ represent the maximal and minimal lookup times with M2 respectively, and we have

$$ln2_{max} = \sum_{p=1}^{N} l_p = N + \sum_{p=1}^{N} \left(l_p - 1\right) > N + h_{max} - 1 = ln1_{max}, \tag{50}$$

$$ln2_{min} = \min_{p=1}^{N} l_p = ln1_{min}. \tag{51}$$

According to (50) and (51), M1 has smaller maximal lookup times than M2 and the same minimal lookup times as M2, thus Corollary 1 is proved. □

Appendix B. Theorem 2 Proof

Proof. Theorem 2. The computation time of Algorithm 2 depends on content lookup of CS, content concentration perception of PIT and forwarding decision of FIB in terms of an inp-ant within one iteration. According to Theorem 1, content lookup of CS runs in $O(n \cdot N)$. Suppose that perception times and forwarding times are denoted by Np and Nf respectively, and we have

$$Np = Nf = 2e \le 2n(n-1), \tag{52}$$

where e is the number of edges in network topology.

Then, the time complexity of an inp-ant within one iteration depends on

$$O(n \cdot N) + O(2e) + O(2e) = O(n \cdot N + 4e) = O(n \cdot N + n^2). \tag{53}$$

In terms of m inp-ants and I iterations, the time complexity is $O\big(I \cdot m(n \cdot N + n^2)\big)$. To sum up, Theorem 2 is proved.

Especially when $N = 1$, Algorithm 2 runs in $O(I \cdot m \cdot n^2)$. □

Appendix C. Theorem 3 Proof

Proof. Theorem 3.

- i) $\forall\ ia_\lambda$, if its requested content name is found, the routing process of ia_λ is finished. In terms of this condition, the content must be found, i.e., $P(X) = 1$.
- ii) $\forall\ ia_\lambda$, if $Fw_i^\lambda = \Phi$, it means that all CRs have been traversed. When the content exists, the content must be found, i.e., $P(X) = 1$.
- iii) For all ia_λ and I, let $\overline{X}$ represent the event which TTL is over before the content is found, since the operation of each inp-ant is independent, and we have

$$P(\overline{X}) = \left(\frac{1}{3}\right)^{m\cdot I}. \tag{54}$$

Furthermore,

$$P(X) = 1 - P(\overline{X}) = 1 - \left(\frac{1}{3}\right)^{m\cdot I} \geq 1 - \varepsilon. \tag{55}$$

Consider $I \to +\infty$, and we have

$$\lim_{I\to+\infty} P(X) = 1 - \lim_{I\to+\infty} \left(\frac{1}{3}\right)^{m\cdot I} = 1. \tag{56}$$

To sum up i), ii) and iii), Theorem 3 is proved. □

Appendix D. Theorem 4 Proof

Proof. Theorem 4. The content can be retrieved with the probability approaching 1 according to Theorem 3. Based on this, we first discuss any two paths between interest requester and content provider(s), denoted by $\mathbf{P}'$ and $\mathbf{P}''$. Their total distances are denoted by L' and L'' respectively. The numbers of their passed inp-ants are denoted by m' and m'' respectively. We use the classified discussion method to prove Theorem 4. Then, let $L' < L''$.

- i) $m' > m''$
  Consider $I = 2$. The content concentration over $\mathbf{P}'$ is higher than that over $\mathbf{P}''$ according to (20) since $m' > m''$ and $L' < L''$. As a result, m' begins to increase while m'' begins to decrease. Then, when I reaches a certain level, $\mathbf{P}''$ with lower content concentration cannot be regarded as the optimal solution (s^*). In this case, $\mathbf{P}'$ with higher content concentration is possible to be regarded as the optimal solution.
  Consider $I > 2$. Let any two paths generalize to all paths, and each path is regarded as a solution. When s^* is found at the first time, its corresponding content concentration over the path will become higher and higher. Therefore, in this case, $P^*(I) = 1$.
- ii) $m' = m''$
  The proof is similar to the condition where $m' > m''$. In this case, $P^*(I) = 1$.
- iii) $m' < m''$
  We use the analytical method to prove this case.

Let $e'_{1,2}$ and $e''_{1,2}$ represent the first edge in terms of $\mathbf{P}'$ and $\mathbf{P}''$ respectively. Suppose that m' and m'' inp-ants spend t' and t'' accomplishing the first iteration respectively, and we have

$$TCC'_{1,2}(t,1) = \sum_{\lambda=1}^{m'} \frac{L'}{cc_{1,2}^\lambda(t)}, \tag{57}$$

$$TCC''_{1,2}(t,1) = \sum_{\lambda=1}^{m''} \frac{L''}{cc_{1,2}^\lambda(t)}, \tag{58}$$

$$t = max\left\{t', t''\right\}, \tag{59}$$

where $TCC'_{1,2}(t,1)$ and $TCC''_{1,2}(t,1)$ are the total content concentration over $e'_{1,2}$ and $e''_{1,2}$ after the first iteration respectively.

If $\mathbf{P}'$ wants to own higher content concentration than $\mathbf{P}''$, the following inequality (60) has to hold.

$$\frac{TCC'_{1,2}(t,1)}{TCC''_{1,2}(t,1)} < \frac{d''_{1,2}}{d'_{1,2}} = \frac{m'}{m''}, \tag{60}$$

where $d'_{1,2}$ and $d''_{1,2}$ are the distances of $e'_{1,2}$ and $e''_{1,2}$ respectively.

Putting (57) and (58) into (60), and we have

$$\frac{L'}{cc'_{1,2}(t)} < \frac{L''}{cc''_{1,2}(t)}, \tag{61}$$

where

$$cc'_{1,2}(t) = \frac{v_2\cdot\tau_0^\lambda}{\theta}\cdot e^{-\theta\left(t-\frac{\varpi}{v_1}\right)}\left(e^{\frac{\theta\cdot d'_{1,2}}{v_2}} - 1\right), \tag{62}$$

$$cc''_{1,2}(t) = \frac{v_2\cdot\tau_0^\lambda}{\theta}\cdot e^{-\theta\left(t-\frac{\varpi}{v_1}\right)}\left(e^{\frac{\theta\cdot d''_{1,2}}{v_2}} - 1\right). \tag{63}$$

Furthermore, (61) is changed to

$$\frac{e^{\frac{\theta\cdot d''_{1,2}}{v_2}} - 1}{e^{\frac{\theta\cdot d'_{1,2}}{v_2}} - 1} < \frac{L''}{L'}. \tag{64}$$

Since $m' < m''$, according to (29) and (36), we have

$$d''_{1,2} < d'_{1,2}. \tag{65}$$

Putting (65) into (64), and we have

$$\frac{e^{\frac{\theta\cdot d''_{1,2}}{v_2}} - 1}{e^{\frac{\theta\cdot d'_{1,2}}{v_2}} - 1} < e^{\frac{\theta}{v_2}\left(d''_{1,2}-d'_{1,2}\right)} < 1 < \frac{L''}{L'}. \tag{66}$$

It is obvious that (66) always holds. As I goes on, m' begins to increase, and m'' begins to decrease. After a certain number of iterations, we have $m' > m''$. The following proof is similar to the condition where $m' > m''$. Therefore, in this case, $P^*(I) = 1$.

Let $L' = L''$ and $L' > L''$, and their proof is similar to the condition where $L' < L''$.

In conclusion,

$$\lim_{I\to+\infty} P^*(I) = 1. \tag{67}$$

In summary, Theorem 4 is proved.

In fact, Theorem 4 indicates that the shortest path to which s^* corresponds has the highest content concentration. □

References

[1] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, G.C. Polyzos, A survey of information-centric networking research, IEEE Commun. Surv. Tutor. 16 (2) (2014) 1024–1048.

[2] M. Lee, J. Song, K. Cho, S. Pack, T.T. Kwon, J. Kangasharju, Y. Choi, Content discovery for information-centric networking, Comput. Netw. 83 (2015) 1–14.

[3] X. Wang, H. Cheng, M. Huang, QoS multicast routing protocol oriented to cognitive network using competitive coevolutionary algorithm, Expert Systems with Applications. 41 (10) (2014) 4513–4528.

[4] X. Wang, H. Cheng, K. Li, J. Li, J. Sun, A cross-layer optimization based integrated routing and grooming algorithm for green multi-granularity transport networks, J. Parallel Distrib. Comput. 73 (6) (2013) 807–822.

[5] X. Wang, H. Cheng, M. Huang, Multi-robot navigation based QoS routing in self-organizing networks, Eng. Appl. Artif. Intell. 26 (1) (2013) 262–272.

[6] J. Lv, X. Wang, M. Huang, Ant colony optimization inspired ICN routing with content concentration and similarity relation, IEEE Commun. Lett. 21 (6) (2017) 1313–1316.
[7] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybernetics 26 (1) (1996) 29–41.
[8] F. Loukos, H.D. Karatza, C.X. Mavromoustakis, An ant intelligence inspired routing scheme for peer-to-peer networks, Simul. Modell. Pract. Theory 19 (2) (2011) 649–661.
[9] H. Hsin, E. Chang, K. Su, A. Wu, Ant colony optimization-based adaptive network-on-chip routing framework using network information region, IEEE Trans. Comput. 64 (8) (2015) 2119–2130.
[10] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, Z. Tang, Ant colony optimization with clustering for solving the dynamic location routing problem, Appl. Math. Comput. 285 (2016) 149–173.
[11] G.S. Pavani, A. Queiroz, J.C. Pellegrini, Analysis of ant colony optimization-based routing in optical networks in the presence of byzantine failures, Inf. Sci. 340–341 (2016) 27–40.
[12] Y. Kim, E. Lee, H. Park, J. Choi, H. Park, Ant colony based self-adaptive energy saving routing for energy efficient internet, Comput. Netw. 56 (2012) 2343–2354.
[13] M. Perez, S. Herreria-Alonso, M. Fernandez-Veiga, C. Lopez-Garcia, An ant colonization routing algorithm to minimize network power consumption, J. Netw. Comput. Appl. 58 (2015) 217–226.
[14] S. Chatterjee, S. Das, Ant colony optimization based enhanced dynamic source routing algorithm for mobile ad-hoc network, Inf. Sci. 295 (2015) 67–90.
[15] S. Miyashita, Y. Li, An ant-based routing algorithm with multi-phase pheromone and power-saving in mobile ad hoc networks, in: International Symposium on Computing and Networking, 2015, pp. 154–160.
[16] S. Shanbhag, N. Schwan, I. Rimac, M. Varvello, SoCCer: services over content-centric routing, in: ACM SIGCOMM (ICN), 2011, pp. 62–67.
[17] P. Hu, J. Chen, Improved CCN routing based on the combination of genetic algorithm and ant colony optimization, in: International Conference on Computer Science and Network Technology, 2013, pp. 846–849.
[18] A.Z. Khan, S. Baqai, F.R. Dogar, Qos aware path selection in content centric networks, in: IEEE International Conference on Communications, 2012, pp. 2645–2649.
[19] J. Eymann, A. Timm-Giel, Multipath transmission in content centric networking using a probabilistic ant-routing mechanism, in: International Conference on Mobile Networks and Management, 2013, pp. 45–56.
[20] C. Li, W. Liu, L. Wang, M. Li, K. Okamura, Energy-efficient quality of service aware forwarding scheme for content-centric networking, J. Netw. Comput. 58 (2015) 241–254.
[21] Q. Huang, F. Luo, Ant-colony optimization based qos routing in named data networking, J. Comput. Methods Sci. Eng. 16 (3) (2016) 1–12.
[22] H. Qian, R. Ravindran, G. Wang, D. Medhi, Probability-based adaptive forwarding strategy in named data networking, in: IFIP/IEEE International Symposium on Integrated Network Management, 2013, pp. 1094–1101.
[23] S. Manome, T. Asaka, Routing for content oriented networks using dynamic ant colony optimization, in: Asia-Pacific Symposium on Network Operations and Management, 2015, pp. 209–214.
[24] J. Mun, H. Lim, New approach for efficient ip address lookup using a bloom filter in trie-based algorithms, IEEE Trans. Comput. 65 (5) (2016) 1558–1565.
[25] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, A survey of naming and routing in information-centric networks, IEEE Commun. Mag. 50 (12) (2012) 44–53.
[26] J. Lv, X. Wang, M. Huang, ACO-inspired ICN routing mechanism with mobility support, Appl. Soft Comput. 58 (2017) 427–440.
[27] Deltacom, 2017, http://www.topology-zoo.org/ [accessed 20.07.17].

Jianhui Lv 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. degree in computer science from the Northeastern University, Shenyang, China in 2014. He is currently working toward the Ph.D. degree in the Northeastern University, Shenyang, China. His research interests include ICN routing and operational research, etc. He has published more than 15 journal and conference papers.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1989, 1992, and 1998 respectively. He is currently a Professor at the College of Software, Northeastern University, Shenyang, China. His research interests include future Internet and cloud computing, etc. He has published more than 100 journal papers, books and book chapters, and refereed conference papers. He has received several best paper awards.

Kexin Ren received the B.S. degree in network engineering from the Harbin University of Science and Technology, Harbin, China in 2015. She is currently working toward the M.S. degree in the Northeastern University, Shenyang, China. Her research interests include ICN routing.

Min Huang received the B.S. degree in automatic instrument, the M.S. degree in systems engineering, and Ph.D. degree in control theory from the Northeastern University, Shenyang, China in 1990, 1993, and 1999 respectively. She is currently a Professor at the College of Information Science and Engineering, Northeastern University, Shenyang, China. Her research interests include modeling and optimization for logistics and supply chain system, etc. She has published more than 100 journal papers, books, and refereed conference papers.

Keqin Li is a SUNY Distinguished Professor of computer science. His research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published over 390 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Journal of Parallel and Distributed Computing. He is an IEEE Fellow.