Research Article

Location Prediction-Based Data Dissemination Using Swarm Intelligence in Opportunistic Cognitive Networks

Jie Li,^{1,2} Xingwei Wang,³ Jie Jia,³ Pengfei Wang,⁴ Yan Zhou,⁴ and Zhijie Zhao⁵

¹ Computing Center, Northeastern University, Shenyang 110819, China

² Key Laboratory of Networked Control System, The Chinese Academy of Sciences, Shenyang 110016, China

³ College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

⁴ Software College, Northeastern University, Shenyang 110819, China

⁵ Information and Technology Center of China Mobile Group Liaoning Co., Ltd., Liaoning 110179, China

Correspondence should be addressed to Jie Li; lijie@mail.neu.edu.cn

Received 8 June 2014; Revised 8 August 2014; Accepted 11 August 2014; Published 25 September 2014

Academic Editor: Baozhen Yao

Copyright © 2014 Jie Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Swarm intelligence is widely used in the application of communication networks. In this paper we adopt a biologically inspired strategy to investigate the data dissemination problem in the opportunistic cognitive networks (OCNs). We model the system as a centralized and distributed hybrid system including a location prediction server and a pervasive environment deploying the large-scale human-centric devices. To exploit such environment, data gathering and dissemination are fundamentally based on the contact opportunities. To tackle the lack of contemporaneous end-to-end connectivity in opportunistic networks, we apply ant colony optimization as a cognitive heuristic technology to formulate a self-adaptive dissemination-based routing scheme in opportunistic cognitive networks. This routing strategy has attempted to find the most appropriate nodes conveying messages to the destination node based on the location prediction information and intimacy between nodes, which uses the online unsupervised learning on geographical locations and the biologically inspired algorithm on the relationship of nodes to estimate the delivery probability. Extensive simulation is carried out on the real-world traces to evaluate the accuracy of the location prediction and the proposed scheme in terms of transmission cost, delivery ratio, average hops, and delivery latency, which achieves better routing performances compared to the typical routing schemes in OCNs.

1. Introduction

Cognitive networks [1] have already been prototyped for many commercial and civilian applications. Combined with the social intelligence, cognitive networks promise to support services like citizen journalism, mobile social networking, environmental monitoring, and traffic monitoring by integrating ubiquitous sensing, large-scale data collection, and cloud computing. Opportunistic network [2] provides an ideal solution for promoting the evolution of communication among human beings and machines. Cognitive network technology can be applied to the communication system of opportunistic networks to provide heuristic schemes on the algorithm design and system implementation. Smart handheld devices (e.g., smart phone or PDA) carried by

a large number of participants contribute to opportunistic cognitive networks with their sensing and communication parts [3, 4].

Moreover, by including people in the loop, it is now possible to design applications that can dramatically improve daily lives of individuals and communities. The inherent mobility of participants provides unprecedented spatiotemporal coverage and also makes it possible to observe various events. CarTel [5] is a mobile sensor computing system designed to collect, process, deliver, and visualize data from sensors located on mobile units such as automobiles. PEIR [6] is an application that uses location data sampled from everyday mobile phones to calculate personalized estimates of environmental impact and exposure.

The opportunistic cognitive network is a kind of delay tolerant networks (DTNs), and it utilizes communication opportunities obtained from node movement to relay packets. Mobile nodes are enabled to communicate with each other even if a route connecting them never exists. The basic routing strategy of opportunistic networks is “store-carry-forward.” Messages are routed between the sender and the destination(s), and any possible node can be used as a relay, provided that it is likely to bring the message closer to the final destination. Message transmission is mainly dependent on intermediate relays, so relay node selection is critical for efficient data dissemination in opportunistic cognitive networks.

Since human mobility is mostly unpredictable, conventional routing algorithms for mobile ad hoc networks (MANETs) no longer perform well in the opportunistic cognitive networks. Therefore, new algorithms are required to overcome intermittent connectivity in the opportunistic networks.

Stochastic routing protocols, such as Epidemic [7], First Contact (FC) [8], and Direct Delivery (DD) [9], broadcast messages to any encountered node in order to increase the delivery ratio. Epidemic routing diffuses messages similar to the way viruses or bacteria spread in biology. Whenever encountering another node, a node replicates and transfers messages. After receiving messages, a node will move to other places and continuously replicate and deliver the messages to other encountered ones. First Contact sends messages to the first encountered node without copying the messages to other nodes. Direct Delivery comprises the trade-off between Epidemic and First Contact by finding an optimal number of message copies.

Creating more copies of a message increases the message delivery but decreases the network lifetime. These stochastic routing approaches usually consider the destinations of messages as nodes rather than locations.

Different from stochastic routing, current contact-based routing chooses the most appropriate nodes relaying messages to the destinations based on historical contact information, such as contact times, contact duration, and contact cycle.

Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [10] is a context-based routing protocol based on the history of encounters. PROPHET estimates the delivery predictability for each known destination at each node before passing a message. The estimation is based on the history of encounters between nodes. SimBet [11] uses historical contacts to calculate two metrics, similarity, and betweenness. The similarity is calculated by how frequently a node and its destination have met. The betweenness is calculated by how many nodes which a node has met. However, if the utility metrics are equal, SimBet will prevent its forwarding behavior. To improve this flaw, BUBBLE [12] adds the knowledge of community structure to ensure message dissemination. In addition, the betweenness may be useless if the message is near its destination. SimBetAge [13], an improved version of SimBet, was proposed to address these shortcomings. Spray and Wait [14], that uses only a handful of copies per message, can achieve

comparable delays to an oracle-based optimal scheme that minimizes delay while using the lowest possible number of transmissions.

The work in [15] does routing by simply calculating the delivery probability for a node to be at a location in MobySpace, which is a high-dimensional Euclidean space based on the preknown mobility model. However, the required assumption of that each node has the knowledge about mobility patterns of other nodes in the network makes this work impractical in realistic scenarios.

For the concerns of optimality, robustness, and flexibility, some routing protocols are developed by inspiration from biology [16–21]. Such routing schemes are also well applied in the transportation domain [22–26], which are based on prediction mechanism to improve the performance of transportation systems.

BeeHive is developed with inspiration of the foraging principles of honeybees [27]. Communication between real bees is modeled by designing intelligent bee agents, which are able to make routing decisions in large and complex topologies. The simulation results conclude that bee agents occupy smaller bandwidth and require less processor time. Artificial Bee Colony algorithm [17] is developed with scanning strategy for periodic vehicle routing problem. The work in [28] presents a biologically inspired discrete-event modeling approach for simulating alternative computer network protocols. Adaptation and probabilistic specifications are introduced into honeybee (BEE) and Routing Information Protocol (RIP) routing algorithms.

The ant colony optimization (ACO) methods have been inspired by operating principles of ants [29], which empower a colony of ants to perform complex tasks such as nest building and foraging [30].

Schoonderwoerd [31] first developed ACO-based approach for routing in telecommunication networks. The basic principle in the approach is about the use of stigmergy in multiagents interaction. Randomized ants traverse the network nodes probabilistically and select the highest probability path. The approach is shown to be sufficiently adaptive and demonstrates robustness across difficult network conditions. Yao et al. [19, 20] improved ACO for delivery routing problem and PROMETShop scheduling problems.

The aforementioned routing algorithms show good performance on message delivery. However geographic coordinates of nodes have little or no correlation with their contact times, so the algorithms do not perform well when each device frequently appears at different regions, as most people daily do.

In this paper, we propose the location-prediction and swarm-intelligence-based data dissemination (LOPSI) algorithm for opportunistic cognitive networks. The LOPSI algorithm is a probabilistic routing protocol combining location prediction and the ant colony optimization. It firstly predicts possible locations of relay nodes and destination(s) in successive time series. The mobile nodes calculate intimacy (contact frequency) with potential relay nodes using ACO and then make a forwarding decision based on node intimacy and probabilities of node mobility.

FIGURE 1: Network model of data dissemination in opportunistic cognitive networks.

The remainder of this paper is organized as follows. The system model is illustrated in Section 2. Section 3 specifies the data dissemination algorithm. Extensive simulations have been done for performance evaluation in Section 4. Section 5 concludes the paper.

2. Network Model

As mentioned in Section 1, the location-prediction and swarm-intelligence-based data dissemination (LOPSI) algorithm is a probabilistic routing protocol using location and intimacy information of potential relay nodes and destination nodes. The network is a mixture of an opportunistic network and a centralized infrastructure as shown in Figure 1. The centralized infrastructure consists of a number of wireless access points (APs) and a backbone connecting the APs.

Mobile nodes (carrying smart devices) can only access to the network when they are walking into the transmission range of any AP. Each AP periodically uploads connection records of mobile nodes to the location prediction server (LPS) which will eventually maintain a mobility database of all nodes.

The data exchanged in this scenario is assumed to be of high magnitude, and data transmission can only occurs between peer counterparts as in normal opportunistic networks. The LPS is merely responsible for storing node mobility records and predicting node location upon receiving query from mobile nodes via one of the APs. The accuracy of the localization prediction algorithm can increase the delivery ratio of the proposed approach.

In the proposed system, there exist two major entities: the mobile nodes and the LPS. The mobile nodes work in a decentralized manner, while the LPS is a powerful central unit which collects the trajectories of mobile users and performs

complex computation tasks to provide location prediction service to the request mobile users.

The mobile nodes only store the encountering information of the contact nodes in the local buffer. Based on the encountering information, each mobile node obtains the intimacy between the contact nodes and the destination node and then refines the forwarding probability to the destination node. The mobile nodes can forward the data message to the next hop by the optimal wireless channel according to its own system state. The mobile nodes can solve the data dissemination locally without the need of coordination with a central server or the other clients. Thus the data dissemination process is decentralized.

The LPS is applied to track the movement trajectories of all mobile nodes; thus it needs to collect global movement information predicting the encounter opportunities. Specifically, it uses the long-term trajectory information to construct the Markov chain of a mobile user and to determine the probable mobility trajectory based on partial encountering history. The computations in the server side and in the client side are independent, and their optimization results will not affect each other. Therefore, we have designed three data dissemination algorithms to adapt different infrastructure of the network environment. If the mobile nodes can connect to the LPS, it can obtain the location prediction information to optimize the accuracy of forwarding node selection, whereas the mobile nodes can use the local information, that is, intimacy, to estimate the delivery probability of the contact nodes.

The APs are deemed as living and working locations in the system. Nodes can communicate with each other through short-range communication media, for instance, ZigBee, Bluetooth, NFC, or WiFi Direct. Each node migrates from one location to another according to its own mobility model. When encountering other nodes, the mobile node dynamically calculates intimacy with them using ACO. The intimacy truly reflects historical contact information. It accumulates upon each contact and on the other hand decays over time.

Once a mobile node would like to send data to a destination node, it firstly consults a nearest AP for its neighbors' trend of movement. The LPS performs location prediction using Markov process inference and returns back to the query node an ordered list indicating probabilities of the neighbor nodes meeting the destination node in successive time series (usually more than one time slot).

The mobile node will compute forwarding probability of its neighbors by considering their intimacy with the destination node and then make a forwarding decision.

The mixture network model enhances traditional opportunistic networks with the centralized infrastructure which takes good advantage of existing AP assets but not burdens current network. The data forwarding is a probabilistic scheme guided by location prediction rather than stochastic or trivially probabilistic.

By this means, data will be delivered to the destination with higher probability and hence the network efficiency is improved.

3. Algorithm Design

In this section, we formulate a data dissemination problem in the network environment lacking contemporaneous end-to-end connectivity. To tackle the problem, we propose a contact-based probability routing algorithm, LOPSI, which implements the data dissemination by calculating the forwarding probability based on location prediction scheme and a swarm intelligence heuristic method. Therefore, LOPSI is fundamentally based on two routing schemes that are Location Prediction-Based Data Dissemination (LOPDAD) and Ant Colony Optimization- (ACO-) Based Data Dissemination (ACODAD). LOPDAD uses location prediction information to calculate the maximum probability of the location where the forwarding nodes and the destination node encounter, which is suitable for the opportunistic environment deploying a centralized infrastructure. ACODAD uses the swarm intelligence mechanism, ACO, to select the forwarding nodes according to the intimacy between the forwarding nodes and the destination node, which is suitable for the fully distributed data dissemination in opportunistic networks. LOPSI combines the merits of the two aforementioned algorithms. Depending on the requirements of applications, researchers can select the suitable algorithm to apply.

3.1. Location Prediction-Based Data Dissemination (LOPDAD). Studies on human mobility patterns have shown that people daily activities exist in a high degree of repeatability. People usually visit several fixed places regularly in each day and do activities in a relatively fixed period. According to mobile trajectories and regular behavior pattern of mobile nodes, it can be used to model the scene based on the location of mobile nodes and use relevant algorithms to predict the probability of the node arriving at a certain position and to estimate the location of the mobile node.

Markov chain algorithm is currently the most widely used in location prediction algorithm with high accuracy. Here we use the second-order Markov chain model to predict the location of the mobile node, which has higher accuracy than the first-order Markov chain model according to the simulation.

It can use Markov model to describe the application scenario such as Campus, where it is assumed that there are m locations. Location i is the i th status X_i of Markov process, and the state space is $E = \{X_1, X_2, \dots, X_m\}$. Thus scene mobility model is defined as $\{X, T\}$, and T is time series.

For each application scenario, Markov chain model can be used to predict the future location state of each mobile node. Specific modeling and forecasting process is as follows.

3.1.1. Preparation Process. Preparation before prediction process includes the following steps.

(1) *Determination of State Set.* According to the collection of mobile nodes trajectories from the system server, the location elements in the collecting data are counted, which is denoted as set L . As set L contains a number of location elements,

the locations of higher visiting frequency are chosen as state space of the system, denoted as set E , $E \subset L$.

(2) *Discretization of Data Set.* Statistical data of all users related to state set E is made. Then the dataset of each user is processed to be discrete set of the fixed time period, so the set after discretization is denoted as follows:

$$\{(t_k, X_i)\}, \quad k = 1, 2, 3, \dots, i \in \{1, 2, 3, \dots, m\}. \quad (1)$$

3.1.2. Location Prediction Algorithm Based on O2MM. Order-1 Markov chain model (O1MM) uses the state transition matrix and the initial distribution to predict, which is simple and intuitive [16]. However, as the indeterminacy of the state transition probabilities is unscientific division of the initial state of the system, the prediction result of this method tends to produce larger errors.

Unlike the first order, a Markov chain of higher order is a Markov model with memory, that is, a Markov chain that depends on not only the current state, but also on $n - 1$ states before, where n is the order and n is finite [17]. The chain is dependent on where it is right now and also where it was in the last occasion. Order-2 Markov chain model (O2MM) is used to improve the accuracy of prediction method. Compared with the prediction based on order-1 Markov chain, order-2 Markov chain model can be more complete and rational use of information and effectively integrated with correlation analysis, so as to improve prediction accuracy. O2MM depends on the current state and also the just visited state.

The finite state space of O2MM is

$$E = \{X_1, X_2, \dots, X_r, X_i, X_j, \dots, X_m\}, \quad i = 1, 2, \dots, m, \quad (2)$$

$$r = 1, 2, \dots, m, \quad j = 1, 2, \dots, m$$

and if the conditional probability is

$$P\{X(t_n) = X_j\} \\ = P\{X(t_n) = X_j \mid X(t_{n-1}) = X_i, X(t_{n-2}) = X_r\}, \quad (3)$$

the transition probability of the node located at X_j at time slice t_n under the condition that the node is located at X_i at time slice t_{n-1} and X_r at time slice t_{n-2} is

$$P_{rij}\{X(t_n) = X_j\} \\ = \sum_{i=1, r=1}^m P\{X(t_n) = X_j \mid X(t_{n-1}) = X_i, X(t_{n-2}) = X_r\}. \quad (4)$$

Formula (4) approximately equals the frequency that the node visits the location X_j when the state space tends to infinity [18]:

$$p_{rij} = \frac{\sum_{r=1, i=1}^{r=m, i=m} c_{rij}}{\sum_{k=1}^m c_{rik}}, \quad (5)$$

- (1) **input:** State Space Set $E = \{X_i, i \in \{1, 2, 3, \dots, m\}\}$, Nodes Set $N = \{N_j, j \in \{1, 2, 3, \dots, n\}\}$, the initial probability distribution is $P(n-1, n-2) = \{p_{ri}, r, i \in \{1, 2, 3, \dots, m\}\}$;
- (2) **Discretization of data set:** Statistical data of all users related to state set E is made. Then the data set of each user is processed to be discrete set of the fixed time slice, so the set after discretization is denoted as follow: $\{(t_k, X_i), k \in N^+, i \in \{1, 2, 3, \dots, m\}\}$;
- (3) calculate the probability of the node to visit location X_j according to (5) where the location state of the node at current time slice and also the just visited state is respectively X_i and X_r .
- (4) Calculate one step transition probability matrix according to (6);
- (5) Calculate the probability of each state at time slice t_n
 - (a) $P(n) = P(n-1, n-2) P$
- (6) the location state at time slice t_n is
 - (b) $X_j = \arg \max \{P_j^{(n)}\}$
- (7) **return** X_j ;

ALGORITHM 1: L -Markov(N_c, t_n): location prediction based on O2MM.

where c_{rij} is the number of times that the observation node visits location X_j by records statistics, $\sum_{k=1}^m c_{rik}$ is the total number of times that the node visits all the locations in E , and then the probability of the node to visit location X_j and the location state of the node at current time slice and also the just visited state is, respectively, X_i and X_r .

If there are m location states in state space set, one-step transition probability matrix is a $m \times m^2$ matrix:

$$P = \left[\begin{array}{ccccc} p_{111} & p_{112} & \cdots & p_{11(m-1)} & p_{11m} \\ p_{211} & p_{212} & \cdots & p_{21(m-1)} & p_{21m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{m11} & p_{m12} & \cdots & p_{m1(m-1)} & p_{m1m} \\ p_{m21} & p_{m22} & \cdots & p_{m2(m-1)} & p_{m2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{mm1} & p_{mm2} & \cdots & p_{mm(m-1)} & p_{mmm} \end{array} \right] m^2. \quad (6)$$

The description of the location prediction algorithm based on O2MM is as shown in Algorithm 1.

According to Algorithm 1, given the initial probability distribution of the node at time slice $n-1, n-2$, we can recurrence the probability distribution at time slice t_n by O2MM. The state set $E = \{X_1, X_2\}$, $m = 2$, the initial state is $\{(t_{n-2}, X_1), (t_{n-1}, X_2)\}$, and the initial probability distribution is

$$\begin{aligned} &P(n-1, n-2) \\ &= (p_{11}^{(n-1, n-2)}, p_{12}^{(n-1, n-2)}, p_{21}^{(n-1, n-2)}, p_{22}^{(n-1, n-2)}) = (0, 1, 0, 0). \end{aligned} \quad (7)$$

The transition matrix for Markov chain of order two is

$$P = \begin{bmatrix} p_{111} & p_{112} \\ p_{121} & p_{122} \\ p_{211} & p_{212} \\ p_{221} & p_{222} \end{bmatrix}. \quad (8)$$

Here the first two numbers of the index representing the current state and the last number represent the next state. And the probability distribution at time slice n is

$$P(n) = P(n-1, n-2) P = \{p_1^{(n)}, p_2^{(n)}\}. \quad (9)$$

The location state is obtained by the location prediction server at time slice n is

$$X_j = \arg \max \{p_1^{(n)}, p_2^{(n)}\}. \quad (10)$$

In our system the location prediction server gathers the location data of the mobile nodes by the APs. The server executes Algorithm 1 to predict the trajectory of each node according to the discrete time slice.

3.1.3. Transmission Probability of LOPDAD. The data dissemination mechanism can use the result of the location prediction algorithm using O2MM. At a certain time slice, the data forwarding probability equals the probability of the forwarding node visiting the location where a destination node is, which is given by

$$p_L^d = p_{X_j^d}^{(n)}, \quad (11)$$

where X_j^d is the location state of destination node N_d at time slice t_n , and $p_{X_j^d}^{(n)}$ is the probability that node N_c visits the location X_j^d at time slice t_n .

Algorithm 2 describes the process of the location-based data dissemination algorithm, which obtained the forwarding probability p_L^d of N_c to the destination node N_d .

3.1.4. The Execution of Location Prediction. The LPS predicts the locations where the destination node and the forwarding nodes will encounter at the future time slices. The threshold of time slices is l . At time slice, t_n , when the LPS receives the service request information $\text{REQ}(C\{N_f\}, N_d)$ which includes the destination node N_d and the encounter nodes set $C\{N_f\}$, from the data carrier node N_s , the LPS calculates the

Input: State Space Set $E = \{X_i, i \in \{1, 2, 3, \dots, m\}\}$, N_d, N_c , the initial probability distribution is $P(n-1, n-2) = \{p_{ri}, r, i \in \{1, 2, 3, \dots, m\}\}$;

Output: p_L^d
 (1) $X_j^d = L_Markov(N_d, t_n)$
 (2-5) Algorithm 1 steps 1-5;
 (6) calculation of forwarding probability according to (11)
 (7) return p_L^d ;

ALGORITHM 2: $P_Markov(N_c, t_n)$: data dissemination probability based on location prediction.

```

(1)  $N_s \rightarrow LPS: REQ(C\{N_f\}, N_d)$   $i = 0, j = 0, k = 0, l = 3$ ;
(2) for all  $n \in [1, l]$  do
(3)  $X(N_d, t_n) = L\_Markov(N_d, t_n)$ ;
(4) for all  $N_f \in C\{N_f\}$  do
(5)  $X(N_f, t_n) = L\_Markov(N_f, t_n)$ ; //calculate the location state of  $N_f$  at  $t_n$ 
(6)   if  $(X(N_f, t_n) == X_i(N_d, t_n))$  then //  $N_f$  and  $N_d$  encounter at  $t_n$ 
(7)      $F.N[j++] = N_f$ ; //  $N_f$  is stored as the forwarding node in set  $F.N$ 
(8)      $F.P[i++] = P\_Markov(N_f, t_n)$ ; //The probability of  $N_f$  meets  $N_d$  is stored in set  $F.P$ 
(9)      $F.T[k++] = n$ ; //the encounter time slice of  $N_f$ 
(10)   end if
(11) end for
(12) end for
(13)  $LPS \rightarrow N_s: SEI(F\{N_f, P, T\}, \{X(N_d, t_n)\})$ ;

```

ALGORITHM 3: $LoP_Service(C\{N_f\}, N_d)$: the location prediction service in the LPS.

location state set $X = \{X(N_d, t_l)\}$ of N_d in the time slice $t_{n+1}, t_{n+2}, \dots, t_{n+l}$:

$$\{X(N_d, t_l)\} = \{(t_1, X_i), (t_2, X_j), \dots, (t_l, X_p)\} \subseteq \{(t_k, X_i)\},$$

$$k \in N^+, \quad i, j, p \in \{1, 2, 3, \dots, m\}. \quad (12)$$

And the LPS also calculates the forwarding probability F . P of each node N_f in the set $C\{N_f\}$. The forwarding nodes set $F.N$ in which the node N_f will visit the location where the destination node N_d locates at the same time slice, and the encounter time slice is recorded in $F.T$. Finally, the server sends the service information $SEI(F\{N_f, P, T\}, \{X(N_d, t_n)\})$ to the node N_s . The execution process is described in Algorithm 3.

3.1.5. The Selection Mechanism of the Forwarding Nodes Set. When the data carrier N_s receives the SEI from the predictive server, considering the cache management and the load of the network, the number of copies of the forwarding data is a fixed value, COPY, which is decided by the average buffer size of each node and the current load of the network. If N_s finds that the location of N_d is the same as N_s during the threshold time slice, it only transmits to the nodes which can encounter N_d at earlier time slice than N_s . Otherwise, if the number of nodes in the forwarding nodes set is less than COPY, N_s transmits the data to the nodes in the forwarding nodes set and delete the data in its own buffer. If the number of nodes

in the forwarding nodes set is more than COPY, N_s transmits the data only to the nodes having maximum probability no more than COPY, according to the following:

$$P_{Lsf}^d = \left\{ \frac{P_Markov(N_f, t_n)}{n}, N_f \in F, t_n \in T, n \in [1, l] \right\}, \quad (13)$$

where $P_Markov(N_f, t_n)$ is the forwarding probability which is equal to the probability that N_f encounters N_d at time slice t_n , where n is the number of time slice intervals when N_f meets N_d . The larger the number of time slice intervals is, the lower the probability of forwarding node is. And the forwarding nodes set F' which is selected by N_s :

$$F' \subseteq F, \quad F' \{N_f\} = \operatorname{argmax}_{f \leq \text{COPY}} \{P_{Lsf}^d\}. \quad (14)$$

3.2. Swarm Intelligence Heuristic Data Dissemination (ACO-DAD). The inspiring source of ACO is the pheromone trail laying and following behavior of real ants which use pheromones as a communication medium. Artificial ants used in ACO are stochastic solution construction procedures that probabilistically build a solution by iteratively adding solution components to partial solutions by taking into account (i) heuristic information on the problem instance being solved, if available, and (ii) (artificial) pheromone trails which change dynamically at run-time to reflect the agents' acquired search experience [19]. In our algorithm,

ACO is improved to be applied in data dissemination in opportunistic cognitive networks.

3.2.1. Ant Colony Optimization- (ACO-) Based Data Dissemination in OCN (ACODAD). The pheromone in ACODAD is the intimacy between two nodes. The more frequency and continuous the contact between two nodes is, the higher the value of intimacy is. It means that the higher the pheromone is. The data carrier node tends to choose the node which has high intimacy value with the destination nodes to forward the data. It means that the data forwarding probability of the node having high intimacy with the destination node is high. The comparison of the characteristics between ACODAD and ACO is shown in Table 1.

3.2.2. Intimacy between Two Nodes in OCN. Each node in opportunistic cognitive networks maintains a relationship table with other nodes using the value of intimacy.

For example, at time slice t_s , the data carrier node N_s arrives at location X_i and senses all the other nodes within the communication range via the communication channels (such as ZigBee, Bluetooth, NFC, and other short-range communication protocols). Those nodes are added in encounter nodes set $C\{N_f\}$ of N_s and are recorded in the relationship table with the calculation results of intimacy.

The intimacy between two nodes N_i and N_j depends on the frequency of two nodes in connection based on contact times n_{ij} , lasting time of one connection ΔD_{ij}^c , and the encounter intervals between two contacts ΔI_{ij}^c . The mathematical description of intimacy is as follows, where $R_{ij}(t)$ is the intimacy of N_i and N_j at time slice t :

$$R_{ij}(t) = \begin{cases} \frac{n_{ij} \times \sum_{c=1}^{n_{ij}} \Delta D_{ij}^c}{\sum_{c=1}^{n_{ij}} \Delta I_{ij}^c}, & \forall \Delta I_{ij}^c < k \\ \frac{n_{ij} \times \sum_{c=1}^{n_{ij}} \Delta D_{ij}^c}{\sum_{c=1}^{n_{ij}} \Delta I_{ij}^c} \times (1 - \rho)^e, & \exists \Delta I_{ij}^c > k \\ 0, & \Delta I_{ij}^c > T, \text{ over the system running time,} \end{cases} \quad (15)$$

where ρ is the evaporation rate, k is the threshold of encounter time intervals of two nodes, and e is the times when the time intervals is more than k .

Although the intimacy is defined in consideration of encounter time intervals, if the two nodes are not in connection for a long time (more than a certain time threshold value k), it enables the intimacy evaporation mechanism to ensure the most frequently contact nodes with a high degree of intimacy. If the encounter time intervals are beyond the system running time T , the value of intimacy is zero.

When two nodes encounter, they, respectively, compute the intimacy based on records of encounter time t_s and the departure time t_e at the contact times c . Each node records

the time according to its own time clock. When $c = 1$, ΔI_{ij}^c equals the encounter time of the first contact. The algorithm of intimacy is as shown in Algorithm 4.

Algorithm 4 captures the essence of (15). The intimacy between two nodes is used to compute the forwarding probability in ACODAD.

The intimacy updates during the time duration Δt ; the variation of the intimacy is given by

$$R_{ij}(t + \Delta t) = R_{ij}(t) + \Delta R_{ij}(t),$$

$$\Delta R_{ij}(t) = \begin{cases} \left(\frac{(n_{ij} + \Delta n_{ij}) \times \sum_{c=1}^{(n_{ij} + \Delta n_{ij})} \Delta D_{ij}^c}{\sum_{c=1}^{(n_{ij} + \Delta n_{ij})} \Delta I_{ij}^c} - \frac{n_{ij} \times \sum_{c=1}^{n_{ij}} \Delta D_{ij}^c}{\sum_{c=1}^{n_{ij}} \Delta I_{ij}^c} \right), & \Delta n_{ij} > 0, \forall \Delta I_{ij}^c < k \\ -\rho \times \frac{n_{ij} \times \sum_{c=1}^{n_{ij}} \Delta D_{ij}^c}{\sum_{c=1}^{n_{ij}} \Delta I_{ij}^c}, & \Delta n_{ij} = 0, \Delta t > k, \forall \Delta I_{ij}^c < k \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

3.2.3. Transmission Probability of ACODAD. When the data carrier node N_s encounters the nodes in set $C\{N_f\}$ and chooses the forwarding nodes from $C\{N_f\}$ to the destination N_d based on the intimacy between the node in $C\{N_f\}$ and N_d , the forwarding probability is defined as the following equation:

$$p_{R_{sf}}^d = \begin{cases} 1 & j = d \\ 0 & \text{Intimacy}(N_f, N_d) < \text{Intimacy}(N_i, N_d) \\ \frac{[R_{fd}(t)]^\alpha \cdot [\eta_{fd}(t)]^\beta}{\sum_{N_f \in \text{allowd}_d} R_{jd}^\alpha \cdot \eta_{jd}^\beta} & N_f \in \text{allowd}_d, \end{cases} \quad (17)$$

where allowd_d is given by

$$\text{allowd}_d^t = F\{N_f \mid (R_{fd}(t) > R_{id}(t))\}, \quad (18)$$

$$\text{allowd}_d^t \in F\{N_f\}, \quad \text{allowd}_d = \text{allowd}_d^t - \text{tabu}_d,$$

where tabu_d is the node set including those nodes which have already carried the transmission data copy and also the node had the copy before. Such nodes will not be selected as the forwarding nodes.

The parameters α and β control the relative importance of the pheromone versus the heuristic information, η_{fd} , which is given by

$$\eta_{fd} = \frac{1}{L_{fd}}, \quad (19)$$

where L_{fd} is the time slice intervals that N_f and N_d will encounter which is an estimation value given by the records in the contact vector.

A heuristic value η , respectively, represents a priori information about the problem instance definition or run-time information provided by a source different from the ants.

TABLE 1: Comparison of characteristics between ACODAD and ACO.

Characteristics	ACODAD	ACO
Transmission process of data	Data dissemination based on the encounter of two nodes	Artificial ants move from one location to the neighbor one
Transition probability	The probability that the forwarding node encounters the destination node	The state transition probabilities that are from one location to the next
Pheromone	Intimacy between two nodes	Pheromone trail laying by ants
Path length	The interval of time slices that two nodes encounter	The distance between two locations
Pheromone evaporation	No contact between the two nodes during threshold time	As time goes by the pheromone evaporates

```

Input:  $N_i, N_j$ 
Output:  $R_{ij}(T)$ 
(1) define  $T$  the system running time
(2)  $L$  encounter time interval threshold
(3) Contact vector array  $\vec{M}_{ij}(c, t_s, t_e)$ 
(4) initialization:  $\vec{M}_{ij}[0] = (0, 0, 0)$ ,  $R_{ij}(0) = 0$ ,  $k = L$ ,  $flag = 1$ ;
(5) forall  $t = 0 : T$ ;
(6) if  $N_i$  recieve HELLO from  $N_j$ 
(7)  $n_{ij} = ++c$ ;
(8) for all  $c = 1 : n_{ij}$ 
(9)  $\Delta D_{ij}^c[c] = t_e - t_s$ ;
(10)  $D_{ij}^c += \Delta D_{ij}^c$ ;
(11)  $\Delta I_{ij}^c[c] = t_s - t$ ;
(12)  $I_{ij}^c += \Delta I_{ij}^c$ ;
(13)  $t = t_e$ ;
(14) if  $(\Delta I_{ij}^c[c] > k)$ 
(15)  $e++$ ;
(16)  $flag = 0$ ;
(17) end if
(18) end for
(19) end if
(20) end for
(21) if  $(flag == 0)$ 
(22) return  $R_{ij}(T) = ((n_{ij} \times D_{ij}^c) / I_{ij}^c) \times (1 - \rho)^e$ 
(23) else
(24) return  $R_{ij}(T) = (n_{ij} \times D_{ij}^c) / I_{ij}^c$ 
(25) end if

```

ALGORITHM 4: Intimacy(N_i, N_j) computes the intimacy value between two nodes at encounter time T .

```

(1) Initialization: node set  $N = \{N_i, i = 1, 2, \dots, k, k \in N^+\}$ , state space  $E = \{X_j, j = 1, 2, \dots, m, m \in N^+\}$ , time slice set  $T = \{t_n, n = 0, 1, 2, \dots, l, l \in N^+\}$  Maximum data copies: COPY;
(2) for  $n = 0 : l$  // the system running time
// the function executes the one hop data dissemination to the forwarding node set
(3) LOPSI_Sec( $N_s$ )
(4) for  $\forall N_f \in F'\{N_f\}$ 
(5) execution of LOPSI_Sec( $N_f$ );
(6) end for
(7) end for

```

ALGORITHM 5: LOPSI(N_s, N_d).

In many cases η is the cost, or an estimate of the cost, of extending the current state. These values are used by the ants' heuristic rule to make probabilistic decisions on how to move on the graph [32].

In our algorithm, the heuristic value is defined as the distance between N_f and N_d . As the location state of nodes is discretized by the time slice. And the delay of data forwarding between the two nodes is not decided by the absolute path length or the distance between the two nodes but by calculating the number of time slice intervals between forwarding node and the destination node to meet with each other.

The data carrier node N_s chooses the forwarding nodes from set $C\{N_f\}$ according to the forwarding probability based on intimacy. Considering the cache management, our algorithms set the maximum copy quantity of a unique data message. If the data forwarding operation is executed, N_s will transmit the data to the nodes in set $C'\{N_f\}$, which is given by (20), and the quantity of nodes, denoted by f , is no more than COPY. Finally, N_s will delete the data stored in its buffer:

$$C' \subseteq C\{N_f\}, \quad C'\{N_f\} = \underset{f \leq \text{COPY}}{\operatorname{argmax}} \{P_{R_{sf}}^d\}. \quad (20)$$

3.3. Location Prediction Based Data Dissemination Using Swarm Intelligence (LOPSI). The data dissemination algorithm LOPSI considers not only the intimacy between the forwarding node and the destination node, but also the location where the two nodes may encounter. According to the location prediction algorithm, the set of forwarding nodes, $F\{N_f\}$, which visit the location where the destination node will be during the threshold time slices, can be obtained. And the data carrier node N_s only sends message to nodes in $F\{N_f\}$ and compares the intimacy between N_d and $N_f \in F\{N_f\}$. Combined with the prediction results from LOPDAD and ACODAD, the transmission probability from node N_s to the forwarding node N_f is calculated by the weight formula

$$p_{sf}^d = \gamma * p_{R_{sf}}^d + \delta * p_{L_{sf}}^d, \quad \gamma + \delta = 1, \quad (21)$$

where $p_{R_{sf}}^d$ is the forwarding probability obtained from ACODAD, $p_{L_{sf}}^d$ is the forwarding probability obtained from LOPDAD, and parameters γ and δ are the weights of the two probabilities.

In our campus environment the mobile node can communicate to the location prediction server anywhere and anytime. The prediction server provides only location prediction service and no other services. Algorithm 5 describes the whole execution process of LOPSI. LOPSI.Sec is part of Algorithm 5, which is a function that the data message transfers from the current node to the forwarding node set. When the source node N_s wants to send the data message to the destination node N_d , N_s will choose the potential nodes by location prediction schemes and then obtain the intimacy between the potential nodes and the destination node. Thus the forwarding node set will be determined. The execution process of LOPSI is described in Algorithm 5.

FIGURE 2: The simulation based on a realistic campus scenario.

TABLE 2: The accuracy of location prediction algorithms based on O1MM and O2MM.

	O1MM	O2MM
Prediction accuracy	0.5610	0.8030
Time complexity	$O(N)$	$O(N^2)$
Storage space	$O(N^2)$	$O(N^3)$

For any node N_i carrying data message, LOPSI.Sec(N_i) will determine the forwarding node set. The description of LOPSI.Sec(N_i) is given as shown in Algorithm 6.

Algorithm 5 describes the essence of LOPSI. Each data has a TTL, which indicates how long the data can live in the network. It is set by the provider at the time of data generation. In Algorithm 5 the length of time slices l equals TTL. The data carried by each node within TTL duration cannot be forwarded and then be automatically discarded.

4. Performance Evaluation

In this section, we present the simulation results to demonstrate the performance of proposed data dissemination algorithms. Note that the recent work studying the nature of human mobility has proved that suitable movement models can sufficiently present the behavior of human mobility [33]. The model of mobility pattern deployed in our platform is SPMBM model [34], which is a mobility model that integrates temporal and spatial relationships and selects the shortest path for the node randomly walking in the map area.

4.1. Simulation Settings. The data sets to evaluate the location prediction algorithm based on O2MM are obtained from wireless topology discovery (WTD) [35], which are employed in our simulation. The accuracy of the location prediction algorithms based on O1MM and O2MM can be obtained from our previous work [36], which is given in Table 2.

The simulation is based on a realistic campus scenario shown in Figure 2. There are 40 locations and each one installed a WiFi access point, which can cover the campus

```

(1)  $N_i$  senses the contact nodes set  $C\{N_f\}$ ;
(2) for  $\forall N_f \in C\{N_f\}$ 
(3)   update intimacy( $N_i, N_f$ );
(4) end for
    /* the source node deliver the data directly to the destination node if the destination node is in the contact
    nodes set  $C\{N_f\}$  and update the node set  $\text{tabu}_d$  */
(5) if  $N_d \in C\{N_f\}$ 
(6)    $N_i$  transmits data to  $N_d$ ;
(7)    $\text{tabu}_d \leftarrow \{N_i, N_d\}$ ;
(8)   break the Algorithm 5;
(9) else
(10)   $N_s$  sends REQ( $C\{N_f\}, N_d$ ) to the server;
(11)  the server executes LoP_Service( $C\{N_f\}, N_d$ );
(12)   $N_s$  receives SEI;
(13)   $N_s$  sends  $N_d$  to the nodes in  $F\{N_f\}$ ;
(14)  for  $\forall N_f \in F\{N_f\}$ 
(15)    Send intimacy( $N_f, N_d$ ) to  $N_s$ ;
(16)  end for
(17)   $N_s$  calculates  $p_{sf}^d$  according to (21);
(18)  if ( $f \leq \text{COPY}$ ) //  $f$  is the number of forwarding nodes
(19)     $F'\{N_f\} = F\{N_f\}$ ;
(20)  else
(21)     $F'\{N_f\} = \{\arg \max_{f \leq \text{COPY}} (p_{sf}^d)\}$ 
(22)  end if
(23)   $N_i$  sends data to  $F'\{N_f\}$ ;
(24)   $\text{tabu}_d \leftarrow \{N_i, F'\{N_f\}\}$ ;
(25)   $N_i$  delete the data copy in its buffer;
(26) end if

```

ALGORITHM 6: LOPSI_Sec(N_i): the function executes one-hop data dissemination to the forwarding node set.

TABLE 3: Simulation parameters.

	Parameter	Value
Scene features	Simulation time	12 h
	Field area	4500 m * 3400 m
	Scene	NEU Campus
	APs	40
Node features	Mobility model	SPMBM
	Movement speed for cars	2.7–13.9 m/s
	Movement speed for pedestrians	0.5–1.5 m/s
	Transmission rate	250 KB/s
	Maximum transmission range	10 m
	Transmission mode	Broadcast
Message features	Cache size	10 MB/1 G
	Packet size	500 KB–1 MB at random
	Frequency of creating packets	From 25 s to 35 s at random
	Number of copies	8
	TTL	5 hours

area. The mobile nodes can be cars and pedestrians with smart phones. The location prediction server can communicate with mobile nodes via WiFi and only provide location prediction service and no other services. The only way to exchange and obtain data is through the contact of two nodes. If the communication range increases and is out of the range

of APs, the mobile devices can exchange data by the ACO-DAD without using location prediction scheme by different wireless communication techniques including WiFi Direct, Bluetooth, and ZigBee. In order to evaluate the performance of the proposed data dissemination algorithms, we conduct a series of experiments under the parameters in Table 3. The

First In First Out is applied on buffer management. In order to avoid the heavy traffic load and cache load, each data is set a TTL, and the maximum copies of each data is a fixed constant in the system.

With the above settings, the three data dissemination algorithms proposed in our work LOPDAD, ACODAD, and LOPSI are evaluated and compared with well-known opportunistic routing protocols: PRoPHET and Spray and Wait.

4.2. Evaluation Metrics. Four metrics are used to evaluate the performance requirements of the aforementioned data dissemination algorithms: average hops, delivery ratio, average latency, and transmission cost.

Average Hop H . This hop-count metric is to assess the delivery cost in time and in cache. N denotes the total number of forwarding nodes of every transmission of data including both successful and failure delivery. Y is the total number of created unique data messages. H is given by

$$H = \frac{N}{Y}. \quad (22)$$

Delivery Ratio R . This metric is to evaluate the effectiveness and utility of the algorithm. S is the total number of successfully delivered unique data messages. R is given by

$$R = \frac{S}{Y}. \quad (23)$$

Average Latency L [35]. The average latency of a unique message is calculated by the following equation:

$$L = \frac{1}{Y} \sum_{i=1}^Y (T_{d_i} - T_{s_i}), \quad (24)$$

where T_{s_i} is the moment that a unique data message i is originated, and T_{d_i} is the time when the first replicate of unique message i arrives at the destination. The replicate is a copy of a unique message. The number of replicates depends on the methodology of the routing algorithm, single or multiple copies [35].

Transmission Cost C . It is a key metric to characterize the resource consumption and evaluate the data dissemination algorithms in OCNs. The total number of copies of unique data messages to deliver in the system, denoted by Q , is divided by the number of copies of successfully delivered messages, denoted by P :

$$C = \frac{Q}{P}. \quad (25)$$

4.3. Influence on Performance with the Variation of Total Number of Nodes

Average Hops. As shown in Figure 3(a), with the increasing of numbers of nodes, due to the hop limitation of Spray and Wait, the average hops are relatively small; PRoPHET has no restrictions in this respect; the frequency of nodes encountering increases, resulting in an increase in the average hops. Our algorithms are proposed to consider the management of cache space and limit the maximum number of copies of the nodes in the network, which constrains the hops of data messages transmission. LOPDAD and LOPSI especially set a time threshold on executing location prediction, which guaranteed the data message delivery to the destination node within the time threshold. The location state is discrete by the time slices, so threshold of time slices corresponds to the number of hops. Therefore, the average hops of our algorithms are relatively small.

Delivery Ratio. As shown in Figure 3(b), whether for Spray and Wait, PRoPHET, and our data distribution algorithms, the delivery ratio significantly increases with the increase of the nodes. LOPDAD, ACODAD, LOPSI, and Spray and Wait have constrained the number of copies of the data messages transmitted in the network. Even if the number of nodes and the amount of data messages increases, the storage space and the network overhead maintain a good status, avoiding data transmission failure by the heavy load of cache and network resources depletion. Therefore, delivery ratio is better than the PRoPHET transmission. LOPDAD selects forwarding nodes which are most likely to complete the task based on location prediction. ACODAD selects the best forwarding nodes by high intimacy. LOPSI tends to make more “assertive” options to select the forwarding nodes based on the two aforementioned factors, so the delivery ratio has been significantly improved.

Average Latency. As shown in Figure 3(c), the average latency is reduced with the increase of the nodes, indicating that our data distribution algorithms are assertive to select the forwarding nodes which are more likely to contact with the destination node. The data transfer of LOPDAD occurs at the location that can connect with AP points, not at any other encounter places, so the average delay is longer than that of ACODAD and LOPSI. The forwarding nodes selection mechanism of LOPSI is better than that of ACODAD, which has less transmission operation but high delivery ratio, so the average latency is lower than that of ACODAD.

Transmission Cost. Figure 3(d) shows the transmission cost of our proposed algorithm and some existing algorithms. LOPSI has the lowest transmission cost since it only transfers messages to the nodes with the highest forwarding probability to the destination and the quantity of copies of the message is a constant which equals the hops estimated by the location prediction algorithm. With the increase of number of nodes, the opportunity of forwarding messages to the potential nodes increases, which leads to increase of the

(a)

(b)

(c)

(d)

FIGURE 3: (a) Average hops variation with the increase of the number of nodes. (b) Delivery ratio variations with the increase of the number of nodes. (c) Average latency variations with the increase of the number of nodes. (d) Transmission cost variations with the increase of the number of nodes.

delivery ratio, and also the average hops decrease (as shown in Figure 3(a)). PRoPHET has the highest transmission cost since it has no consideration in the cache management. The transmission cost is at the value of more than 1100, much more than the transmission cost of the other four algorithms. Thus the curve of transmission cost of PRoPHET cannot be drawn in the scale of the graph. The transmission costs of the other three algorithms are lower than that of PRoPHET since they constrained the quantity of copies, but not better than LOPSI because the selection schemes of potential forwarding nodes are no better than that of LOPSI.

4.4. Influence on Performance with the Variation of TTL. In this scenario, in order to illustrate the influence on performance by the variation of TTL, the simulation parameter of the number of nodes is set to be 200.

Average Hops. As shown in Figure 4(a), with the increase of TTL, the data messages live long in the network, which will increase the load of cache and the network. However, LOPDAD and LOPSI have little change in the average hops since the most data messages are successfully delivered to

FIGURE 4: (a) Average hops variations with the increase of TTL. (b) Delivery ratio variations with the increase of TTL. (c) Average latency variations with the increase of TTL. (d) Transmission cost variations with the increase of TTL.

the destination in the time threshold according to the accuracy of O2MM being 80%. During the TTL, if the data dissemination is failure in the first time threshold period, then start the second time threshold to deliver the data message; the average hops may be doubled; however, the probability of that condition is less than 20%. The average hops of ACODAD are more than LOPDAD and LOPSI, as it has no consideration of location where the destination node will be, only by random encounter of two nodes. With the increase of the network load, the average hops will increase.

Delivery Ratio. As shown in Figure 4(b), with the increase of TTL, the delivery ratio of LOPDAD changes little. According

to the characteristic of Markov chain prediction, the future status of long-time prediction tends to be stabilized, which means that the prediction accuracy is reduced. If the time threshold is set too big, then delivery ratio will decrease. The delivery ratio of ACODAD and LOPSI increase since the copy of data message in the network is not changed but the opportunistic of encounter is increased.

Average Latency. As shown in Figure 4(c), with the increase of TTL, the average latency of those five algorithms will increase since the network load is heavy and the total number of data messages increases, which makes the buffer overcrowded. Since the algorithms except PRoPHET constrain the copies of messages to avoid the traffic loads, the latency is lower

than that of PROPHET which transfers the message to any potential nodes without copies constraint.

Transmission Cost. As shown in Figure 4(d), with the increase of TTL, the transmission cost of LOPSI and LOPDAD decreases since the delivery ratio is higher (as shown in Figure 4(b)) and the number of copies of a unique message is slightly changed. It results in that LOPSI and LOPDAD estimate the optimal path and then determine the number of copies of a unique message. Spray and Wait and ACODAD constrain the number of copies, so the variation of transmission cost is slight. As regards PROPHET, since it does not constrain the copies of a unique message, the transmission cost is much higher than the transmission cost of the other four algorithms and at a more than thousand value. Thus the curve of transmission cost of PROPHET cannot be drawn in the scale of the graph.

5. Conclusion and Future Work

In this paper we consider the efficient data dissemination mechanism in opportunistic cognitive networks. We propose a swarm intelligence heuristic data dissemination algorithm based on location prediction. The algorithm can select the efficient forwarding nodes with the maximum probability to encounter the destination node at the location where they most likely to encounter and a high value of intimacy with the destination node. Moreover the algorithm considers the cache management and has good performances in the transmission cost and delivery ratio and simultaneously decreases the average hops and delivery delay. The algorithm has good scalability, which can consider credibility, incentives, energy control, and buffer management mechanism in the future work. Furthermore, the swarm intelligence scheme can also be used in group construction of the mobile nodes, which can promote the application performance in opportunistic cognitive networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank all the editors of this paper. They read the manuscript very carefully and provided valuable feedbacks, which are helpful to improve the quality of the draft.

References

- [1] Q. H. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*, John Wiley & Sons, London, UK, 2007.
- [2] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.
- [3] R. Yu, P. Wang, and Z. Zhao, "NDI: node-dependence-based dynamic gaming Incentive algorithm in opportunistic networks," in *Proceedings of the 23rd International Conference on Computer Communications and Networks (ICCCN '14)*, pp. 581–588, Shanghai, China, 2014.
- [4] R. Yu, R. Liu, X. Wang, and J. Cao, "Improving data quality with an accumulated reputation model in participatory sensing systems," *Sensors*, vol. 3, pp. 5573–5594, 2014.
- [5] B. Hull, V. Bychkovsky, K. Chen et al., "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems*, pp. 125–138, November 2006.
- [6] M. Mun, S. Reddy, K. Shilton et al., "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 55–68, June 2009.
- [7] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep., Department of Computer Science, Duke University, Durham, NC, USA, 2000.
- [8] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceeding of the Conference on Computer Communications (ACM SIGCOMM '04)*, pp. 145–158, New York, NY, USA, September 2004.
- [9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 235–244, October 2004.
- [10] A. Lindgren and A. Droia, "Probabilistic routing protocol for intermittently connected networks," Internet Draft draft-lindgren-dtnrg-prophet-02, Work in Progress, 2006.
- [11] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, ACM, New York, NY, USA, 2007.
- [12] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 241–250, May 2008.
- [13] J. Á. B. Link, N. Viol, A. Goliath, and K. Wehrle, "SimBetaAge: utilizing temporal changes in social networks for pocket switched networks," in *Proceedings of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities (U-NET '09)*, pp. 13–18, ACM, New York, NY, USA, December 2009.
- [14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshops: Conference on Computer Communications*, pp. 252–259, August 2005.
- [15] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proceedings of the ACM SIGCOMM Workshops: Conference on Computer Communications*, pp. 276–283, August 2005.
- [16] B. Z. Yao, P. Hu, X. H. Lu, J. J. Gao, and M. H. Zhang, "Transit network design based on travel time reliability," *Transportation Research C*, vol. 43, pp. 233–248, 2014.
- [17] B. Yao, P. Hu, M. Zhang, and S. Wang, "Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem," *Simulation*, vol. 89, no. 6, pp. 762–770, 2013.

- [18] B.-Z. Yao, C.-Y. Yang, and J.-B. Yao, "Tunnel surrounding rock displacement prediction using support vector machine," *International Journal of Computational Intelligence Systems*, vol. 3, no. 6, pp. 843–852, 2010.
- [19] B. Yao, P. Hu, M. Zhang, and X. Tian, "Improved ant colony optimization for seafood product delivery routing problem," *PROMET—Traffic & Transportation*, vol. 26, no. 1, pp. 1–10, 2014.
- [20] B. Yao, C. Yang, J. Hu, J. Yao, and J. Sun, "An improved ant colony optimization for flexible job shop scheduling problems," *Advanced Science Letters*, vol. 4, no. 6-7, pp. 2127–2131, 2011.
- [21] B. Z. Yao, J. B. Yao, and M. H. Zhang, "Improved support vector machine regression in multi-step-ahead prediction for rock displacement surrounding a tunnel," *Scientia Iranica*. In press.
- [22] B. Yu, Z. Z. Yang, and K. Chen, "Hybrid model for prediction of bus arrival times at next station," *Journal of Advanced Transportation*, vol. 44, no. 3, pp. 193–204, 2010.
- [23] Y. Bin, Y. Zhongzhen, and Y. Baozhen, "Bus arrival time prediction using support vector machines," *Journal of Intelligent Transportation Systems*, vol. 10, no. 4, pp. 151–158, 2006.
- [24] B. Yu, Z. Z. Yang, and B. Z. Yao, "A hybrid algorithm for vehicle routing problem with time windows," *Expert Systems with Applications*, vol. 38, no. 1, pp. 435–441, 2011.
- [25] B. Yu, H. B. Zhu, W. J. Cai, N. Ma, and B. Z. Yao, "Two-phase optimization approach to transit Hub location—the case of Dalian," *Journal of Transport Geography*, vol. 33, pp. 62–71, 2013.
- [26] B. Yu, Z. Yang, and J. Yao, "Genetic algorithm for bus frequency optimization," *Journal of Transportation Engineering*, vol. 136, no. 6, pp. 576–583, 2010.
- [27] M. Farooq, *Bee-Inspired Protocol Engineering: From Nature to Networks*, Springer, New York, NY, USA, 2009.
- [28] A. Zengin, H. Sarjoughian, and H. Ekiz, "Discrete event modeling of swarm intelligence based routing in network systems," *Information Sciences*, vol. 222, pp. 81–98, 2013.
- [29] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [30] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.
- [31] R. Schoonderwoerd, *Collective intelligence for network control [M.S. thesis]*, Faculty of Technical Informatics, Delft University of Technology, 1996.
- [32] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: algorithms, applications, and advances," in *Handbook of Metaheuristics*, pp. 251–285, Springer, 2002.
- [33] V. D. Le, H. Scholten, P. J. M. Havinga, and H. Ngo, "Location-based data dissemination with human mobility using online density estimation," in *Proceedings of the 11th Annual IEEE Consumer Communications & Networking Conference*, pp. 747–754, Las Vegas, Nev, USA, November 2014.
- [34] A. Ahmed and K. Abu Bakar, "A simulation based study of well known routing protocols for delay tolerant network," *World Applied Sciences Journal*, vol. 28, no. 3, pp. 353–360, 2013.
- [35] M. McNett and G. M. Voelker, UCSD Wireless Topology Discovery Project [EB/OL], 2013, <http://www.sysnet.ucsd.edu/wtd/wtd.html>.
- [36] J. Li, X. Xing, R. Yu, X. Wang, and Y. Zhou, "Social relationship-based mobile node location prediction algorithm in opportunistic cognitive networks," *WIT Transactions on Information and Communication Technologies*, vol. 59, pp. 113–119, 2014.