QoS-Oriented Mobile Service Composition Over Oppotunistic Networks

QingLan Peng, *Student Member, IEEE,* ChunRong Wu, *Student Member, IEEE,*

Abstract—Opportunistic mobile service computing is a promising paradigm capable of utilizing the pervasive mobile computational resources around us. Mobile users can exploit nearby mobile service to boost their computing power and therefore overcome the limitations of their own resources. In this paper, we first propose a framework named mobile service opportunistic network (MSON) and a QoS model for service composition to address the dynamic and mobile characteristic in mobile environment. Then formulate the service composition over MSON as an optimal problem and utilize improved Krill-Herd algorithm (IHK) to solve it. Lastly, a real-world opportunistic network data and QoS data is used to verify our approach's effectiveness and efficiency.

Index Terms—Mobile Computing, Mobile opportunistic network, Mobile Service Composition, Service-Oriented Architecture.

1 Introduction

RECENT YEARS have witnessed the rapid development of mobile devices and mobile communication. The number of mobile devices is still booming and it has already surpassed stationary Internet hosts. Web services are no longer limited to traditional stationary platforms and they can be more flexible and pervasive. The hardware of mobile devices will continue to make breakthroughs on extending the capabilities of mobile devices in terms of computational power, RAM, storage capacity, and so on [1]. The huge potential of mobile technology brings a great opportunity to traditional service computing in the mobile environment. As a result, the global interest of mobile service is on the rise and both academia and industry are inspired to pave the way for mobile service provisioning [2].

Nowadays, most of mobile applications are content-based (e.g., News app, video app, music app) and there are no need for every mobile user to request these content through expensive and power-hungry cellular network. While mobile device have powerful computing and communication capabilities now, the resources-intensive mobile applications, however, drain out the energy of the device much faster than before.

To achieve the goal of saving communication cost and reducing mobile device energy consumption, we propose a QoS-oriented mobile service composition approach in this paper, where a mobile user in mobile service opportunistic network can combine and exploit nearby devices' resources to boost their computing power and therefore overcome the limitations of their own resources [3]. As shown in Fig. 1, this architecture can reduce communication energy consumption and avoid the extreme centralization of traditional mobile cloud computing [3]. Its main rationality is three-fold. First, opportunistic user encounters are prevalent and sufficient in daily life [4], which offers plenty of opportunities to exploit nearby mobile worker for task solving [5]. Second, many mobile tasks require huge computational resources or data transfer (e.g., Tensorflow on mobile, Photoshop on mobile, Online video), for energy consumption and cost perspective, nearby mobile service provider are more adept at executing these tasks than the online workers, because this paradigm can reduce data transfer over cellular network which consume more energy than device to device (D2D) communications such as Bluetooth, WiFi and NFC [6]. Third, D2D communications are promising to replenish traditional cellular communications in terms of user throughput increase, cellular traffic reduction and network coverage extension, in this way, users can get better quality of service and save communication cost [7]. In a word, this framework shares the similar spirit with the emerging paradigm "cyber foraging" over opportunistic networks, such that mobile users opportunistically exploit nearby device resources to facilitate their computational task processing [8], [9], [10].

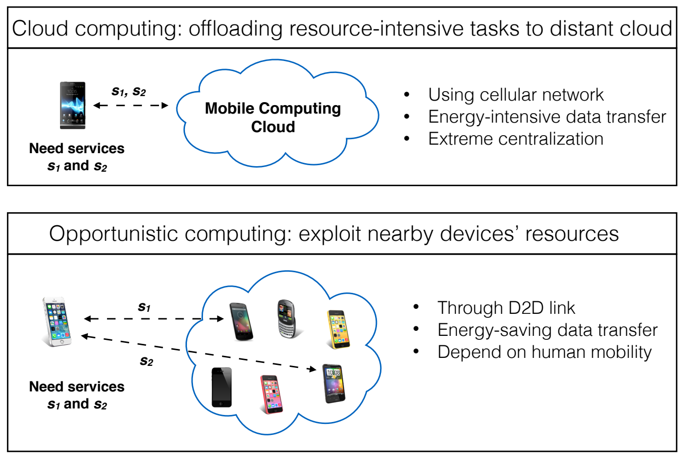


Fig. 1. Opportunistic computing

To address the aforementioned challenges and concerns, we propose a new approach for mobile service composition over opportunistic network. The main contributions are:

1) We propose a framework (mobile service opportunistic network MSON) to address the problem of service provision in the mobile encounter environment where both service requesters and providers are nonstationary. In such environment, mobile user can invoke service exposed by

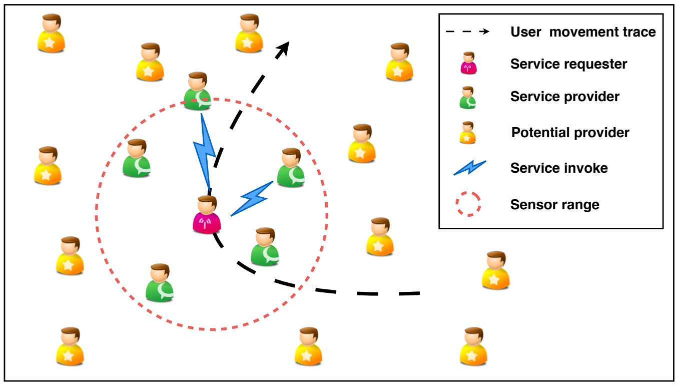


Fig. 2. Mobile service opportunistic network

nearby mobile devices through D2D links.

2) For MSON, we propose a mobile service QoS model for service provision which consider mobile service's availability as an important QoS attribute to capture user's mobility behavior.

3) Based on MSON and the proposed mobile service QoS model, we transfer the mobile service composition problem to an optimal problem and use an improved Krill-Herd algorithm (IKH) to solve it.

The remainder of this paper is structured as follows. Section II describe the MSON framework and its application scenario. Section III introduces the mobile service composition model. The approach to make service compositions is presented in Section IV. Section V presents experiments and evaluations. Section VI reviews the related work. Section VI concludes this paper.

2 MSON AND APPLICATION SCENARIO

In this section, we first introduce the characteristics of mobile service opportunistic network (MSON), then a user case is presented to illustrate its application scenario.

MSON has three main characteristics:

1) Locality: An MSON is not established through the stable Internet, it do not consider user mobility a problem but as an opportunity to exploit. Mobile user in MSON can perceive nearby service and establish self-organized local communication network within transmission distance.

2) Mobility: Services requesters and providers are not fixed at the same location, and they are mobile when invoking or provisioning a mobile service.

3) Nondeterminacy: Mobile services shared in MSON are note permanent because one node can enter or leave the data transmission range of other nodes at any time.

Fig. 2 illustrates the working procedure of the mobile service provision over MSON. In an opportunistic network scenario, a mobile service requester can perceive mobile service exposed by nearby devices through D2D links and launch a mobile composition request. A composer can be implemented and deployed on the requester's mobile device, which is in charge of discovering available mobile services nearby, selecting appropriate concrete service, and composing multiple services. During the execution of mobile service compositions, all concrete services interact with the composer directly [1].

Note that, our framework only considers one-hop mechanism for both service requester and provider, since some realistic dataset analyses reveal that users' one-hop neighbors are sufficient, compared with multi-hop mechanisms in existing researches [4], [5], [11], [12], [13], D2D communication which hops are larger than two would incur long delay [9], this one-hop feature can lower the network overhead (e.g., no need to transfer a large volume of task contents hop by hop) and ensure framework choose only local relatively reliable service.

We use an user case to illustrate the related features of service provision over MSON. Assume a mobile user Mike just complete his tour and now he is on the subway to airport. Now he wants to edit some videos he recorded and add some effects and share these video clips to his friends. But due to his mobile phone's limited battery, if he edit videos in his own mobile device, his mobile phone will run out of energy before he reach the airport. As one option, he can upload original videos to cloud and use cloud service to get all things done, but offloading quest into cloud will result in heavy cellular traffic, that means expensive communication fee and high energy consumption. If Mike participate in MSON and several video processing services is provided by some nearby mobile devices, Mike can invoke such mobile services on nearby mobile devices through D2D communication techniques. If these services cannot meet his requirement, several services can be composed. Due to users' mobility, the availability of service to Mike can vary, invoking mobile services provided by other users may face new challenges that traditional composition methods cannot handle. Thus, a mobile service composition model which can capture mobile services' availability need to be proposed.

3 MOBILE SERVICE COMPOSITION MODEL

In this section, we first give some basic concepts of mobile service composition, then introduce the concept of mobile service availability, finally we propose a specific QoS model for mobile service composition over MSON.

3.1 Preliminaries

In order to describe the problem addressed in this paper, we first provide the basic concepts of mobile service composition.

Definition 1 (Mobile Service): A mobile service can be represented as a two-triple $ms = (info, QoS)$, where:

1) $info$ is the description of a mobile service which include service name, functionality, parameters and result.

2) $QoS = \{q_{rt}, q_{price}, q_{ava}, ...\}$ is a set of quality attributes, including response time, price, availability, etc.

Definition 2 (MSON participant): A MSON participant is mobile service user who can be both service provider and requesters, it can be represented by a two-tuple $u = (P, C)$, where:

1) P is the set of mobile services exposed to other MSON participant.

2) C is the set of mobile services discovered from nearby MSON participant.

Definition 3 (Mobile Service Composition Plan): A service composition plan is a tuple $mscp = (T, R)$, where:

1) $T = \{t_1, t_2, ..., t_n\}$ is a set of tasks;

2) $R = \{d(t_i, t_j)|t_i, t_j \in T\}$ is a set of relations between tasks in T.

A service composition plan is an abstract description of a business process. Each task t_i can be realized by invoking an individual service. R is used to describe the structure of the composition. $d(t_i, t_j) = 1$ represents that the inputs of t_j depend on the outputs of t_i.

Definition 4 (composite service instance): A service composition instance csi can be represented as $csi = \{ms_{(1,j)}, ms_{(2,k)}, ..., ms_{(n,l)}\}$, where $ms_{(j,k)}$ is the selected concrete services. For example, $ms_{(1,j)}$ means the j-th service candidate is selected to execute $task_1$.

3.2 Concept of Mobile Service Availability

In MSON the availability of mobile service is highly related to the user's mobility. If user j moves outside the transmission range of its neighbouring user i, then user j is unreachable by user i and as a result the services on user j become unavailable to user i either. In this paper, user's mobility is utilized to calculate the mobile service availability [14].

Definition 5 (Mobile Service Availability): mobile service availability can be represented by a three-tuple (r, p, ava), where

1) r is the mobile service requester;

2) p is the mobile service provider;

3) ava is the mobile service availability value between requester and provider, $ava \in [0, 1)$, and $ava = 0$ means service provider moves out of transmission range.

As illustrated in Fig. 3, there are two mobile user i and j, assume user i is mobile service requester and user j is mobile service provider, both device have the same transmission range R. Each user moves randomly and it is assumed that the moving field is a circle with a radius of r. d is the distance between i and j. We use these three parameters (r, R, d) to calculate the availability of a mobile service. The transmission range of a node R is known (e.g., usually 10m for bluetooth, 25m for Wi-Fi).

Note that most of wireless transaction protocol have defined the RSSI (Received Signal Strength Indicator), then distance d between mobile user i and user j can be calculated by signal strength, Finally let us discuss how to calculate r.

The moving radius of a mobile user r is its moving speed v multiplied by the average service time t. Here t can be statistically calculated as the average value of last n times of service invoke, namely, $t = \Sigma_{i=1}^{n} t_i/n$. The speed of a mobile user v can be get through GPS data or other mobile sensor (e.g., Gyro-sensor), then $r = v \times t$.

Once we know these three parameters R, r, and d, the probability of user j staying inside the transmission range of user i (denoted as $P_{i,j}^{IN}$) can be calculated by

$$P_{i,j}^{IN} = \frac{S_{i \bigcap j}}{S_j} \tag{1}$$

Where $S_{i \bigcap j}$ is the area of the user j moving field inside the transmission range of user i, S_j is the overall area of the user j moving field. $S_{i \bigcap j}$ can be calculated as follow

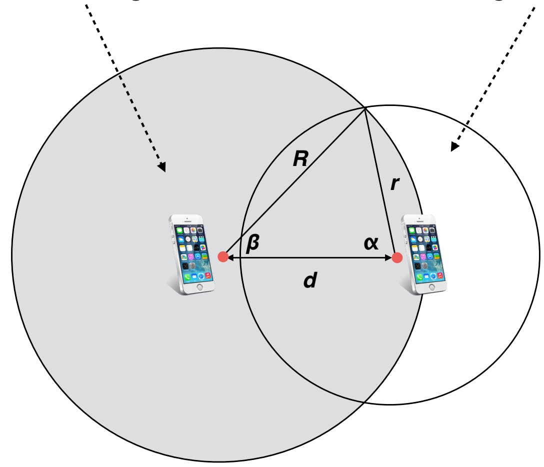


Fig. 3. Mobile service availability

$$\begin{aligned} S_{i \bigcap j} &= [(\frac{2\alpha}{2\pi} \times \pi r^2) - (\frac{r sin\alpha cos\alpha}{2} \times 2)] \\ &+ [(\frac{2\beta}{2\pi} \times \pi R^2) - (\frac{R sin\beta cos\beta}{2} \times 2)] \\ &= \alpha r^2 + \beta R^2 - (r^2 sin\alpha cos\alpha + R^2 sin\beta cos\beta) \end{aligned} \tag{2}$$

Where

$$\begin{aligned} \alpha &= arccos(\frac{r^2 + d^2 - R^2}{2r \times d}) \\ \beta &= arccos(\frac{R^2 + d^2 - r^2}{2R \times d}) \end{aligned} \tag{3}$$

S_j can be calculated by

$$S_j = \pi r^2 = \pi \times (s \times t)^2 \tag{4}$$

Therefore, the probability of user j staying inside the transmission range of user i (i.e., $P_{i,j}^{IN}$) can be calculated as follow

$$P_{i,j}^{IN} = \frac{S_{i \bigcap j}}{\pi s^2 t^2} \tag{5}$$

Suppose a mobile service ms running on mobile node j is a candidate service for a task requested by user i, and the availability of candidate service can be calculated as follow

$$\begin{aligned} q_{ava}(ms) &= P_{i,j}^{IN} \\ &= \frac{S_{i \bigcap j}}{\pi s^2 t^2} \\ &= \frac{\alpha r^2 + \beta R^2 - (r^2 sin\alpha cos\alpha + R^2 sin\beta cos\beta)}{\pi s^2 t^2} \end{aligned} \tag{6}$$

Mobile service availability can capture user's mobile behavior, and we use it as an important QoS attribute to construct QoS model for service composition in next subsection.

3.3 QoS Model for Mobile Service Composition

For mobile service requesters to select candidate service, QoS must be considered [15], [16], [17]. Generally, QoS attributes include response time, price, reliability, and reputation, we introduce mobile service availability as an important QoS attribute in this paper to describe user's mobility

TABLE 1
Examples of aggregation functions for QoS

Pattern	Resopnse Time	Availability
sequence	$\sum_{i=1}^{n} q_{rt}(S_i)$	$\prod_{i=1}^{n} q_{ava}(S_i)$
parallel	$Max\{q_{rt}(S_i)\}$	$\prod_{i=1}^{n} q_{ava}(S_i)$
choice	$\sum_{j=1}^{n} p_j \times q_{rt}(S_j)$	$\sum_{j=1}^{n} p_j \times q_{ava}(S_j)$
loop	$k \times q_{rt}(S_i)$	$[q_{ava}(S_i)]^k$

behavior. QoS attributes in this paper can be classified into two categories: positive (Q^+) and negative (Q^-). For positive attributes, larger values indicate better performance (e.g., reputation and availability), while for negative attributes, smaller values indicate better performance (e.g., response time and cost).

For a composite service instance csi, its each QoS attribute is determined by its concrete components and orchestration patterns. Table.1 lists the aggregation functions for response time and availability for sequential, loop, choice, and parallel composition patterns. We can find more aggregation functions found in [18] and [19].

In order to facilitate ranking of different composite service instances csi in terms of QoS, we utilize simple additive weighting (SAW) as the QoS utility function to map the QoS value into a real value. SAW first normalizes the QoS attribute values into real values between 0 and 1, through comparison with the maximal and minimal values; then it sums the normalized values multiplied with a preference weight w_t. According to SAW, the QoS utility of a csi can be calculated using e.q (7), where, $q_t(csi)$ is the aggregated value of the type-t QoS attribute of csi, and $q_{t,max}$ and $q_{t,min}$, respectively, denote the maximal and minimal possible aggregated values of the type-t QoS attribute [15].

$$\begin{aligned} U(csi) = & \sum_{q_t \in Q^-} \frac{q_{t,max} - q_t(csi)}{q_{t,max} - q_{t,min}} \times w_t \\ & + \sum_{q_t \in Q^+} \frac{q_t(csi) - q_{t,min}}{q_{t,max} - q_{t,min}} \times w_t \end{aligned} \quad (7)$$

3.4 Problem Formulation

Base on the above discussion, we can give the definition of the service composition over MSON problem.

Definition 6 (MSON Service Composition): Given a service composition request req by a mobile user u, perceive nearby service and select suitable concrete services to achieve an optimal service composition instance csi with the best QoS, that is

$$\begin{aligned} maximize \ & : U(csi) \\ subject\ to \ & : t_i \in \{1, 2, 3, ..., n\} \\ & \quad s_i \in \{1, 2, 3, ..., m\} \end{aligned} \quad (8)$$

where $U(csi)$ is the objective function mentioned in e.q (7), $t_i \in [1, n]$ is the index of the tasks in the composition plan, $s_i \in [1, m]$ is the index of service candidates for the i-th task.

Theorem 1: The service composition problem over MSON (Definition 6) is NP-hard.

Proof: We can reduce the service composition problem over MSON problem to a knapsack problem, and this problem can be solved by integer programming. The canonical form of integer linear program to search the optimal solution of this problem can be expressed as follow [20]:

$$\begin{aligned} maximize \ & : F(X) = c^T X \\ subject\ to \ & : Ax \leq b, \\ & \quad x_i \in \mathbb{Z}^n \\ & \quad x_i > 0 \end{aligned} \quad (9)$$

Where $F(x)$ is objective function, X is feasible solution, n is a positive integer, b and c are vectors, A is a matrix. For the problem of selecting optimal services composition over MSON, the vector $X = \{ms_{(1,j)}, ms_{(2,k)}..., ms_{(n,l)}\}$ can describe a feasible solution as a service composition with n tasks. The optimal solution X^* satisfies the following conditions:

1) X^* is a feasible solution.
2) for all feasible X, $F(X^*) \leq F(X)$.

The target of the mobile service composition problem over MSON is to find the biggest $F(x)$. Thus, the problem is equivalent to the integer programming problem which is known to be NP-hard. Then the service composition problem over MSON is NP-hard.

4 The IKH Algorithm for Mobile Service Composition

For the problem we formulate above, integer programming can be utilized to obtain the optimal solution. However, integer programming might cost much more time with the increment of problem size because of its poor scalability [21]. To solve this problem in polynomial time, an meta-heuristic algorithms such as GAs and PSO, can be utilized to find the near optimal solution.

Krill-Herd algorithm [22] is new generic stochastic optimization approach for the global optimization problem which is inspired by predatory behavior and communication behavior of krill. In this section, we will introduce an improved Krill-Herd algorithm (IKH) to solve the problem of mobile service composition over MSON.

4.1 Encoding

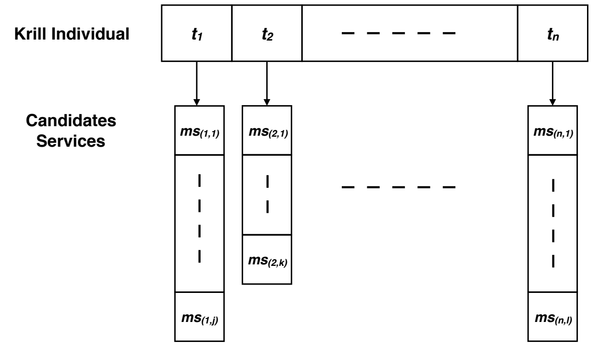


Fig. 4. Krill encoding

In IKH, composite service instance is encoded as a krill individual, the krill individual with the best position corresponds to the optimal mobile service composition. The target of algorithm is to find the krill individual with the best position, which means to find the best mobile service composition with the best QoS utility. Therefore, once the optimal krill individual is found, the best mobile service composition is obtained.

In this paper, the position vector of each krill individual is represented by an integer array with its length equal to the number of involved tasks. The i-th entry in the array, in turn, refers to the selection result of the task t_i . That is to say, given that the value of the i-th entry is j, it indicates that $ms_{(i,j)}$ is the selected concrete service to execute t_i. Fig. 4 illustrates this krill encoding.

4.2 Motion operator

Motion operator is the key component of IKH algorithm. As shown in e.q (10), the position of each krill individual is determined by three main factors: 1) motion influenced by other krill; 2) foraging action; 3) physical diffusion.

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{10}$$

where individual $X_i = \{ms_{(1,j)}, ms_{(2,k)}, ..., ms_{(n,l)}\}$ represents the i-th composition service instance in population, n is the number of tasks in the service composition, N_i, F_i, and D_i denote the motion influenced by other krill individuals, the foraging motion, and the physical diffusion, respectively.

1) Movement induced by other krill individuals

The motion induced by other krill individuals N_i means to learn from neighbor mobile service compositions. It can be formulated as follow:

$$N_i^{new} = N_{max}\alpha_i + \omega_n N_i^{old} \tag{11}$$

where

$$\alpha_i = \alpha^{target} + \alpha^{local} \tag{12}$$

α_i is the direction of the induced motion and it can be evaluated by target swarm density (target effect α^{target}), local swarm density (local effect α^{local}). N_{max} is the maximum induced speed, $\omega_n \in [0, 1]$ the inertia weight of the induced motion, N_i^{old} is the last induced motion influenced by other krill individuals.

2) Foraging Motion

Similarly, the foraging motion F_i is to learn from the current optimal composite service instance. F_i has two parts: the current food location and the information about the previous location. For the individual X_i, we can formulate this motion as follow

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{13}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{14}$$

where V_f is the foraging speed, $\omega_f \in [0, 1]$ is the inertia weight of foraging, and F_i^{old} is the last foraging motion. β_i is the direction of the foraging motion.

3) Random diffusion

For individual X_i, the physical diffusion is considered to be a random process. This motion includes two components: a maximum diffusion speed and a random directional vector, it can be formulated as follow

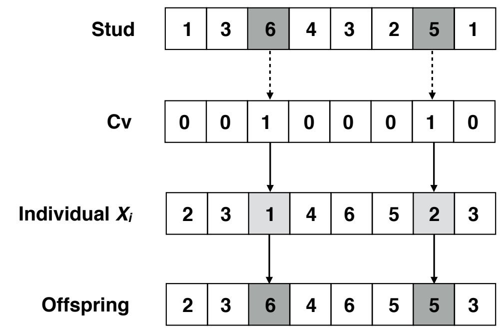


Fig. 5. Crossover operator

$$D_i = D_{max}\delta \tag{15}$$

where D_{max} is the maximum diffusion speed and $\delta \in [-1, 1]$ is a random directional vector.

4.3 Stud selection and crossover operator

The crossover operator plays an important role in Genetic algorithm for global optimization, we use this operator in IKH algorithm to enhance the search capability. The crossover operator in this paper is controlled by a dynamic crossover rate C_{rate} which can be obtain as follow

$$C_{rate} = Cr + (1 - Cr) \times \frac{U_{best} - U_i}{U_{best} - U_{worst}} \tag{16}$$

Where C_r is a pre-set fixed crossover rate, U_i is the i-th individual's QoS utility, U_{best} is the current best QoS utility value, similarly, U_{worst} is the current worst QoS utility value.

Then we can use C_{rate} to generate i-th individual's crossover vector $Cv = \{c_1, c_2, ..., c_n\}$, it can be manipulated as follows

$$c_i = \begin{cases} 1, & if \ \ rand(0,1) < C_{rate} \\ 0, & else \end{cases} \tag{17}$$

Inspired by SGA [23] (a type of GA which employs the optimal genome for crossover at each generation), we introduce stud selection procedure to improve IKH's search capability. From algorithm 1, we can see that for each individual X_i to crossover, we choose the optimal individual $Stud$ (i.e., the individual with highest QoS utility value) to mating. As shown in Fig. 5, the characteristics from individual $Stud$ are copied to individual X_i according to crossover vector Cv.

4.4 Update position

After crossover, the offspring should be evaluated and updated to current evolutionary sequence. According to the three motion actions, the time-relied position from time t and Δt can be formulated by the following equation

$$X_{i+1} = X_i + \Delta t \frac{dX_i}{dt} \tag{18}$$

where

$$\Delta t = C_t \sum_{j=1}^{n} (UB_j - LB_j) \tag{19}$$

Algorithm 1 Crossover operation

```
Input: Population X; Individual X_i to crossover; The number of tasks taskNumber;
Sort all krill individuals in population X by its QoS utility, get optimal individual Stud, save the best QoS utility value as U_best and the worst QoS utility value as U_worst
C_rate ← calcCrossoverRate(X_i, U_best, U_worst)
for i = 0 to taskNumber do
    r ← rand(0, 1)
    if r < C_rate then
        Cv[i] ← 1
    else
        Cv[i] ← 0
    end if
end for
for i = 0 to taskNumber do
    X_i[i] ← X_i ∧ (1 − Cv[i]) + Stud ∧ Cv[i]
end for
```

where n is the tasks number of composition service, UB_j and LB_j are upper and lower bounds of candidate services for the j-th task, respectively. C_t is a constant value to scale the searching space and we set it to $1/2n$ in this paper. Finally, the overall IKH algorithm process can be describe in Algorithm 2.

Algorithm 2 IKH algorithm

```
Input: Number of population size PS, Number of max iteration MI;
Generate initial population as X = (X_1, X_2, ..., X_PS)
Evaluate the QoS utility value of each krill individual in X
for i = 0 to MI do
    for i = 0 to PS do
        X'_i ← motionOperator()
        X''_j ← crossoverOperator(X'_j)
        U'_i ← evaluateQoSutility(X'_i)
        U''_i ← evaluateQoSutility(X''_i)
        if U''_i < U'_i then
            update position by equation (18) as X_{i+1}
        else
            accept X''_i as X_{i+1}
        end if
    end for
end for
Output the best solution
```

5 SIMULATION AND EVALUATION

In this section, we first discussed the experimental environment settings, and then the improved KH-based approach for the mobile service composition algorithm (IKH) is evaluated from the perspective of optimality and scalability, respectively.

5.1 Simulation Setting

To evaluate the optimality and scalability of the proposed approaches, the experiment is run on a personal computer with an Intel Core i5 CPU with 2.4 GHz, 4 GB RAM, macOS and Matlab R2015b Edition.

TABLE 2
User u 's D2D contract traces

Time	Available service provider
t1	Rabbit, Tony, S10, BlueRadios, NORTHOLT
t2	Tony, S10, Rabbit, NORTHOLT, BlueRadios
t3	Rabbit, NORTHOLT, BlueRadios, S10, Tony, Henrymobile, S4
t4	Tony, NORTHOLT, BlueRadios, S10, Rabbit, S4
t5	BlueRadios, S4, AliKatz, NORTHOLT, Rabbit, S25, S10
t6	S25, S10, NORTHOLT, BlueRadios, Rabbit
...	...

TABLE 3
Services exposed by provider

Service Provider	Exposed Service
AliKatz	ms_1, ms_2, ms_3, ms_4
BlueRadios	ms_1, ms_5
Henrymobile	ms_2, ms_4
NORTHOLT	ms_4, ms_5, ms_6
Rabbit	ms_1, ms_4
S4	ms_1, ms_2
S10	ms_6, ms_7
S25	ms_4, ms_5
Tony	ms_1, ms_4
...	...

Since we can not find available realistic datasets which involving both user D2D contact traces and quality of mobile service so far, we attempt to simulate the scenarios for mobile services provision by integrating realistic user D2D contact traces with quality of Web service datasets.

We consider MIT Reality dataset [24] as user D2D contact traces, where user location, Bluetooth devices in proximity, application usage, and phone status (such as charging and idle) were collected from 100 users over several months. This dataset can really reflect diverse network scenarios.

The publicly available quality of Web service (QWS) dataset [25] can be used to characterize the service candidates. This dataset consists of 4500 Web services from 142 users over 64 different time slices (at 15-minute interval) and each QoS data includes two measurements (response time and throughput).

Table 2 is part of D2D contract traces in MIT Reality dataset. For example, there there are five nearby devices within D2D transmission distance at time $t1$ and these devices can be regarded as MSON participant who provision mobile services. Table 3 shows MSON participants and the services they exposed to nearby devices. These mobile service are random chosen from QWS dataset. Table 4 is the Cartesian product of Table 2 and Table 3, it shows how many kinds of services user can exploit at a certain time and how many candidates for each kind of service (i.e., task). For example, there are five kinds of service available at time $t1$ and there are three candidates for task t_1, three candidates for task t_4, two candidates for task t_5, one candidate for task t_6 and one candidate for task t_7.

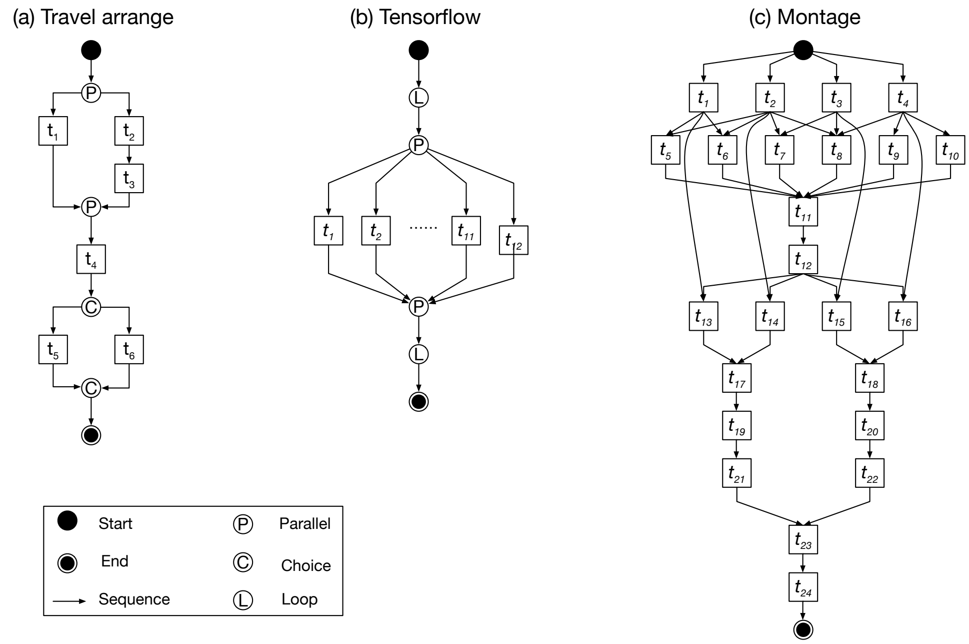


Fig. 6. The composition plan for case study.

TABLE 4
Available candidates

Time	Available service
t1	$ms_1^{(3)}, ms_4^{(3)}, ms_5^{(2)}, ms_6^{(1)}, ms_7^{(1)}$
t2	$ms_1^{(3)}, ms_4^{(3)}, ms_5^{(2)}, ms_6^{(2)}, ms_7^{(1)}$
t2	$ms_1^{(4)}, ms_2^{(2)}, ms_4^{(4)}, ms_5^{(2)}, ms_6^{(2)}, ms_7^{(1)}$
t4	$ms_1^{(4)}, ms_2^{(1)}, ms_4^{(3)}, ms_5^{(2)}, ms_6^{(2)}, ms_7^{(1)}$
t5	$ms_1^{(4)}, ms_2^{(2)}, ms_3^{(1)}, ms_4^{(4)}, ms_5^{(3)}, ms_6^{(2)}, ms_7^{(1)}$
t6	$ms_1^{(2)}, ms_4^{(3)}, ms_5^{(3)}, ms_6^{(2)}, ms_7^{(1)}$
...	...

Fig. 9 shows three mobile service composition plans for case study in this section. Fig. 9(a) is a well known composition plan for booking tickets [26], it has 6 tasks. Fig. 9(b) is a simple workflow with 12 tasks for Tensorflow [27], Tensorflow is a heterogeneous distributed system for machine learning and it already can be deployed in mobile devices. Fig. 9(c) is a scientific workflow with 24 tasks for Montage. Montage is an astronomical image mosaic engine, it can be used for simulating some picture edit application in mobile phone. We use these three kinds of composition plans to represent different meaningful service composition with different tasks.

5.2 Impact of Parameters

There are six parameters can be adjusted to improve the IKH's performance: population size PS, maximum iteration number MI, crossover rate Cr, foraging speed V_f, maximum induced speed N_{max} and physical diffusion speed D_{max}. As shown in Table 5, we generate six groups of parameters configuration to evaluate the impact of each parameter. For each group of parameters configuration, we tune one parameter and fix the other parameters. For each configuration setting, the IKH algorithm is executed 50 times independently and the average performance was recorded.

Fig. 5(a) shows the impact of population size, we observe that with the increase of population size, the average QoS utility of IKH significantly improved before $PS = 10$, and no significant improvement is observed after population size over 20. Therefore, an excessively large population size (e.g., $PS = 50$) has limited impact on the performance of IKH, and it will result in computing resources waste and high time cost. Similarly, Fig. 5(b) shows that the value of QoS utility significantly increased for higher number of iteration times until to a limit: $MI = 30$. Fig. 5(c) shows the impact of the crossover rate Cr. The performance of KH increases with Cr firstly, then decrease, the best performance is achieved for $Cr = 0.6$. Fig. 5(d) shows the impact of the foraging speed V_f. The performance of IKH increases with V_f firstly, then decrease, the best performance is achieved for $V_f = 0.8$. Fig. 5(e) shows the impact of the induced speed N_{max}. The performance of IKH increases with N_{max} firstly, then decrease, the best performance is achieved for $N_{max} = 0.3$. Fig. 5(f) shows the impact of the physical diffusion speed D_{max}. It shows that with the increase of D_{max} the performance of IKH fluctuation irregularly and slightly, without loss of generality, we random generate D_{max} from $[0.2 \sim 0.5]$ in this paper.

5.3 Optimality Evaluation

To verify the the capability of finding the optimal mobile service composition of our algorithms, we compare IKH with the basic GA algorithm, basic PSO algorithm and a brute-force algorithm.

Algorithm 1: Improved Krill-herd mobile service composition algorithm (IKH) which described in Section 4.

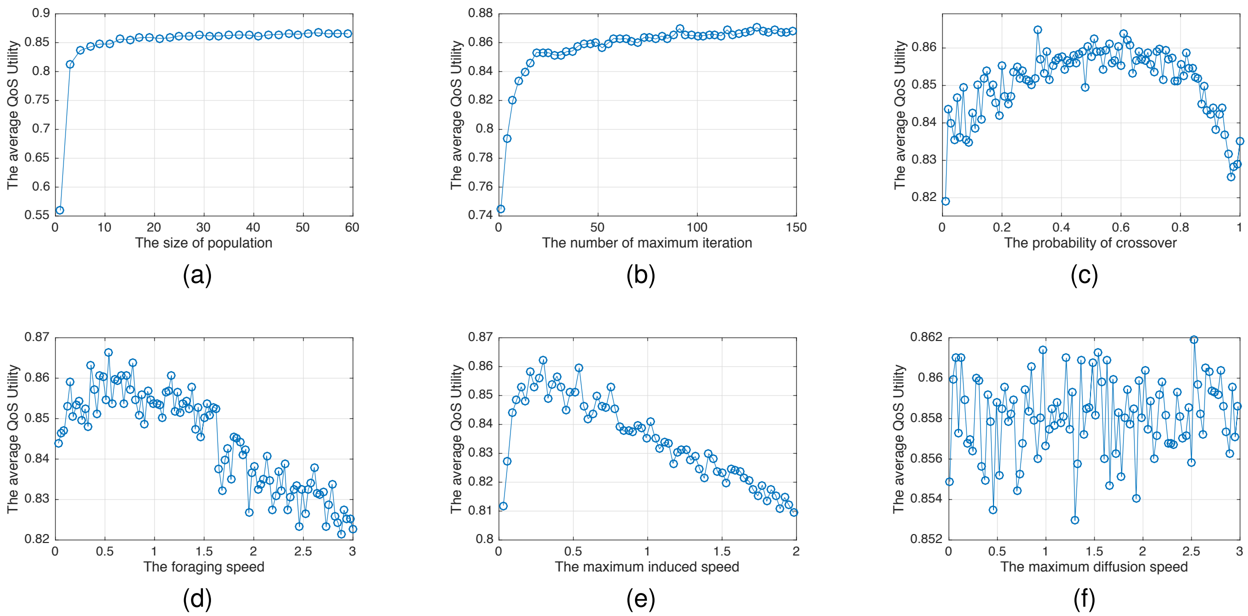


Fig. 7. Impact of parameters.

TABLE 5
parameters configuration

PS	MI	CR	V_f	N_{max}	D_{max}
1∼60	50	0.6	0.8	0.3	0.2
20	10∼150	0.6	0.8	0.3	0.2
20	50	0.01∼1.00	0.8	0.3	0.2
20	50	0.6	0.01∼3.00	0.3	0.2
20	50	0.6	0.01	0.01∼2.00	0.2
20	50	0.6	0.01	0.3	0.01∼3.00

TABLE 6
Parameters setting for different algorithms

Algorithms	Parameter Setting
GA	crossover rate = 0.7; mutate rate = 0.3
PSO	inertia weight = 0.8; c1 = 2.0; c2 = 2.0

Algorithm 2: A basic GA which uses two-point crossover and uniform mutation.

Algorithm 3: A basic PSO.

Algorithm 4: A brute-force algorithm that traverses all feasible composition to find the optimal solution.

To evaluate the optimality of above algorithms, we tune the parameters of each algorithm to achieve its best performance. The most suitable parameters are shown in Table 6.

The result in Fig. 7(a) shows that, as expected, the brute-force algorithm leads to the best performance with the highest QoS utility value. The basic PSO algorithm has the worst performance with the lowest QoS utility values. Both IKH and Ga can find over 95% optimal composition before 30 iterations and IKH performs little better than the basic GA algorithm. Fig. 7(a) and Fig. 7(b) shows that with the increasement of task number and service candidate number, our approach performs better than other algorithms. Although GA performs closely to IHK, IHK's superiority is it runs more faster than GA, in next subsection we will evaluate the scalability of above four algorithms.

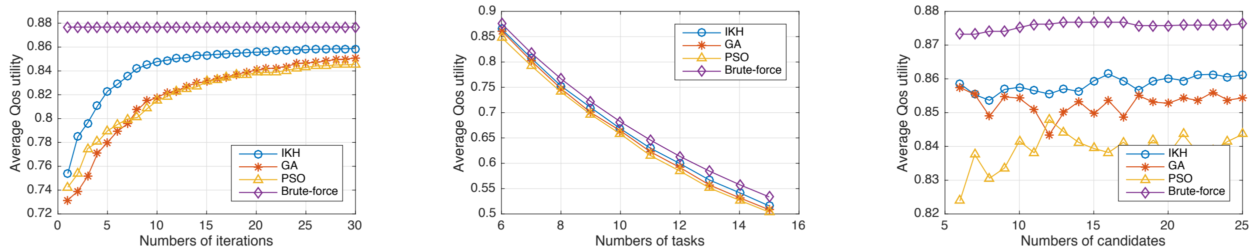
5.4 Scalability Evaluation

In this experiment, we compared the run time of IKH with the other algorithms to evaluate the scalability of our algorithm.

In order to evaluate the impact of task number on the execution time of our algorithm, The task number is varied from 6 to 25. For each task number, the simulation is repeated 50 times, and the average performance was recorded as the result. We observe from Fig. 8(a) that with the increasement of task number, the run time of brute-force method is almost exponential to the problem size, while the run time of other three meta-heuristic algorithms (IKH, GA, PSO) are increases almost linearly with tasks increasing. In mobile environment, due to the dynamic movement and limited of computing resources, algorithm must meet the requirement of finding feasible composition within a short time. Therefore, although brute-force can obtain the optimal result, it is not suitable to the problem due to its poor scalability.

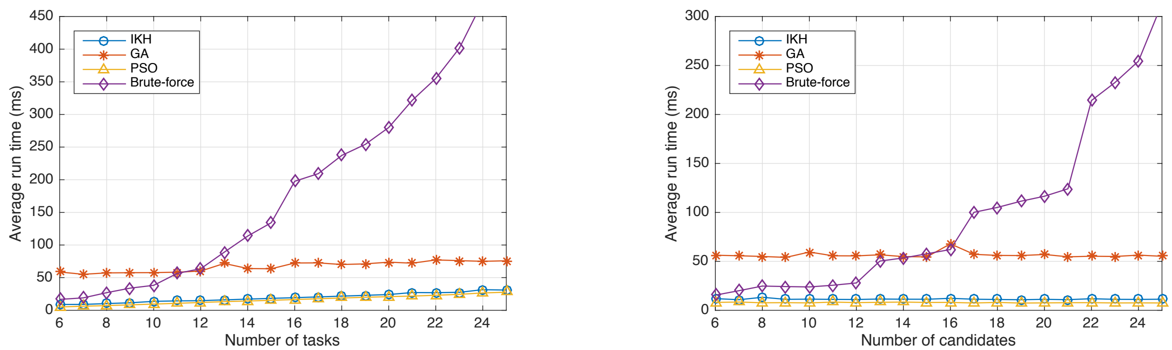
We also evaluate the impact of candidate services per task, In this experiment, we vary candidate service number from 6 to 25. As shown in Fig. 8(b), the number of candidate services does not affect the execution time of the three meta-heuristic algorithms, because in the initialization step, the scale of the algorithm is determined, and it is only affected by the algorithm parameters. Similarly, brute-force method is also exponential to the problem size.

Among all the algorithms, PSO has the lowest run time, then IKH, finally GA. But PSO's poor optimality makes it unsuitable for this problem, IKH runs almost five times faster than GA. Based on the experimental results, we can



(a) (b) (c)

Fig. 8. Optimality evaluation.



(a) Case I (b) Case II

Fig. 9. Scalability evaluation.

conclude that IKH maintains acceptable performance (optimality and scalability) with large data sets for both tasks and candidates.

6 RELATED WORK

Service-oriented computing (SOC) is a novel paradigm to develop and integrate enterprise information system [28], with the development of mobile device and communication technology, the research of mobile service composition has grain much attention from both industry and academia. In this section, we first briefly review some recent work on mobile service composition, then review opportunistic network and its application in mobile devices.

6.1 mobile service composition

Mobile service computing is the combination of service computing and mobile computing, With the development of mobile device and application industry, more and more study have emerged to address the problem of mobile service composition. Deng et al. [29] presented an detailed introduction to mobile service computing, they first discussed the limitations of mobile computing, then classify mobile service computing into three categories: C2M, M2M, Hybrid. They also discussed the challenge toward mobile service provision and mobile service consumption in terms of performance, energy and security perspective. At their [1] work, they proposed a mobile service sharing community to address the problem of service provision in mobile environment. They extent the random way point (RWP) model to capture user mobility and utilize the meta-heuristic algorithm to solve the service composition problem. Yang et al. [14] presented a comprehensive QoS model specifically for pervasive services. They considered not only mobile wireless network characteristics but also user-perceived factors, and devised a corresponding formula to calculate the QoS criterion. Zhang2016 zhang et al. [30] gives a context-aware service selection algorithm based on Genetic Algorithm to solve the problem of mobile service selection, they introduce a tree-encoding method to improve the capacity and efficiency of GA. However, this work did not consider user mobility. Wang et al. [31] solve the problem of dependable service composition in wireless mobile ad hoc networks by taking the mobility prediction of the service providers into consideration. They use a probability-free model and a probabilistic model to characterize the uncertainty to compose a service that can tolerate the uncertain mobility of service provider. However, this work only focus on the case of sequential service workflows and the heuristic algorithms they presented does not seek the optimal QoS service compositions.

6.2 mobile opportunistic network

Opportunistic networking is one of the most interesting evolutions of the multi-hop networking paradigm. Instead of constructing stable end-to-end paths as in the Internet, opportunistic networks do not consider node mobility a problem but as an opportunity to exploit. Marco et al. [32] give a review of opportunistic network and regarded it as the first step in people-centric networking, they also discuss the focused research problem such as mobility model and routing problem. Turkes et al. [33] proposed a middleware named Cocoon to support mobile opportunistic network, they design a routing protocol above Wi-Fi and Bluetooth standards, their experiments which use real-world data setups show that Cocoon performs well on the aspects of dissemination rate, delivery latency and energy consumption. Fortuna et al. [34] presented an review of dynamic service composition over both wired and wireless environment, However, their work does not present any technical details to describe how to composite service in mobile networks. Giordano et al. [3] proposed a novel paradigm that utilize Opportunistic computing as an appealing complement to the mobile computing cloud, in this way, mobile device can combine and exploit heterogeneous resources from other devices. Pu et al. [35] presented QoS-oriented self-organized mobile crowdsourcing framework, in this work, the prevalent and sufficient characteristics of opportunistic user encounters in our daily life are utilized to solve crowdsourcing problem.

7 CONCLUSIONS AND FURTHER STUDIES

In this paper, we propose a comprehensive framework for optimal mobile service compositing on mobile environment. We present a mobile service opportunistic network model (MSON) that fully integrates human mobility behavior factors for mobile service provisioning and introduce a QoS model for mobile service composition. Then we formulate the mobile service composition over MSON problem as an optimal problem to maximize the quality of service composition, propose an improved Krill-herd (IKH) algorithm to solve it, and a case study based on real-world opportunistic network and some well-known web service dataset show that our proposed approach outperforms current standard composition methods in mobile environments.

We plan to consider the following topics for future work: 1) Some prediction methods (e.g., hidden Markov model and neural networks) can be used to predict user future's movement to formulate a better user mobility model; 2) more QoS metrics (e.g., service price and service reputation) are supposed to be analyzed and blends into our QoS model; 3) this work doesn't consider Service-Level-Agreement (SLA) constrains. We intend to consider SLA constraints and introduce corresponding algorithms to generate better composition.

REFERENCES

[1] S. Deng, L. Huang, J. Taheri, J. Yin, M. Zhou, and A. Y. Zomaya, "Mobility-aware service composition in mobile communities," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 555–568, 2017.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] S. Giordano and D. Puccinelli, "The human element as the key enabler of pervasiveness," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*. IEEE, 2011, pp. 150–156.

[4] S. Liu and A. D. Striegel, "Exploring the potential in practice for opportunistic networks amongst smart mobile devices," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 315–326.

[5] W. Chang and J. Wu, "Progressive or conservative: Rationally allocate cooperative work in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 2020–2035, 2015.

[6] R. Balani, "Energy consumption analysis for bluetooth, wifi and cellular networks," *Online Httpnesl Ee Ucla Edufwdocumentsreports2007PowerAnalysis Pdf*, 2007.

[7] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[8] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2012, pp. 145–154.

[9] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Infocom, 2014 proceedings ieee*. IEEE, 2014, pp. 1060–1068.

[10] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.

[11] G. S. Tuncay, G. Benincasa, and A. Helmy, "Participant recruitment and data collection framework for opportunistic sensing: a comparative analysis," in *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*. ACM, 2013, pp. 25–30.

[12] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2319–2327.

[13] C. Jiang, L. Gao, L. Duan, and J. Huang, "Exploiting data reuse in mobile crowdsensing," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[14] K. Yang, A. Galis, and H.-H. Chen, "Qos-aware service selection algorithms for pervasive service composition in mobile wireless environments," *Mobile Networks and Applications*, vol. 15, no. 4, pp. 488–501, 2010.

[15] Q. Wu, F. Ishikawa, Q. Zhu, and D.-H. Shin, "QoS-Aware Multigranularity Service Composition: Modeling and Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 11, pp. 1565–1577, 2016.

[16] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[17] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing qos data via aggregating nonnegative latent factor models," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 3, pp. 524–537, 2016.

[18] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "Qos aggregation for web service composition using workflow patterns," in *Enterprise distributed object computing conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*. IEEE, 2004, pp. 149–159.

[19] H. Zheng, W. Zhao, J. Yang, and A. Bouguettaya, "Qos analysis for web service compositions with complex structures," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 373–386, 2013.

[20] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[21] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, vol. 20, pp. 8–12, 1988.

[22] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[23] W. Khatib and P. J. Fleming, "The stud ga: a mini revolution?" in *International Conference on Parallel Problem Solving from Nature*. Springer, 1998, pp. 683–691.

[24] N. Eagle and A. S. Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.

[25] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating qos of real-world web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.

[26] Q. Wu and Q. Zhu, "Transactional and qos-aware dynamic service composition based on ant colony optimization," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1112–1119, 2013.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[28] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. IEEE, 2003, pp. 3–12.

[29] S. Deng, L. Huang, H. Wu, W. Tan, J. Taheri, A. Y. Zomaya, and Z. Wu, "Toward Mobile Service Computing: Opportunities and Challenges," *IEEE Cloud Computing*, vol. 3, no. 4, pp. 32–41, 2016.

[30] C. Zhang, L. Zhang, and G. Zhang, "Qos-aware mobile service selection algorithm," *Mobile Information Systems*, vol. 2016, 2016.

[31] J. Wang, "Exploiting mobility prediction for dependable service composition in wireless mobile ad hoc networks," *IEEE Transactions on Services Computing*, vol. 4, no. 1, pp. 44–55, 2011.

[32] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014.

[33] O. Turkes, H. Scholten, and P. J. Havinga, "Cocoon: A lightweight opportunistic networking middleware for community-oriented smart mobile applications," *Computer networks*, vol. 111, pp. 93–108, 2016.

[34] C. Fortuna and M. Mohorcic, "Dynamic composition of services for end-to-end information transport," *IEEE Wireless Communications*, vol. 16, no. 4, 2009.

[35] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowd foraging: A qos-oriented self-organized mobile crowdsourcing framework over opportunistic networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 848–862, 2017.