QoS-Oriented Mobile Service Composition Over Oppotunistic Networks

Qinglan Peng, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—The abstract goes here.

Index Terms—Computer Society, IEEE, IEEEtran, journal, LaTeX, paper, template.

1 Introduction

Recent years have witnessed the rapid development of mobile devices and mobile communication technology, mobile devices has already surpassed stationary Internet hosts in numbers and 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], [3].

While mobile device have powerful computing and communication capabilities now, the resources-intensive services, however, drain out the energy of the device much faster than before. To achieve the goal of reducing mobile device energy consumption, we propose a QoS-oriented mobile service composition approach in this paper, where a mobile user in mobile opportunistic network can combine and exploit nearby devices' resources to boost their computing power and overcome the limitations of their own resources. As shown in fig.1, this architecture can reduce communication energy consumption and avoid the extreme centralization of traditional mobile cloud computing [**?**]. 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], [6], [7]. 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 [**?**]. 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 fee [8]. 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 [9], [10], [11].

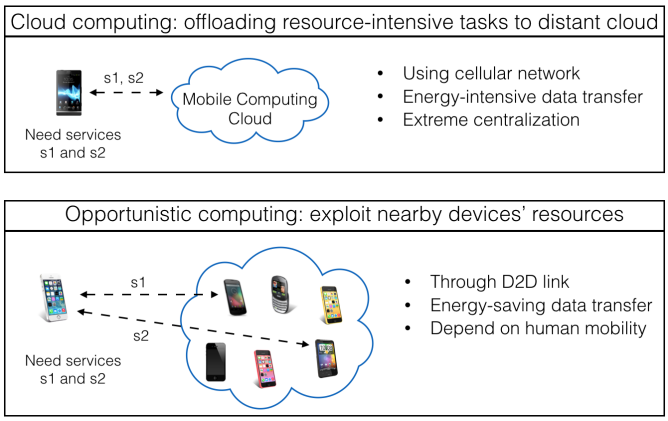


Fig. 1. Opportunistic computing

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 nearby mobile devices through D2D links.

2) For MSON, we propose a mobile service QoS model for service provision which consider mobile service 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 optimization problem and use the Krill-Herd algorithm (KH) to solve it. A series of evaluations have

- *M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: see http://www.michaelshell.org/contact.html*
- *J. Doe and J. Doe are with Anonymous University.*

Manuscript received April 19, 2005; revised August 26, 2015.

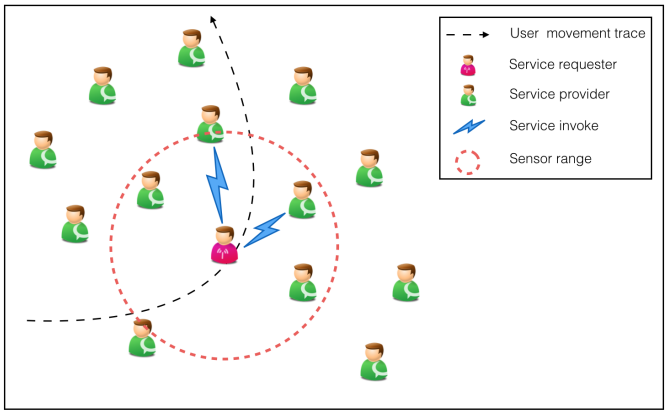


Fig. 2. Mobile service opportunistic network

been conduct to validate the optimality and scalability of our algorithm, and it shows our algorithm can get approximately optimal solution and better performance than other standard composition approaches.

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 mobile service opportunistic network (MSON), then an example 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 opportunistic network can discover nearby service and establish self-organized local communication network within sensor 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: MSON participants are not stationary, 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 opportunistic network. In an opportunistic network scenario, a mobile service requester R can discover mobile service exposed by nearby devices through D2D link 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], [12], [13], [14], [15], [16], D2D communication which hops are larger than two would incur long delay [10], 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 example 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 with his friends. But due to mobile devices' limited battery, if he edit videos in his own mobile device, his mobile phone will run out of energy before he reach the destination. 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, we will discuss mobile service availability in next section [17].

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 opportunistic network.

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 three-triple $ms = (id, info, QoS)$, where:

1) id is the unique identifier of the service;

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

3) $QoS = \{q\}_{j=1}^{n}$ 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 three-tuple $u = (id, P, C)$, where:

1) id is the unique identifier of a MSON participant;

2) p is the set of mobile services exposed by mobile service user u.

3) c is the set of discovered mobile services from nearby mobile service providers.

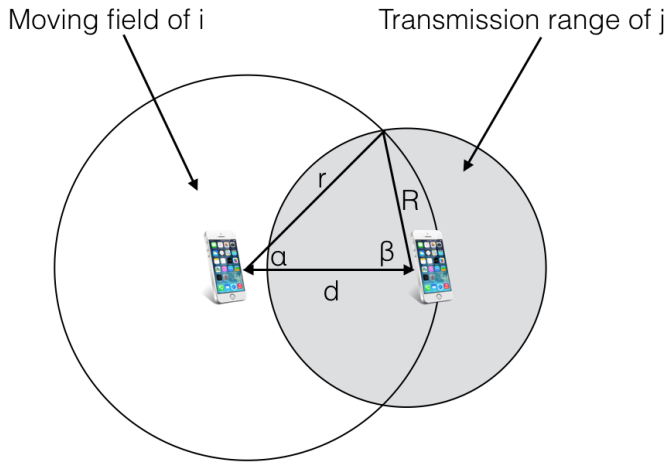


Fig. 3. Mobile service availability

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 is a tuple $csi = (mscp, S)$, where:

1) $mscp$ is mobile service composition plan which defined in definition 3;

2) $S = s_1, s_2, , s_n$ is a set of selected concrete services.

3.2 Concept of Mobile Service Availability

In mobile service opportunistic network (MSON) the availability of mobile service is highly related to the users mobility. If user i moves outside the transmission range of its neighbouring user j, then user i is unreachable by user j and as a result the services on user i become unavailable to user j either. Here user mobility is utilized to calculate the mobile service availability [18].

Definition 5 (Mobile Service Availability): mobile service availability can be represented by a three-tuple (i, 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 r and provider p, $ava \in [0, 1)$, and $ava = 0$ means service provider p moves out of transmission range.

As illustrated in Fig.3, there are two mobile user i and j whose 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., pre-defined or changing according to certain algorithm). Suppose that the location of each mobile user is known (e.g., via GPS data or other location-based services provided by telecommunications service providers [19]), then distance d can be calculated using the Euclidean distance formula, i.e.,$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ where (x_i, x_j) and (y_i, y_j) are the coordinates of user i and j respectively. Finally let us discuss how to calculate r [18].

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 calculated based on its moving distance during a period from t_1 to t_2 [20], namely: $s = \sqrt{(x_i - x_j)^2 + y_i - y_j^2}/(t_2 - t_1)$, then $r = s \times t$.

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

$$P_{i,j}^{IN} = \frac{S_i^{IN}}{S_i^T} \tag{1}$$

Namely, $P_{i,j}^{IN}$ equals to the area of the user i moving field inside the transmission range of user j (denoted as S_i^{IN}) divided by the overall area of the user i moving field (S_i^T) .

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

Then,

$$\begin{aligned} S_i^{IN} &= [(\frac{2\beta}{2\pi}\pi R^2) - (\frac{Rsin\beta cos\beta}{2}2)] \\ &+[(\frac{2\alpha}{2\pi}\pi r^2) - (\frac{rsin\alpha cos\alpha}{2}2)] \\ &= \beta R^2 + \alpha r^2 - (R^2 sin\beta cos\beta + r^2 sin\alpha cos\alpha) \end{aligned} \tag{3}$$

There is also

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

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

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

Suppose a mobile service s running on i is a candidate service for a task requested by user j, and the availability of candidate service s can be denoted as $q_{ava}(s)$, that is

$$\begin{aligned} q_{ava}(s) &= P_i^{IN} \\ &= \frac{A_i^{IN}}{\pi s^2 t^2} \\ &= \frac{\beta R^2 + \alpha r^2 - (R^2 sin\beta cos\beta + r^2 sin\alpha cos\alpha)}{\pi s^2 t^2} \end{aligned} \tag{6}$$

Mobile service availability $q_{ava}(s)$ 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 [21], [22], [23]. 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	$\sum_{i=1}^{n} q_{time}(S_i)$	$\sum_{i=1}^{n} q_{price}(S_i)$
parallel	$Max\{q_{time}(S_i)\}$	$\sum_{i=1}^{n} q_{price}(S_i)$
choice	$\sum_{j=1}^{n} p_j \times q_{time}(S_j)$	$\sum_{j=1}^{n} p_j \times q_{price}(S_j)$
loop	$k \times q_{time}(S_i)$	$k \times q_{price}(S_i)$

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) [24].

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, cost, and availability for sequential, loop, choice, and parallel composition patterns. We can find more aggregation functions found in [25] and [26].

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 t-th QoS attribute of csi, and $q_{t,max}$ and $q_{t,min}$, respectively, denote the maximal and minimal possible aggregated values of the t-th QoS attribute [24].

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

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 provided to achieve an optimal service composition instance csi with the best QoS, that is

$$\begin{aligned} target &: max\ U(csi) \\ subject\ to &: i \in \{1,2,3,...,n\} \\ & x_i \in \{1,2,3,...,m\} \end{aligned} \tag{8}$$

where $U(csi)$ is the objection mentioned in e.q (7), $i \in [1,n]$ is the index of the tasks in the composition plan, $x_i \in [1,m]$ 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 our studied problem to a knapsack problem, and this problem can be solved by integer programming. The standard integer program to find the smallest value of a given objective function $F(\Theta)$ with a feasible parameter follows [27]:

$$\begin{aligned} &inf\ F(\Theta) \\ &subject\ to:\ \theta_i \in \{1,2,3,...,N\} \end{aligned} \tag{9}$$

For the problem of selecting optimal services composition over MSON, the vector $\Theta = (\theta_1, ..., \theta_n)$ can describe a possible solution as a service composition with n tasks. An element θ_i in corresponds to a selected service from the candidates for the i-th task. The optimal solution $\Theta*$ satisfies the following conditions:

1) $\Theta*$ belongs to the feasible set.

2) $\forall\Theta, F(\Theta*) \leq F(\Theta)$.

The target of the mobile service composition problem in MSON is to find to obtain the biggest $F(\Theta)$. 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 Composition Algorithm

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 [28]. 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. In this section, we will illustrate our algorithm for making mobile service compositions over MSON based on the Krill-Herd algorithm.

KH algorithm [29] is new generic stochastic optimization approach for the global optimization problem which is inspired by predatory behavior and communication behavior of krill. In this paper, the position vector of each krill individual corresponds to a feasible mobile service composition, the foraging motion is to learn from the current optimal service composition. Similarly, the motion induced by other krill individuals means to learn from neighbor mobile service compositions. The krill individual with the best position corresponds to the optimal mobile service composition. The KH optimization's target is to find the krill individual with the best position, which means to find the best mobile service composition with the best fitness value. Therefore, once the optimal krill individual is found, the best mobile service composition is obtained.

As shown in equation (10), the position of a krill individual is determined by three main factors: 1) foraging action; 2) movement influenced by other krill; and 3) physical diffusion.

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

where $csi_i = (s_{i,1}, s_{i,2}, ..., s_{i,n})$ is i-th composition service instance (csi), n is the number of tasks in the service composition, $s_{i,j}$ is the selected candidate for the j-th task in solution csi_i, where N_i, F_i, and D_i denote the motion influenced by other csi, the foraging motion, and the physical diffusion of the csi_i, respectively.

1) Movement induced by other krill individuals

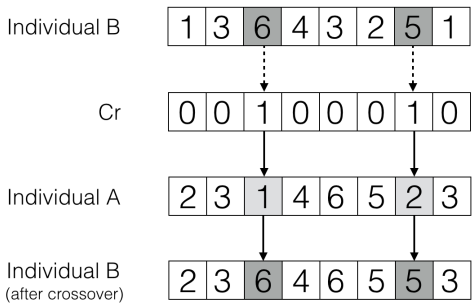


Fig. 4. Cross over

Motion induced by other composition service instance N_i can be formulated as follow

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

where

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

α_i is 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 csi.

2) Foraging Motion

Foraging Motion F_i covered two parts: the current food location and the information about the previous location. For the csi i, we formulated this motion below:

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

where

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

where V_f is the foraging speed (empirically set to 0.02 in this paper), $\omega_f[0,1]$ 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 the i-th csi, 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 follows

$$D_i = D^{max}\delta \tag{15}$$

where D^{max} is the maximum diffusion speed and $\delta \in [-1,1]$ is a random directional vector. In this paper, the maximum diffusion speed is randomly generated in $[0.002, 0.01]$.

4) Crossover operator

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

$$C_{rate} = 0.8 + 0.2 \times \frac{F_{best} - F_i}{F_{best} - F_{worst}} \tag{16}$$

Where C_{rate} is the i-th individual's crossover rate, F_i is the i-th individual's fitness value, F_{best} is the best fitness value so far, similarly, F_{worst} is the worst fitness value.

Then we can use C_{rate} to generate i-th individual's crossover vector $C_r = \{x_1^m, x_2^m, ..., x_n^m\}$, the j-th component of i-th individual x_j^m is manipulated as:

$$x_j^m = \begin{cases} 1, & rand(0,1) < C_{rate} \\ 0, & other \end{cases} \tag{17}$$

For each individual A to crossover, we randomly choose an target individual B from current iteration and the characteristics from Y are copied to X according to C_r, the crossover operator is shown in fig.4.

Algorithm 1 Crossover operation

Input: Population X; Best fitness F_{best}; Worst fitness F_{worst}; Individual A to crossover;

```
C_rate ← calcCrossoverRate(A, F_best, F_worst)
for i = 0 to taskNumber do
    r ← rand(0,1)
    if r < C_rate then
        C_r[i] ← 1
    else
        C_r[i] ← 0
    end if
end for
B ← randomSelect(X)
for i = 0 to taskNumber do
    A[i] ← A&(1 − C_r[i]) + B&C_r[i]
end for
```

5) Update position

According to the three motion actions, the time-relied position from time t to δt can be formulated by the following equation:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \tag{18}$$

where

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

where d is the tasks number of each csi, 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. The whole process of KH algorithm is shown in Algorithm 2.

5 SIMULATION AND EVALUATION

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

Algorithm 2 KH algorithm

Input: Lower bound of service candidates lb; Up bound of service candidates ub; Number of krill individuals nk; Number of max iteration mi
Output: Individual with best fitness csi

generate nk feasible solutions randomly
save them in the population X
$X_{best} \leftarrow$ bestRank(X)
for $i = 0$ **to** mi **do**
  $X_f \leftarrow$ foodLocation(X)
  if $X_f < X_{f(previous)}$ **then**
    $X_f \leftarrow X_{f(previous)}$
  end if
  for $j = 0$ **to** nk **do**
    $d_{food} \leftarrow$ distance(X_j , X_f)
    $d_{best} \leftarrow$ distance(X_j , X_{best}) {M}ovement Induced
    // Movement Induced
    $\alpha^{target} \leftarrow$ bestKrillEffect(X_{best}, d_{best})
    $ds \leftarrow$ neighborDistance(X)
    $\alpha^{local} \leftarrow$ neighborEffect(X, ds)
    $N_i \leftarrow$ calcMovementInduced(α^{local}, α^{target})
    // Foraging Motion
    $\beta_{food} \leftarrow$ foodAttraction(X, d_{food})
    $\beta_{best} \leftarrow$ bestPsitionAttraction(X, d_{best})
    $F_i \leftarrow$ calcForagingMotion(V_f, β_{food}, β_{best})
    // Physical Diffusion
    $D_i \leftarrow$ physicalDiffusion(X, X_{best})
    $D_x \leftarrow$ calcMotionProcess(N_i, F_i, D_i)
    Crossover(X, X_{best})
    UpdatePosition(X, D_x)
  end for
end for
return bestRank(X)

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.

Science 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 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 such as a sparse campus and dense conference.

The publicly available quality of Web service (QWS) dataset [30] 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), for a mobile service ms, response time attribute is randomly selected from the QWS dataset.

TABLE 2
PARAMETERS CONFIGURATION

Configuration	PS	MI	V_f	D_{max}
configuration-1	10∼100	50	0.2	0.2
configuration-2	50	10∼100	0.2	0.2
configuration-3	50	50	0.01∼1.00	0.2
configuration-4	50	50	0.2	0.01∼1.00
configuration-5	50	50	0.2	0.2

5.2 Impact of Parameters

There are five parameters can be adjusted to improve the KHMSC's performance: population size PS, maximum iteration number MI, foraging speed V_f, maximum diffusion speed D_max and maximum induced speed N_max. As shown in Table 2, we generate five 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 KHMSC algorithm is executed 50 times independently and the average performance was recorded.

Figure 5 shows the results of tuning different parameters for our approach.

From Figure 5(a), we observe that with the increase of the population size, the quality of KHMSC solutions is also improved (answers with higher fitness values are found). This is because a larger population size enlarges the overall searching field and improves the probability of finding superior service composition.However, when the population size exceeds a certain value, e.g., N = 50, no significant improvement is observed. Therefore, an excessively large population size has little impact on improving the performance of KHMSC. Similar result occurs in Figure 5(b), which shows the impact of the maximum number of iterations, where the quality of the best solution is increased for higher number of iterations up to a limit: I = 60 in this case.

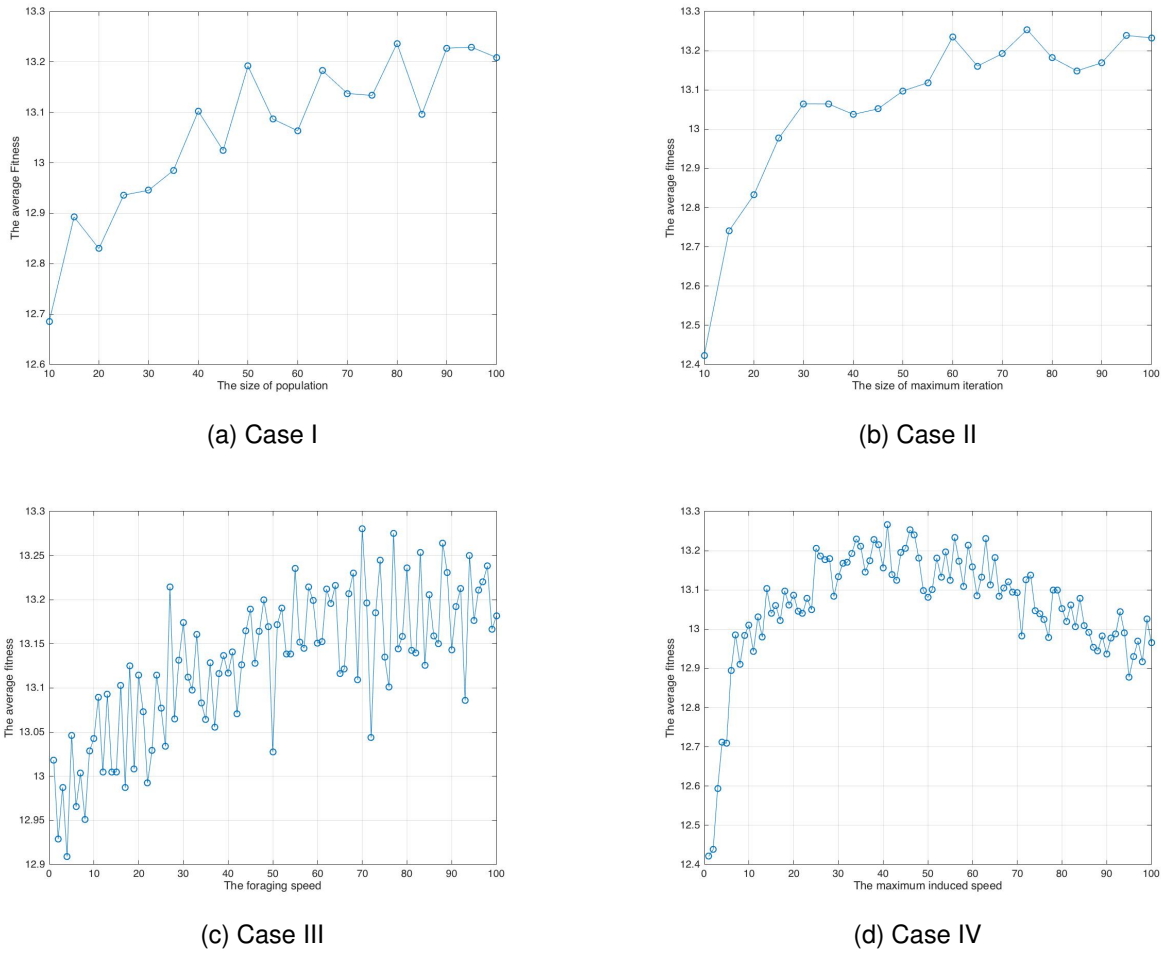
Fig. 5(c) shows the impact of the foraging motion speed V_f. The performance of KHMSC increases with V_f. The best performance is achieved for $V_f = 0.8$. As V_f increases, the performance of KHMSC increases mainly because when the the foraging motion speed is large, early convergence to a local optimum can be avoided.

Fig. 5(d) shows the impact of the inertia weight of the induced motion N_max. It shows that solution quality is improved up to a limit (N_max = 0.4) and then decreases afterward. This observation shows that although higher values of N_max can lead to better population diversity, their very high values lead to chaotic updating of positions in which the induced motion is not toward appropriate krill individuals.

5.3 Optimality Evaluation

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

Algorithm 1: KH-based mobile service composition algorithm (KHMSC) which described in Section 4.



(a) Case I (b) Case II

(c) Case III (d) Case IV

Fig. 5. Simulation results for the network.

TABLE 3
PARAMETER SETTING OF 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 3.

The result in Figure 6 shows that, as expected, the brute-force algorithm leads to the best performance with the highest fitness. The basic PSO algorithm has the worst performance with the lowest fitness values. Both KHMSC and Ga can find over 90% optimal composition before 50 iterations and KHMSC performs little better than the basic GA algorithm. Although GA performs closely to KHMSC, we finally choose KHMSC because it runs almost five times faster than GA, in next subsection we will evaluate the scalability of above four algorithms.

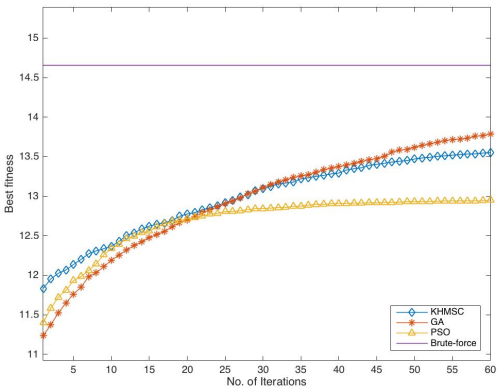
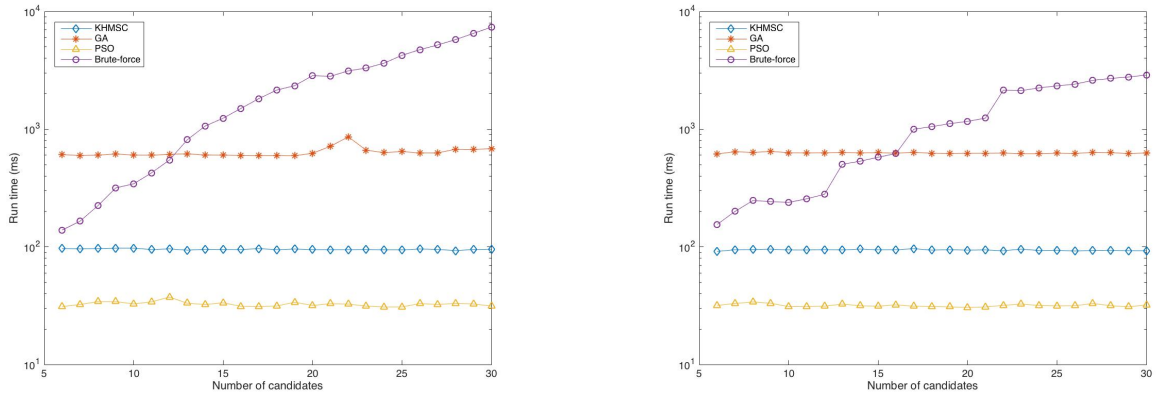


Fig. 6. Optimality Evaluation

5.4 Scalability Evaluation

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



(a) Case I (b) Case II

Fig. 7. Scalability Evaluation.

In order to evaluate the impact of task number on the execution time of our algorithm, The task number is varied from 6 to 30. For each task number, the simulation is repeated 50 times, and the average value is recorded as the result. We observe from figure 7(b) that the run time of brute-force method is almost exponential to the problem size, while the run time of other three meta-heuristic algorithms (KHMSC, 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 30. As shown in figure 7(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 KHMSC and then GA, note that KHMSC runs almost five times faster than GA, but PSO's poor optimality makes it unsuitable for this problem. Based on the experimental results, we can conclude that KHMSC 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 [], 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 network.

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. [31] 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 Krill-Herd (KH) algorithm to solve the service composition problem. Yang et al. [18] 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. zhang et al. [**?**] 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. [32] 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. [33] 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. [34] 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. [35] 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. [**?**] 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. [**?**] 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 CONLUSION

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] X. Hu, X. Li, E. Ngai, V. Leung, and P. Kruchten, "Multidimensional context-aware social network architecture for mobile crowdsensing," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 78–87, 2014.

[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] K. Heimerl, B. Gawalt, K. Chen, T. Parikh, and B. Hartmann, "Communitysourcing: engaging local crowds to perform expert work via physical kiosks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1539–1548.

[7] E. Agapie, J. Teevan, and A. Monroy-Hernández, "Crowdsourcing in the field: A case study using local crowds for event reporting," in *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[8] 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.

[9] 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.

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

[11] 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.

[12] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2254–2262.

[13] Y. Han, T. Luo, D. Li, and H. Wu, "Competition-based participant recruitment for delay-sensitive crowdsourcing applications in d2d networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 2987–2999, 2016.

[14] 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.

[15] 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.

[16] 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.

[17] S. Deng, L. Huang, Y. Li, H. Zhou, Z. Wu, X. Cao, M. Y. Kataev, and L. Li, "Toward risk reduction for mobile service composition," *IEEE transactions on cybernetics*, vol. 46, no. 8, pp. 1807–1816, 2016.

[18] 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.

[19] N. Chadil, A. Russameesawang, and P. Keeratiwintakorn, "Real-time tracking management system using gps, gprs and google earth," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-Con 2008. 5th International Conference on*, vol. 1. IEEE, 2008, pp. 393–396.

[20] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless networks*, vol. 6, no. 4, pp. 307–321, 2000.

[21] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 428–439, 2013.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, no. 5, pp. 533–549, 1986.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.
[33] 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.
[34] 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.
[35] C. Fortuna and M. Mohorcic, "Dynamic composition of services for end-to-end information transport," *IEEE Wireless Communications*, vol. 16, no. 4, 2009.