Availability-aware Mobile Service Composition Over Oppotunistic Networks

XX X, XX X, *Senior Member, IEEE,* XX X, *Fellow, IEEE,*

Abstract—An opportunistic link between two mobile devices or nodes takes place when they are within communication range of each other. Typically, cyber-physical environments comprise a number of mobile devices that are potentially able to establish opportunistic contacts and serve mobile applications in a cost-effective way. Opportunistic mobile service computing is a promising paradigm capable of utilizing the pervasive mobile computational resources around users. Mobile users are thus allowed to exploit nearby mobile services to boost their computing power without investment into their own resource pool. Nevertheless, various challenges, especially its quality-of-service (QoS) and optimal scheduling, are yet to be addressed. Existing studies and related scheduling strategies consider mobile users to be fully stable and available. In this paper, instead, we propose a framework named mobile service opportunistic network (MSON) and an availability-aware QoS model for service composition. We then formulate the problem into an optimization problem and utilize an improved Krill-Herd algorithm to solve it. Finally, we carry out a case study based on some well-known web service workflows and a real-world dataset (the D2D contact traces of MIT Reality dataset and the QoS data of QWS dataset). The comparison implies that our proposed approach outperforms traditional approaches, especially those considering stable and fully available mobille services.

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

1 Introduction

Recent years have witnessed the rapid development of mobile devices (e.g., smartphones, tablets, wearable devices, etc.) and mobile communication. Mobile devices are changing the way people getting the information and the peoples daily lives because they allow you multiple ways of communicating almost anywhere at anytime [1].

The number of mobile devices is still booming and it has already surpassed stationary Internet hosts. Mobile services are also developed and provided at a significant rate, at the same time, the requirements from mobile users are becoming more demanding, i.e., more complicated applications are needed to be run on mobile devices such as virtual reality applications on mobile phones [2] or machine learning applications [3] on mobile phones. However, because of the limited hardware resources of mobile devices (e.g., computational resource, battery life, memory, and storage), these resources-intensive tasks are usually offloaded to mobile computing cloud [4], which result in high data transfer costs (energy cost and communication fee) and high latency.

Opportunistic computing is promising complementary to conventional mobile cloud computing. As illustrated in Fig.1, the basic idea of opportunistic computing is to allow the users to utilize the resources and services that other users share, by exploiting the direct physical contacts between the users, and the resulting potential to exchange data through a direct connection between their devices (e.g. through Wi-Fi or Bluetooth). Resources and services available on mobile devices can be directly shared among users in a elastic and on-demand way without time-consuming and energy-requiring interactions with pre-existing infrastructure, either at the networking level (e.g., cellular networks) or at the computing/service level (e.g., the cloud). Note that, mobile tasks usually require huge computational resources or data transfer (e.g., Tensorflow on mobile, Video editor on mobile, Online video). Nearby mobile service provider are thus more adept, in terms of energy-efficiency, at executing these tasks than the online services or nodes with the help of device to device (D2D) communications such as Bluetooth, Wi-Fi and NFC [5]. D2D communications are featured by extensively-reduced data transfer delays and required energy than traditional cellular network. Thus it provides better user-perceived service quality in terms of reduced waiting time and improved service responsiveness. It is promising to replenish traditional cellular communications in terms of user throughput increase, cellular traffic reduction and network coverage extension.

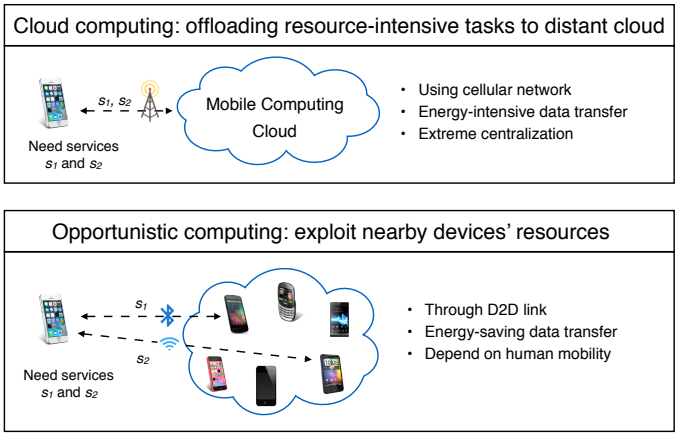

Fig. 1. Opportunistic computing

However, due to the completely different application patterns compared with traditional service computing, service computing in mobile environment faces two inherent challenges.

1) Constant Mobility: Mobile users may change their locations vary frequently in mobile environment, which results in the variation of service availability. Thus, determining how to handle user mobility is a major challenge for providing reliable mobile services in highly dynamic mobile environments.

2) Limited Resource: Mobile devices have limited computing capability compared with other stationary hosts. And mobile service composition schedule must be generated as fast as possible because the availability of mobile service may vary much within a short time. Therefore, how to design an appropriate service selection and composition algorithms which have fast convergence rate and good scalability is another key challenge.

To address the aforementioned challenges and concerns, we propose an availability-aware mobile service composition approach in this paper, where a mobile user in mobile service opportunistic network are allowed to combine and exploit, through D2D communications, nearby devices' resources with time-varying availability (in contrast, traditional approaches mainly consider fully-available mobile services) to boost their computing power and therefore overcome the limitations of their own resources.

The main contributions are:

1) We propose a framework (mobile service opportunistic network, MSON in short) to address the problem of service provisioning in the mobile encounter environment where both service requesters and providers are nonstationary and with time-varying availability. In such environment, mobile user can invoke services exposed by nearby mobile devices through D2D links.

2) For MSON, we propose a availability-aware quality of service (QoS) model for service provisioning to capture users' mobility behavior.

3) Based on MSON and the proposed mobile service QoS model, we formulate the mobile service composition problem to an optimization problem and propose a Krill-Herd based algorithm to solve it.

2 RELATED WORK

2.1 mobile service composition

Mobile service composition refers to the technique of creating composite services with the help of smaller, simpler and easily executable services or components over mobile networks. Recent technological advances in novel mobile device design and development as well as wireless networking materialize a vision where devices all around a user, either embedded as a part of smart spaces, or being carried by other users near by, are enabled to present services probably useful. Users sometimes look for services that are not pre-existent on any device but can be dynamically built by appropriately combining already existing ones. For this purpose, extensive research efforts are carried out in this direction. For example, Deng et al. [7] classify mobile service composition methods into three categories: C2M, M2M, Hybrid. They also discussed the challenge toward mobile service provisioning and mobile service composition in terms of performance, energy and security perspective. Later, Deng et al. [8] proposed a mobile-service-sharing-community model and extend the random way point (RWP) model to capture user mobility. They utilize the meta-heuristic algorithm to decide the optimal compositional plan. Christin Groba et al. [12] present a novel service composition protocol that allocates and invokes service providers opportunistically to minimise the impact of topology changes and to reduce failure. Yang et al. [10] present a comprehensive QoS model specifically for pervasive services. They consider not only mobile wireless network characteristics but also user-perceived factors. They derive a corresponding formula to calculate the QoS criterion. zhang et al. [11] consider a context-aware mobile service selection algorithm based on Genetic Algorithm, They introduce a tree-encoding method to improve the capacity and efficiency of GA. However, for simplicity of the proposed model, they do not consider user mobility. Wang et al. [9] model dependable service composition in wireless mobile ad hoc networks by considering mobility prediction of the service providers. They use a probability-free model and a probabilistic model to characterize the uncertainty of composing. A service that can tolerate a certain level of the mobility of service providers. However, for simplicity of their proposed model, the approach they proposed only apply to the sequential workflows.

2.2 mobile opportunistic network

Opportunistic networking is one of the most promising evolutions of the traditional multi-hop networking. Instead of relying itself on stable end-to-end paths as in the Internet, opportunistic networks do not consider node mobility a problem but as an useful opportunity. Marco et al. [13] 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. [14] 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. Giordano et al. [6] 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. Umair Sadiq et al. [15] propose a algorithm for service composition in opportunistic network. They use Levy walk mobility model and SLAW mobility model to represents some scenarios where each node is equally likely to meet any other node. Multi-hops mechanism is used to provides a direct measure for reachability of nodes between devices when an end-to-end connected path does not exist. Pu et al. [16] 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. Zhan el al. [17] propose a time-sensitive incentive-aware mechanism for mobile opportunistic crowdsensing data collection. They formulate the interaction among data carrier and mobile relay users as a two-user cooperative game and apply a asymmetric Nash bargain solution to obtain the optimal cooperation decision and transfer payment.

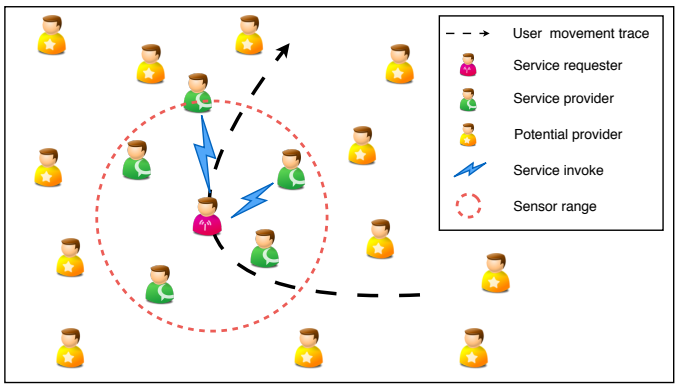

Fig. 2. Mobile service opportunistic network

It can be seen that the limitation of the existing work lie in: 1) the availability of mobile service is varying fastly in real-world scenario due to the mobility of human. Algorithm which doesn't consider service availability as optimization objective or consider service availability is full stable may lead to high invoking fail rate, and recompose after fail result in high response time. 2) the work [15] and [8] consider RWP and Levy model as mobility model, which borrow concepts and methods from random walks and Brownian motion. However, several studies [18] [19] [20] highlighted that individual trajectories are far from random, possessing a high degree of regularity and predictability. And different mobility model applies to different application environment [21], e.g., mobility behavior in a fixed office place is totally different with a crowded subway, thus, it is difficult to find a suitable model which can apply to every scenario. Therefore, an QoS attribute which can be calculated in real-time to indicate service availability is more suitable than these long-range mobility model to guide composition plan to optimal. 3) some works [9] [22] use probabilistic model to characterize the uncertainty (i.e, service invocation fails because of provider out of transmission range). They assume the probability of provider staying within the required distance to the service requester obey a certain rule or distribution. However, in real-world scenario, it is hard to find such a generalized rule or distribution due to different spatial Layout and human traffic (e.g, people flow in a fixed office cubicle is totally different with people flow in a crowded shopping mall). The above limitations could be well avoided by using a service availability analysis instead. We therefore introduce QoS model which consider service availability as an important attribute to capture users mobility. We then feed a Krill-Herd-based algorithm current availability of candidate services, and generate composition plan at run-time.

3 MODELING AVAILABILITY-AWARE MOBILE SERVICE COMPOSITION

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, as done by various existing works discussed in the previous section, it is also assumed that MSON has the following properties:

3.1 Mobile Service Opportunistic Network Framework

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, as done by various existing works discussed in the previous section, it is also assumed that MSON has the following properties:

1) Locality: Rather than stable internet, an MSON bases itself on mobile networks and exploits user mobility. Mobile users in MSON can perceive nearby services and establish self-organized local communication within permitted transmission distance.

2) Mobility: Service requesters and providers keep moving in the mobile network even when they are invoking or provisioning mobile services.

3) Nondeterminism: Mobile services shared in an MSON are transient because the relative distance between any two services keeps changing and could rise above the permitted transmission distance at any time.

Fig. 2 illustrates how the mobile services provision over MSON. In an MSON, a mobile service requester can perceive mobile services exposed by nearby devices through D2D links and launch a request for mobile service composition. A composer process, which is in charge of discovering available mobile services nearby, selecting appropriate concrete services, and composing selected services. All concrete services interact with the composer directly.

Note that, we consider only one-hop D2D links for both service requesters and providers. Because D2D communications which hops are larger than two would incur network overhead [23] while one-hope communications can lower the delay (e.g., no need to transfer a large volume of task contents hop by hop) and ensure framework choose only local relatively reliable service. Besides, some existing researches [24], [25], [26], [27], [28] reveal that users' one-hop neighbors are sufficient enough, compared with multi-hop mechanisms.

We use an simple user case to illustrate the related features of service provision over MSON. Consider a mobile user who is in a crowded subway and his mobile phone has low battery. Now he wants to edit some videos, add some effects and share these video clips to his friends. If he do all these operations on his own mobile phone, his mobile phone will run out of energy because of limited battery. As one option, he can upload original videos to cloud and use cloud service to get all things done, but offloading task into cloud will result in heavy cellular traffic, which means high energy consumption and expensive communication fee. But if he is a participant in MSON and several video processing services is provided by some nearby mobile devices, he can invoke such mobile services through D2D communications. If these services cannot meet his requirement, several services can be composed. Due to users' mobility, the availability of service 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.

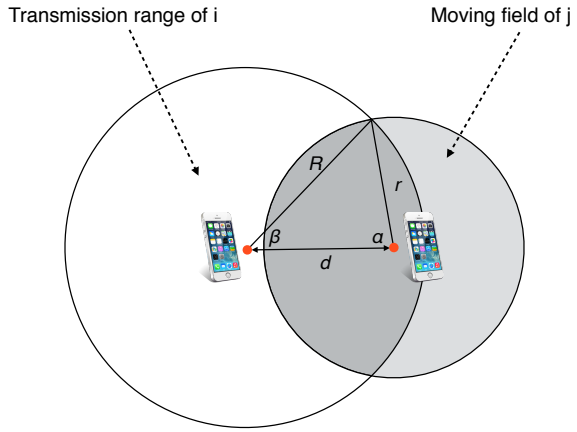

Fig. 3. Mobile service availability

3.2 Mobile Service Availability

In an MSON, the availability of mobile service is varying with time and dynamically decided by users' mobility. As illustrated by an example in Fig. 3, there are two mobile users i and j with identical transmission rage R. User i is a mobile service requester while user j a mobile service provider. Each user moves freely and it is assumed that the moving area is a circle with a radius of r in a certain amount of time, note that this assumption is widely used in related works, e.g., [?][?]. d is the distance between i and j. If user j moves outside the transmission range of its neighbouring user i, then user j is unreachable for user i and consequently the services on user j become unavailable to user i.

Consider that a mobile service running on mobile node j is a candidate service for a task requested by user i, and the availability of candidate service, $Ava(i,j)$ can be calculated as the probability that user j keeps staying inside the transmission range of user i, it serve as the input into the QoS determination model and the scheduling algorithm proposed later:

$$Ava(i,j) = \frac{S_{i \bigcap j}}{S_j} \tag{1}$$

Where $S_{i \bigcap j}$ is the moving area of the user j inside the transmission range of user i, S_j the moving field area of the user j.

The transmission range of a node R is a preset value (e.g., usually 10m for bluetooth and 25m for Wi-Fi). Note that most of wireless transaction protocols have defined the RSSI (Received Signal Strength Indicator), then the distance d between mobile user i and user j can be calculated by signal strength.

The moving radius of a mobile user r is decided by its moving speed v multiplied by the average service time t. t can be statistically calculated as the average service times of recent n trials:

$$t = \frac{\Sigma_{i=1}^{n} t_i}{n} \tag{2}$$

The speed of a mobile user v can be measured and obtained through GPS data or mobile sensors (e.g., Gyro-sensor), then the moving radius can be calculated as:

$$r = t \times v \tag{3}$$

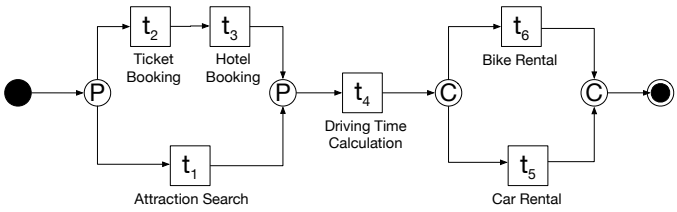

Fig. 4. A sample schedule

Therefore, $S_{i \bigcap j}$ can be formulated as follows:

$$\begin{aligned} S_{i \bigcap j} &= [(\frac{2\alpha}{2\pi} \times \pi r^2) - (\frac{r^2 sin\alpha cos\alpha}{2} \times 2)] \\ &+ [(\frac{2\beta}{2\pi} \times \pi R^2) - (\frac{R^2 sin\beta cos\beta}{2} \times 2)] \\ &= \alpha r^2 + \beta R^2 - (r^2 sin\alpha cos\alpha + R^2 sin\beta cos\beta) \end{aligned} \tag{4}$$

Where

$$\begin{aligned} \alpha &= arccos(\frac{r^2 + d^2 - R^2}{2r \times d}) \\ \beta &= arccos(\frac{R^2 + d^2 - r^2}{2R \times d}) \end{aligned} \tag{5}$$

Finally, S_j can be obtained as:

$$\begin{aligned} S_j &= \pi r^2 \\ &= \pi \times (s \times t)^2 \end{aligned} \tag{6}$$

The availability of mobile service between requester i and provider j thus can be calculated as follow:

$$Ava(i,j) = \frac{\alpha r^2 + \beta R^2 - (r^2 sin\alpha cos\alpha + R^2 sin\beta cos\beta)}{\pi s^2 t^2} \tag{7}$$

3.3 System Model

A Mobile Service Composition Schedule $mscs = (T, E)$ is expressed as a directed acyclic graph (DAG), where $T = \{t_1, t_2, ..., t_n\}$ is a set of tasks and E is a set of directed edges. An edge e_{ij} of the form (t_i, t_j) indicates that a data dependency between t_i and t_j exists and t_i/t_j are the parent/child tasks respectively. A child task is executed after all its parent tasks are completed. A sample Schedule is illustrated in Fig. 1. The most commonly-used composition constructs (a.k.a. composition patterns) for service orchestration include: Sequence, Choice, Parallel and Loop. We denote them by symbols $\rightarrow$, (C), (P) and (L), respectively.

An MSON supports mobile service composition schedule through service providers. If a mobile user is MSON participant, he can decide a set of services of his mobile device to expose to other participants, $P = \{es_1, es_2, ..., es_m\}$. Also he can perceive services exposed by other users, $S = \{s_1^{(i)}, s_2^{(j)}, ..., s_n^{(k)}\}$, $s_n^{(k)}$ means there k candidate services for task t_i. It is assumed that a service provider can provide services in parallel.

The execution order of a schedule can be expressed by assigning an index to each task. The index ranges from 1 to m and the i-th item indicates the order of executing t_i. The relationship between each task and its index can be

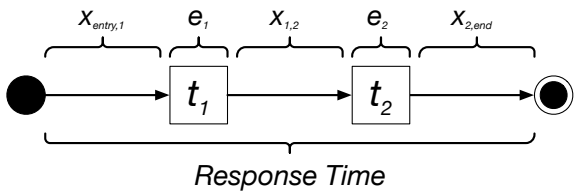

Fig. 5. Responsetime

TABLE 1
Examples of aggregation functions for QoS

Pattern	Availability
sequence	$\prod_{i=1}^{n} Ava(S_i)$
parallel	$\prod_{i=1}^{n} Ava(S_i)$
choice	$\sum_{i=1}^{n} p_i \times Ava(S_i)$
loop	$[Ava(S_i)]^k$

described by a function $l : T \rightarrow N^+$ and encoded as a vector containing a permutation of 1 to m. If i occurs before k in order, it does not necessarily indicate that the execution of t_i is earlier than t_k. The starting time of tasks is decided by the completion time of their preceding tasks.

As shown in Fig. 3, The overall response time of a $mscs$ is made up of execution time of task e_i and data transfer time x_i^k. If task t_i connects t_k through edge $e_{i,k}$ and they are executed by different providers, the transfer time, $x_{i,k}$, is inevitable because inter-provider data and control signal transfer is required. Otherwise, $x_{i,k} = 0$ if both tasks are on the same provider.

$mscs$ tasks executed by different providers usually exhibit varying performance. Moreover, a task executed by the same provider at different time may exhibit fluctuating performance. In this paper, we use the average execution time of recent n trials to represent expected execution time.

3.4 Problem Formulation

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 schedule with the best response time RT while meeting the service availability constraint. The resulting problem can therefore be formulated as:

$$\begin{aligned} Min &: RT \\ s.t &: \tau \leq A \end{aligned} \tag{8}$$

Where A is user's availability on availability. τ is the estimated availability to accomplish the $mscs$, it can be get by aggregation functions shown in Table 1.

The derivation of τ requires some efforts. τ can be calculated as the estimated ending time of the last task in the $mssc$:

$$\tau = d_m \tag{9}$$

where d_i denotes the estimated ending time of task t_i.

d_i can be iteratively calculated as:

$$d_i = e_i + b_i \tag{10}$$

where b_i denotes the estimated starting time of executing t_i and e_i the execution time of t_i itself.

b_i is decided by the estimated ending time of its immediately preceding tasks and the time required for data transfer. Let γ_i denote the estimated time that all earlier tasks executed in the same provider to t_i finished, we have:

$$\gamma_i = max\{d_j \mid l(j) < l(i) \wedge w(i) = w(j)\} \tag{11}$$

where $l(j) < l(i)$ indicates that t_js order index is smaller than that of t_i and $w(i) = w(j)$ means that t_i and t_j are scheduled into the same provider.

Note that the dependency constraint requires that a task be executed only if its all immediately preceding ones successfully terminate and transfer data. We use y_i to denote the estimated earliest time that the described condition holds for t_i.

$$y_i = max\{d_k + x_{k,i} \mid t_k \in {}^*t_i\} \tag{12}$$

where ${}^*t_i$ denotes the immediately preceding tasks of t_i , i.e., those which directly connect t_i through edges in the $mscs$. The earliest possible time to execute t_i, b_i, can therefore be calculated as:

$$b_i = max\{\gamma_i, y_i\} \tag{13}$$

The entry task of a $mscs$ has no preceding tasks and therefore its estimated ending time is obtained as:

$$d_1 = b_1 + e_1 \tag{14}$$

Where b_1 can be obtained as:

$$b_1 = \delta + e_{entry,1} \tag{15}$$

where δ is the time between receiving a service composition request and a schedule plan is generated.

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 [34]:

$$\begin{aligned} Min &: F(X) = c^T X \\ s.t &: Ax \leq b, \\ &\quad x_i \in \mathbb{Z}^n \\ &\quad x_i > 0 \end{aligned} \tag{16}$$

Where $F(x)$ is an objective function, X is a feasible solution, n is a positive integer, b and c are vectors, A is a matrix.

For the service selection and composition problem, the vector $X = \{s_{(1,i)}, s_{(2,j)}..., s_{(n,k)}\}$ 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 smallest $F(x)$. Thus, the problem is equivalent to the integer programming problem which is known to be NP-hard. Then the problem we formulate is NP-hard.

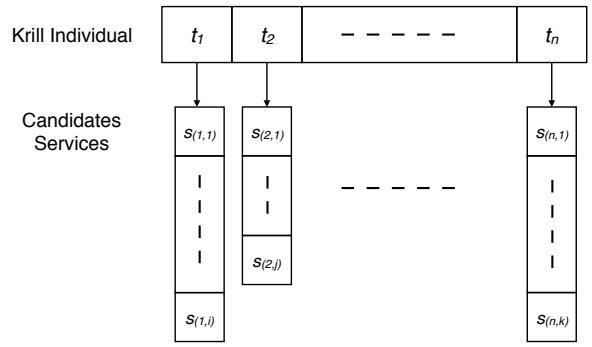

Fig. 6. Krill encoding

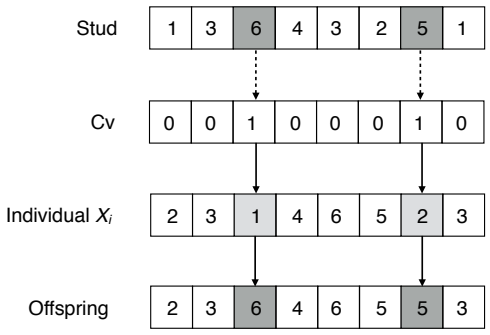

Fig. 7. Crossover operator

4 The KH-based 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 [35]. 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 [36] is new generic stochastic optimization approach for the global optimization problem, it is inspired by predatory behavior and communication behavior of krill. In this section, we will introduce a Krill-Herd based algorithm to solve the problem of mobile service composition over MSON.

4.1 Encoding

In KH, 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 n-th entry is k, it indicates that $s_{(n,k)}$ is the selected concrete service to execute t_n. Fig. 4 illustrates this krill encoding.

4.2 Motion operator

Motion operator is the key component of KH algorithm. As shown in e.q (11), the position of each krill individual is determined by three main factors: 1) motion influenced by other krill; 2) foraging action; 3) physical diffusion.

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{17}$$

where individual $X_i = \{s_{(1,j)}, s_{(2,k)}, ..., s_{(n,l)}\}$ 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:

$$N_i^{new} = N_{max}\alpha_i + \omega_n N_i^{old} \tag{18}$$

where

$$\alpha_i = \alpha^{target} + \alpha^{local} \tag{19}$$

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

$$F_i = V_f\beta_i + \omega_f F_i^{old} \tag{20}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{21}$$

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

$$D_i = D_{max}\delta \tag{22}$$

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 KH 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

$$C_{rate} = Cr + (1 - Cr) \times \frac{U_{best} - U_i}{U_{best} - U_{worst}} \quad (23)$$

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

$$c_i = \begin{cases} 1, & if \ \ rand(0,1) < C_{rate} \\ 0, & else \end{cases} \quad (24)$$

Inspired by SGA [37] (a type of GA which employs the optimal genome for crossover at each generation), we introduce stud selection procedure to improve KH'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.

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}
$C_{rate} \leftarrow$ calcCrossoverRate(X_i, U_{best}, U_{worst})
for $i = 0$ **to** $taskNumber$ **do**
  $r \leftarrow rand(0,1)$
  if $r < C_{rate}$ **then**
    $Cv[i] \leftarrow 1$
  else
    $Cv[i] \leftarrow 0$
  end if
end for
for $i = 0$ **to** $taskNumber$ **do**
  $X_i[i] \leftarrow X_i \wedge (1 - Cv[i]) + Stud \wedge Cv[i]$
end for

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

$$X_{i+1} = X_i + \Delta t \frac{dX_i}{dt} \quad (25)$$

where

$$\Delta t = C_t \sum_{j=1}^{n} (UB_j - LB_j) \quad (26)$$

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 KH algorithm process can be describe in Algorithm 2.

Algorithm 2 KH 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**
    $X_i^{'} \leftarrow motionOperator()$
    $X_i^{''} \leftarrow crossoverOperator(X_i^{'})$
    $U_i^{'} \leftarrow evaluateQoSutility(X_i^{'})$
    $U_i^{''} \leftarrow evaluateQoSutility(X_i^{''})$
    if $U_i^{''} < U_i^{'}$ **then**
      update position by e.q (19) 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 KH-based approach for the mobile service composition algorithm 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.

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 [38] 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 [39] 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

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	s_1, s_2, s_3, s_4
BlueRadios	s_1, s_5
Henrymobile	s_2, s_4
NORTHOLT	s_4, s_5, s_6
Rabbit	s_1, s_4
S4	s_1, s_2
S10	s_6, s_7
S25	s_4, s_5
Tony	s_1, s_4
...	...

TABLE 4
Available candidates

Time	Available service
t1	$s_1^{(3)}$, $s_4^{(3)}$, $s_5^{(2)}$, $s_6^{(1)}$, $s_7^{(1)}$
t2	$s_1^{(3)}$, $s_4^{(3)}$, $s_5^{(2)}$, $s_6^{(2)}$, $s_7^{(1)}$
t2	$s_1^{(4)}$, $s_2^{(2)}$, $s_4^{(4)}$, $s_5^{(2)}$, $s_6^{(2)}$, $s_7^{(1)}$
t4	$s_1^{(4)}$, $s_2^{(1)}$, $s_4^{(3)}$, $s_5^{(2)}$, $s_6^{(2)}$, $s_7^{(1)}$
t5	$s_1^{(4)}$, $s_2^{(2)}$, $s_3^{(1)}$, $s_4^{(4)}$, $s_5^{(3)}$, $s_6^{(2)}$, $s_7^{(1)}$
t6	$s_1^{(2)}$, $s_4^{(3)}$, $s_5^{(3)}$, $s_6^{(2)}$, $s_7^{(1)}$
...	...

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.

5.2 Impact of Parameters

There are six parameters can be adjusted to improve the KH'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 KH algorithm is executed 50 times independently and the average performance was recorded.

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

Fig. 5(a) shows the impact of population size, we observe that with the increase of population size, the average QoS utility of KH 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 KH, 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 KH 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 KH 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 KH fluctuation irregularly and slightly, without loss of generality, we random generate D_{max} from $[0.2 \sim 0.5]$ in this paper.

5.3 Case study

In this section, we present a case study of different service composition plan in mobile environment, to compare traditional mobile service composition approaches with our proposed framework. Fig. 9 shows three mobile service composition plans for case study. Fig. 9(a) is a well known composition plan for booking tickets [40], it has 6 tasks. Fig. 9(b) is a simple workflow with 12 tasks for Tensorflow [3], 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.

As shown by Fig. 7(a), Fig. 8(a) and Fig. 9(a), our proposed method achieves higher success probability (by almost 100for Case I, average 95for Case II, and average

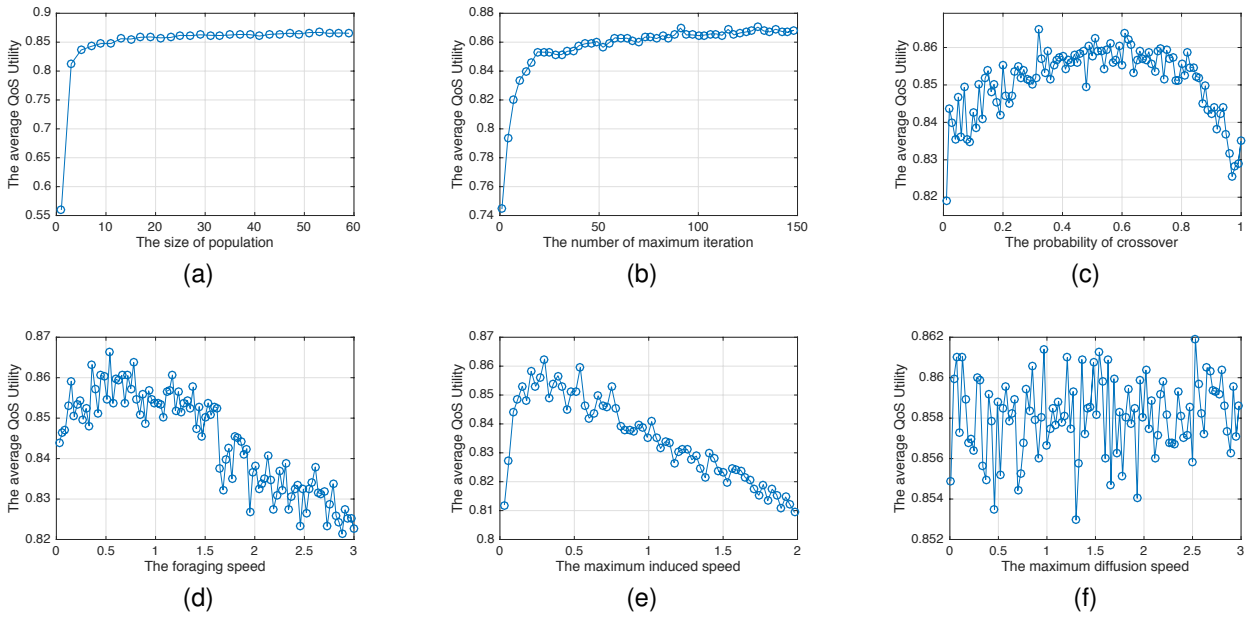

Fig. 8. Impact of parameters.

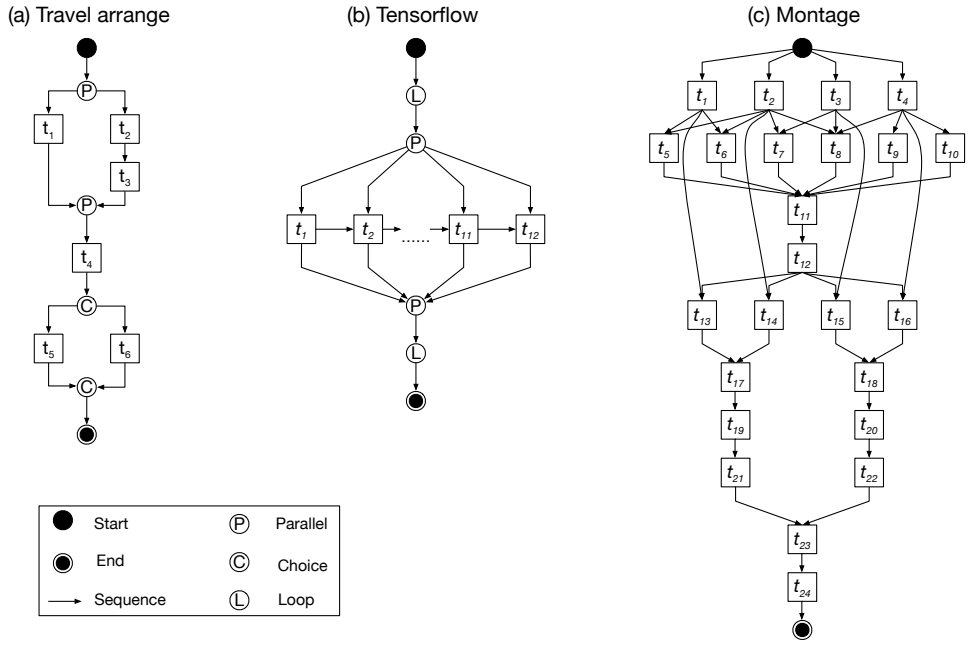

Fig. 9. The composition plan for case study.

90for Case III). From Fig. 7(b), Fig. 8(b) and Fig. 9(b) we can see, the solutions made by traditional mobile service composition approach which doesn't consider mobile service availability some times perform better than our method (especially in Case I). That is because our method consider both response time and service availability as optimization Object and in most of time our method can get the same performance as their approach. But in real mobile environment, low success probability means more time cost. A successful service composition response time is shown in Fig. 7(c), Fig. 8(c) and Fig. 9(c), it indicates that our method perform better in actual mobile environment.

Intuitively, the disadvantage of a non-availability approach lies in that it ignores mobility characteristic of mobile users. It therefore tends to choose the best provider who's QoS data is outstanding but will be not available in few second.

6 CONCLUSIONS AND FURTHER STUDIES

In this paper, we propose a comprehensive framework for optimal mobile service composition 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 an

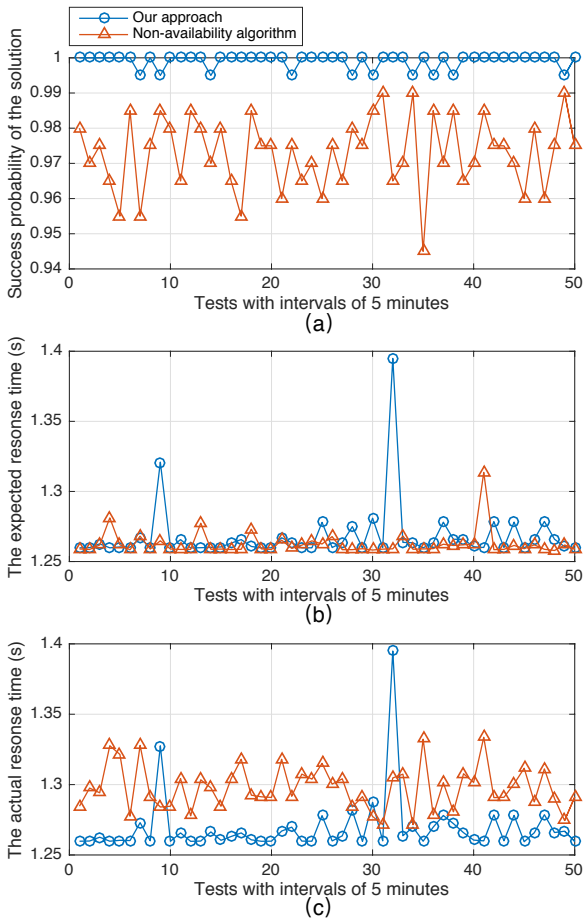

Fig. 10. Case I

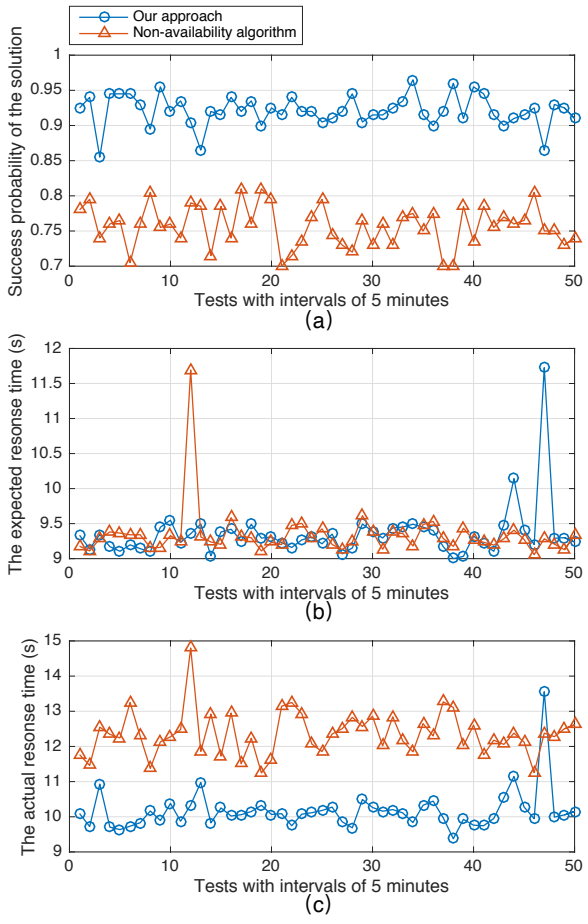

Fig. 11. Case II

availability-aware QoS model for mobile service composition. Then we formulate the composition problem as an optimization problem to maximize the quality of service composition and propose a Krill-Herd-based algorithm to solve it. We also carry out a case study based on real-world opportunistic network and some well-known web service dataset and show that our proposed approach outperforms traditional ones, especially those who consider constant/invariable availability of mobile services, in terms of ?? and ?? .

We consider the following topics for future work as well: 1) Some prediction methods (e.g., hidden Markov model and neural networks) can be used to predict user's future movement instead of our ? prediction method. Such sophisticated prediction methods may help to generate compositional schedules with further improved performance; 2) more QoS metrics (e.g., service price and service reputation) are supposed to be modeled and analyzed; 3) this work considers hard QoS constraints. We intend to consider soft constraints to facilitate analysis and optimization of service-level-agreement (SLA) and introduce corresponding algorithms to generate compositional schedules. In such a context, service completion time is allowed to exceed a threshold value with a bounded given rate.

REFERENCES

[1] M. Satyanarayanan, "Mobile computing: the next decade," in *Proceedings of the 1st ACM workshop on mobile cloud computing & services: social networks and beyond*. ACM, 2010, p. 5.

[2] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

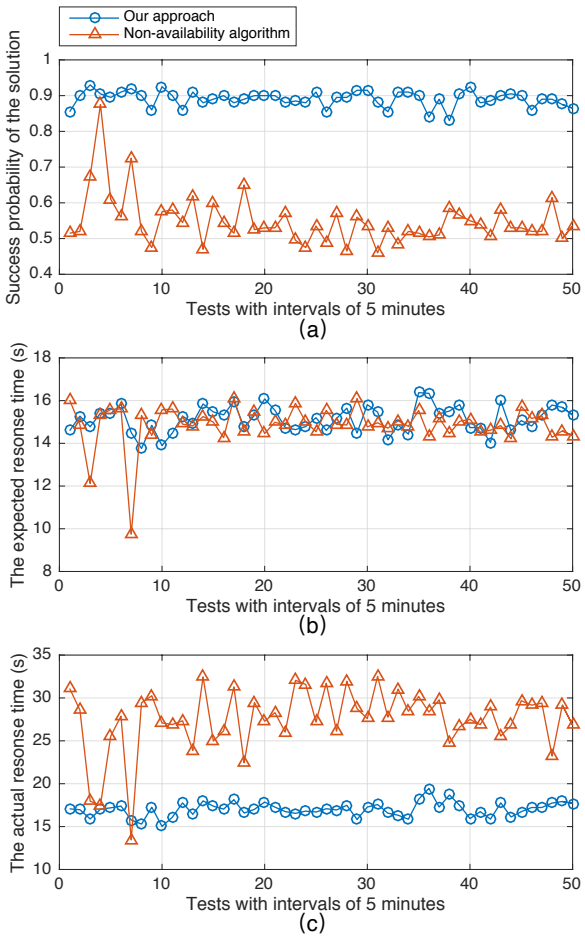

Fig. 12. Case III

[10] 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.

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

[12] C. Groba and S. Clarke, "Opportunistic service composition in dynamic ad hoc environments," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 642–653, 2014.

[13] 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.

[14] 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.

[15] U. Sadiq, M. Kumar, A. Passarella, and M. Conti, "Service composition in opportunistic networks: A load and mobility aware solution," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2308–2322, 2015.

[16] 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.

[17] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Time-sensitive data collection with incentive-aware for mobile opportunistic crowdsensing," *IEEE Transactions on Vehicular Technology*, 2017.

[18] H. Barbosa-Filho, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, "Human mobility: Models and applications," *arXiv preprint arXiv:1710.00004*, 2017.

[19] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on mobile computing*, vol. 2, no. 3, pp. 257–269, 2003.

[20] W. Navidi, T. Camp, and N. Bauer, "Improving the accuracy of random waypoint simulations through steady-state initialization," in *Proceedings of the 15th International Conference on Modeling and Simulation*, 2004, pp. 319–326.

[21] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.