Reliability-aware and Deadline-constrained Mobile Service Composition over Opportunistic Networks

Qinglan Peng, Mengchu Zhou, *Fellow, IEEE,* Yunni Xia, *Senior member, IEEE,* Shuiguang Deng, *Senior member, IEEE,* Xin Luo, *Senior member, IEEE,* Qingsheng Zhu, Wanbo Zheng

***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 and an reliability-aware and deadline-constrained schedule 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 mobile service composition templates and a real-world dataset. The comparison suggest that our proposed approach outperforms traditional approaches, especially those which consider stable and fully available mobile services in their models and algorithms.**

***Index Terms*—Mobile Computing, Mobile Opportunistic Network, Mobile Service Composition, Service Reliability.**

List of abbreviations:

This work is in part supported by NSFC under Grant No. 61472051; Fundamental Research Funds for the Central Universities under project Nos. 106112014CDJZR185503 and CDJZR12180012; Science foundation of Chongqing Nos. cstc2014jcyjA40010 and cstc2014jcyjA90027; Chongqing Social Undertakings and Livelihood Security Science and Technology Innovation Project Special Program No. cstc2016shmszx90002; China Postdoctoral Science Foundation No. 2015M570770; Chongqing Postdoctoral Science special Foundation No. Xm2015078; Universities' Sci-tech Achievements Transformation Project of Chongqing No. KJZH17104.

Qinglan Peng, Yunni Xia, and Qingsheng Zhu are with the school of computers, Chongqing University, Chongqing 400030, China (emails: qlp@cqu.edu.cn, xiayunni@hotmail.com, qszhu@cqu.edu.cn, wbzheng2008@cqu.edu.cn)

X. Luo is with Chinese Academy of Sciences, Chongqing Institute of Green and Intelligent Technology, Chongqing 400714, China (luoxin21@gmail.com)

M. C. Zhou is with Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA (email:zhou@njit.edu).

Yunni Xia is the corresponding author of this work.

C2M	Cloud to Mobile pattern
D2D	Device to Device communications
KH	Krill-Herd algorithm
M2M	Mobile to Mobile pattern
RWP	Random way point mobility model
QoS	Quality of Service
SLA	Service-level-agreement

List of symbols:

α Half of service consumer central angle
β Half of service provider central angle
γ_i The estimated time that all earlier tasks scheduled to the same candidate service to t_i are accomplished
δ The time between the arrival of a service composition request and the generation of its corresponding schedule
τ The estimated complement time of a service composition
$A(s)$ The function to identify availability of mobile service s
b_i The estimated starting time of t_i
C_r The crossover rate of a KH algorithm
C_t The crossover vector of a KH algorithm
d_i The estimated ending time of t_i
D The user-recommended constraint of the completion time of a service composition
D_i The random physical diffusion of a krill individual
$D_{i,k}$ The data transfer time between t_i and t_k
e_i The estimated execution time of t_i
$e_{i,j}$ The edge connecting t_i and t_j
F_i The foraging action of a krill individual
M The max iteration of a KH algorithm
N_i The motion influenced by other krill
P A pool of available mobile services
R Transmission range of a mobile device
S The population size of a KH algorithm
t_i The i-th task of a service composition
$\bar{t}$ The average service time of a concrete service
t_{entry} The dummy beginning task of a service composition
t_{exit} The dummy ending task of a service composition
$w(t_i)$ The function to identify the concrete service that t_i is scheduled into
X The population of krill
X_i The i-th individual in a krill population
y_i The estimated earliest time that all immediately preceding tasks successfully terminate and transfer data

I. 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 people' s 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,

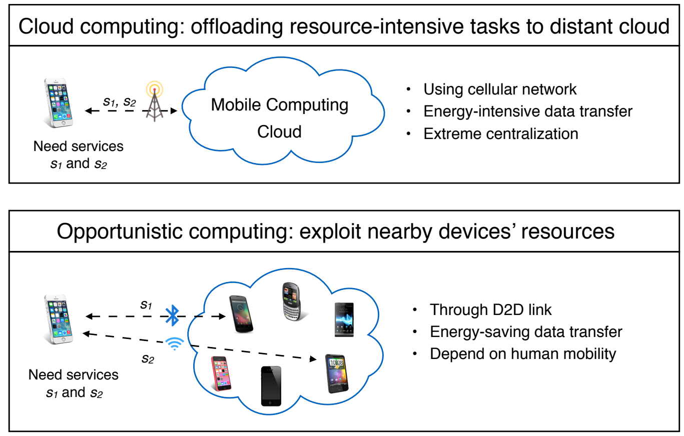

Fig. 1. Opportunistic computing

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 a 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 mobile computing cloud). Note that, mobile tasks usually require huge computational resources or data transfer (e.g., TensorFlow on mobile, Video editor on mobile and 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 widely believed to have potential to replenish traditional cellular communications by providing increased user throughput, reduced cellular traffic, and extened network coverage.

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 very frequently in mobile environment and thus

service availability could be fluctuating and time-varying. It is therefore difficult to guarantee high reliability of service composition when their underlying mobile services are with time-varying availability or even unavailable. Thus, determining how to handle service availability 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 stationary ones. Designing scheduling algorithms with high time-complexity should be avoided. Moreover, such algorithms are supposed to be highly time-efficient because mobile services themselves are with time-varying availability/QoS and only fast ones avoid ineffective scheduling, as well. Nevertheless, the underlying problem for optimal scheduling and composition is well acknowledged to be NP-hard. It is therefore a great challenge to guarantee both low time-complexity and optimality of such algorithms.

To address the aforementioned challenges and concerns in this work, we propose an reliability-aware and deadline-constrained mobile service composition approach, where mobile users in mobile service opportunistic network are allowed to combine and exploit, through D2D communications, nearby devices' resources. Instead of assuming fully available mobile services, we consider time-varying availability of services due to their run-time mobility and develop reliability-aware deadline/reliability estimation models for service compositions. Based on the estimation model, we present a Krill-Herd-based algorithm to decide optimal composition schedules at run-time aiming at maximized reliability with guaranteed deadline. To validate our proposed approach, we carry out a case study based on some well-known mobile service composition templates and a real-world dataset (the D2D contact traces of MIT Reality dataset and the response time data of QWS dataset).

II. RELATED WORK

A. *mobile opportunistic network*

Opportunistic networking is one of the most promising evolutions of the traditional multi-hop one. Instead of relying itself on stable end-to-end paths through Internet, opportunistic networks do not consider node mobility a problem but as an useful opportunity. Extensive research efforts have been paid into this direction. For example, Marco *et al.* [6] give a review of opportunistic networks and regarded it as the first step in people-centric networking, they also discuss challenges to be addressed, especially the mobility modeling and routing-plan decision problems. Turkes *et al.* [7] propose a middleware named Cocoon for mobile opportunistic network. They design a routing protocol on Wi-Fi and Bluetooth connections and and show that their proposed protocol performs well in terms of dissemination rate, delivery latency and energy consumption. Giordano *et al.* [8] consider mobile clouds supported by opportunistic computing, where mobile device can combine and exploit heterogeneous resources from other devices. Pu *et al.* [9] present a QoS-oriented self-organized mobile crowdsourcing framework where prevalent opportunistic user encounters in our daily life are utilized to solve crowdsourcing problem. Zhan *el al.* [10] propose a time-sensitive incentive-aware mechanism for mobile opportunistic crowdsensing. They formulate the interaction among data carrier and mobile relay users as a two-user cooperative game and apply a asymmetric Nash bargain strategy to obtain the optimal cooperation plan.

B. *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. Otherwise, they dynamically built fresh new services by appropriately combining already existing ones. Extensive research efforts are carried out in this direction. For example, Deng *et al.* [11] classify mobile service composition methods into three categories: Cloud to Mobile (C2M), Mobile to Mobile (M2M) and Hybrid. They also discuss related challenges, e.g., performance guarantee, energy efficiency, and security. Later, Deng *et al.* [12] propose a mobile-service-sharing-community model and extend the random way point (RWP) model to model user mobility. They utilize a meta-heuristic algorithm to decide the optimal compositional plan. They consider services shared in community are fully available all the time. Umair Sadiq *et al.* [13] use Levy walk model and SLAW model, where each node is equally likely to meet any other one. Reachability of nodes between devices are guaranteed and supported by their multi-hop connections when their end-to-end connected paths fail. Christin Groba *et al.* [14] present a novel service composition protocol that allocates and invokes service providers opportunistically to minimize the impact of topology changes. However, their protocol only support sequences and parallel service flows. Yang *et al.* [15] present a comprehensive QoS model for pervasive services. They consider not only mobile wireless network characteristics but also user-perceived factors. However, their algorithm considers single service selection rather than multiple service selection for composite service processes. zhang *et al.* [16] consider a bio-inspired and context-aware mobile service selection algorithm, They introduce a tree-encoding method to improve the capacity and efficiency of genetic operations of genetic algorithm. However, for simplicity, they do not consider user mobility. Wang *et al.* [17] employ a probability-free model and a probabilistic model to characterize the uncertainty during service invoking. They assume services are able to tolerate a certain level of the mobility of service providers. However, for

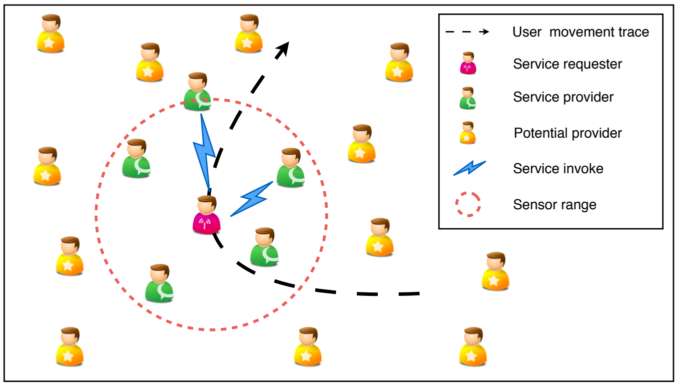

Fig. 2. Mobile services composition over opportunistic network

simplicity, they consider the sequential pattern of service compositions only.

It can be seen that, existing works are limited in several ways: 1) for simplicity, most works consider fully-available services in their QoS determination models and scheduling algorithms. However, real-world mobile services are usually with unreliable and time-varying availability, as suggested by our test data given later. 2) various works consider RWP and Levy model where mobility of services are assumed to follow random walks and Brownian motion patterns. However, recent several studies, e.g., [18]–[20], show 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 pattern of office is different from that of subway. Thus, using a general model to describe such different mobility patterns could be unrealistic. 3) some works, e.g., [17], [22] use probabilistic model to characterize the uncertainty of composite services and consider unavailability of composing concrete services lead to failure of service compositions. They assume the probability that a provider stays within the required distance to its corresponding service requester follows a certain type of given distribution. However, such distribution in real-world is usually unpredictable due to the uncertain spatial layout of mobile devices and the human traffic pattern (e.g., people flow in a fixed office cubicle is totally different from that of a crowded shopping mall).

The above limitations could be well avoided by using a reliability-aware and deadline-constrained service composition mechanism and a Krill-Herd-based algorithm to decide composition schedules, where the reliability of a composition schedule which can be obtained by aggregating availability of its tasks is evaluated and regarded as optimization object to improve successful rate.

III. Modeling Reliability-Aware Mobile Service Composition

A. Mobile Service Composition over Opportunistic Network Framework

Mobile service composition over opportunistic network have three properties:

1) Locality: rather than stable internet, service composition over opportunistic bases itself on mobile networks and exploits user mobility. Mobile users 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 requesting or provisioning mobile services.

3) Dynamicity: the availability of mobile services are time-varying because it is decided by the relative distances between service providers and consumers and such distances are changing.

Fig. 2 illustrates how the mobile services discovery and provision function over an opportunistic network: a mobile service requester perceives mobile services exposed by nearby devices through D2D links and launches a service composition request. A composer process is in charge of discovering available mobile services nearby, selecting appropriate concrete services, and composing selected services. It can be implemented and deployed on mobile devices. All concrete services interact with the composer directly.

Note that, we consider one-hop D2D links for both service requesters and providers due to the fact that D2D communications generally require incur unacceptable network overhead [23]. While one-hope communications generally require low the delay (since it do not need to transfer a large volume of task contents hop by hop) and ensure framework choose local reliable services only. Besides, some existing researches, e.g., [24]–[28], show that a mobile user usually finds sufficient one-hop neighbors to support its applications and thus multi-hop ones are rarely considered.

We use an simple example to explain how services are composed over opportunistic networks. Consider a mobile user situated in a crowded subway whose mobile phone has low battery. The user now wants to edit some videos, add some visual effects, and share these video clips to his friends. Due to low battery of his mobile phone, the user gives up local editing and uploads original videos to a cloud and use cloud services carry out editing jobs. However, offloading tasks into a cloud requires heavy cellular traffic, which requires a lot of energy consumption and expensive communication overhead. Luckily, several video processing services available on nearby mobile devices. The user thus simply decides to choose from and compose invoke such mobile services to get jobs done through D2D communications. Since both the user and candidate services are with high mobility, the composition plan can thus be dynamically, rather than statically, decided based on the availability of services only. It is clear to see that traditional service composition strategies based on static model formulation are insufficient and a novel composition strategy considering dynamic availability is in high need.

B. Service Availability Model

The availability of mobile services is varying with time and dynamically decided by users mobility. As illustrated

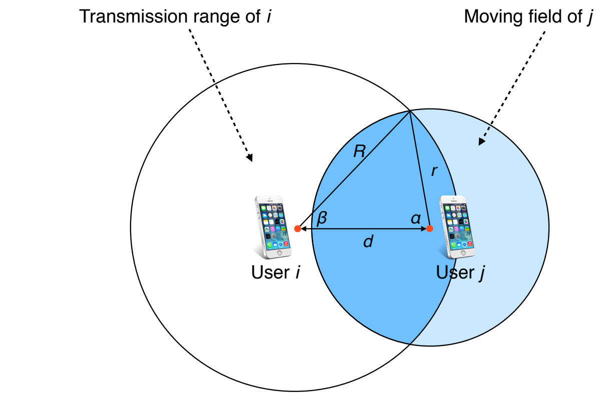

Fig. 3. Mobile service availability

by an example in Fig. 3, mobile users i and j are with identical transmission rage R, which is a preset value (e.g., usually 10m for bluetooth and 25m for Wi-Fi). 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 (note that this assumption is widely used in related works, e.g., [12], [15], [29]). d is the distance between i and j, it can be deduced from RSSI (Received Signal Strength Indicator) easily [30]. If j moves outside the transmission range of its neighbouring i, then j is unreachable for i and consequently the services on j become unavailable to i.

Consider that a mobile service s shared by j is a candidate service for a task requested by i, and its service availability, $A(s)$ can be calculated as the probability that j keeps staying inside the transmission range:

$$A(s) = \frac{S_{i \bigcap j}}{S_j} \tag{1}$$

Where $S_{i \bigcap j}$ is the moving area of j within the transmission range of user i, S_j the moving area of the j. $A(s)$ serves as an input into the reliability-aware composition model and the scheduling algorithm proposed later.

The moving radius of a mobile user r is decided by the product of its moving speed v and the average service time $\bar{t}$. $\bar{t}$ can be statistically calculated as the average service times of recent n trials. 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 the product of $\bar{t}$ and v.

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

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

Where α and β are half of service provider/consumer central angle respectively as shown in Fig. 3, and they

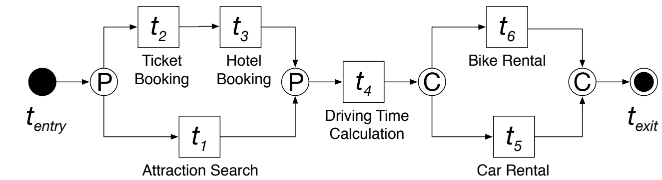

Fig. 4. A sample composite mobile service for arranging travel

can be calculated as follows:

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

Finally, S_j can be obtained as:

$$\begin{aligned} S_j &= \pi r^2 \\ &= \pi \times (v \times \bar{t})^2 \end{aligned} \tag{4}$$

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

$$A(s) = \frac{\alpha r^2 + \beta R^2 - (r^2 sin\alpha cos\alpha + R^2 sin\beta cos\beta)}{\pi v^2 \bar{t}^2} \tag{5}$$

C. Service Composition Model

A mobile service composition can be described by a two-tuple $SC = (T, E)$ [31], where $T = \{t_1, t_2, ..., t_n\}$ is a set of tasks and E is a set of directed edges. An edge $e_{i,j}$ 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. Furthermore, if there is data transmission attached onto $e_{i,j}$, t_j can start only after the data from t_i is received. A dummy tentry/texit task with zero execution time can be added as a sole entry/exit task if the original composition process has multiple entry/exit tasks rather than a single one. D denotes the user-recommended constraint of the completion time of the service composition. Note that this constraint can be either hard or soft one. In this work we consider hard constraint where the actual service composition complement time is bounded by D. A sample composite mobile service for arranging travel plan [32] is illustrated in Fig. 4. We denote the composition patterns (i.e., sequence, choice, parallel and loop) by symbols $\rightarrow$, C, P and L, respectively.

As shown in Fig. 5, a mobile user can perceive services exposed by other users and these available services can be described as service pool $P = \{s_1^{(i)}, s_2^{(j)}, ..., s_m^{(k)}\}$, $s_m^{(k)}$ means there are k candidate services for task t_m. Once a user issues a service composition request, tasks in the composite service is scheduled to the service selected from services pool and executed.

If task t_i connects t_k through edge $e_{i,k}$ and they are executed by different service providers, the transfer time, $D_{i,k}$, is inevitable because inter-provider data and control signal transfer is required. Otherwise, $D_{i,k} = 0$.

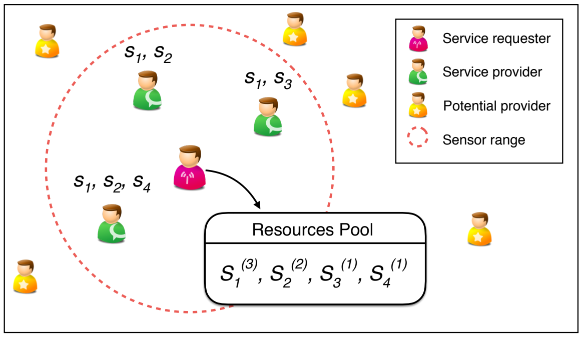

Fig. 5. An example of resource pool

D. Problem Formulation for Optimal Service Composition over Opportunistic Network

As mentioned earlier, a key issue of service composition over opportunistic networks is its promised performance, in terms of deadline of composite service execution, which is decided by the composition schedule carried out at runtime. It should be noted that schedules with high performance but low reliability guarantee should be avoided. The resulting problem can therefore be formulated as:

$$\begin{aligned} Max: \quad & R \\ s.t: \quad & \tau \leq D \end{aligned} \tag{6}$$

Where R is the reliability of a service composition schedule plan, it can be obtained by aggregating all tasks availability thorough a reduction method presented in our earlier work [33]. τ is the estimated complement time of a service composition.

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

$$\tau = d_m \tag{7}$$

where d_i denotes the estimated ending time of task t_i.

d_i can be iteratively calculated as:

$$d_i = e_i + b_i \tag{8}$$

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 scheduled to the same provider to t_i finished, we have:

$$\gamma_i = max\{d_j \mid t_j \in {}^*t_i \wedge w(t_i) = w(t_j)\} \tag{9}$$

where ${}^*t_i$ denotes the immediately preceding tasks of t_i , i.e., those which directly connect t_i through edges in the template. $w(t_i)$ is the function to identify the provider that t_i is scheduled into. $w(t_i) = w(t_i)$ indicates 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.

$$y_i = max\{d_k + D_{k,i} \mid t_k \in {}^*t_i\} \tag{10}$$

The earliest possible time to execute t_i, b_i, can therefore be calculated as:

$$b_i = max\{\gamma_i, y_i\} \tag{11}$$

The first task of a service composition has no preceding tasks and therefore its estimated ending time is obtained as:

$$d_1 = b_1 + e_1 \tag{12}$$

Where b_1 can be obtained as:

$$b_1 = \delta + D_{entry,1} \tag{13}$$

where δ is the time between issue a service composition and a corresponding schedule is generated.

IV. The KH-based algorithm for mobile service composition

The optimization problem discussed in the previous section can be reduced to a knapsack problem. It's widely known that knapsack problem is NP-hard [34], then the optimization problem is also NP-hard. It is therefore extremely time-consuming to yield optimal service composition schedules through traversal-based algorithms. Fortunately, heuristic and meta-heuristic algorithms with polynomial complexity are able to produce approximate or near optimal solutions at the cost of acceptable optimality loss.

Krill-Herd algorithm [35] is novel meta-heuristic generic stochastic optimization algorithm for the global optimization problem, inspired by predatory behavior and communication behavior of krill. Based on our problem descriptions introduced earlier, we present definitions of motion and crossover operations of KH algorithm next and show how resulting composition schedules are generated.

A. Encoding

A service composition schedule is encoded as a krill individual, and the individual with the best position stands for the optimal schedule in terms of its corresponding estimated reliability of a composite service. 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 completion time. Therefore, once the optimal krill individual is found, the best mobile service composition is obtained.

The position vector of each krill individual is represented by an integer array with its length equal to the number of tasks of the service composition request. The i-th entry of the array, in turn, refers to the concrete service selected by task t_i . That is to say, given that the value of the n-th entry is k, $s_{(n,k)}$ is the selected concrete service to execute t_n. Fig. 6 illustrates a simple example of krill encoding.

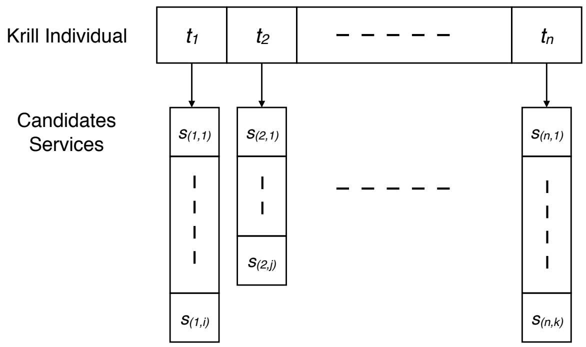

Fig. 6. An example of krill encoding

B. Motion operator

Motion operator is the key component of KH algorithm. As shown in e.q (14), the position of each krill individual is determined by three main factors: 1) motion influenced by other krill; 2) foraging action; 3) physical diffusion.

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{14}$$

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

1) Movement induced by other krill individuals

The motion induced by other krill individuals N_i refers to learning from neighboring individuals with different composition schedules.

$$N_i^{new} = N_{max}\alpha_i + \omega_n N_i^{old} \tag{15}$$

where

$$\alpha_i = \alpha^{target} + \alpha^{local} \tag{16}$$

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

2) Foraging Motion

Similarly, the foraging motion F_i refers to learning from the highest estimated process reliability so far. It has two parts: the current food location and the information about the previous location. The foraging motion of individual X_i, can thus be obtained as follow:

$$F_i = V_f\beta_i + \omega_f F_i^{old} \tag{17}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{18}$$

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

3) Random diffusion

The physical diffusion of individual X_i is considered to be a random process. It includes two components: a maximum diffusion speed and a random directional vector:

$$D_i = D_{max}\theta \tag{19}$$

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

C. Stud selection and crossover operator

Crossover operators aim at enhancing the search capability. Inspired by SGA [36] (a type of genetic algorithm which employs the optimal genome for crossover at each generation), we introduce a stud selection procedure to further improve KH's search capability.

The crossover operator for our proposed problem is designed to be controlled by a dynamic crossover rate C_r:

$$C_r = r + (1-r) \times \frac{R_{best} - R_i}{R_{best} - R_{worst}} \tag{20}$$

Where r is a preset parameter to control crossover rate baseline, R_i the i-th individual's reliability, R_{best} the current highest reliability value as far, and, R_{worst} lowest reliability value so far.

Then a crossover vector $Cv = \{c_1, c_2, ..., c_n\}$ can be generated by C_r:

$$c_i = \begin{cases} 1, & if \quad rand(0,1) < C_r \\ 0, & else \end{cases} \tag{21}$$

Algorithm 1 Crossover operation

Input: Population X; Individual X_i to crossover; The number of tasks $taskNumber$;
Output: A new individual after crossover operation;

```
Sort all krill individuals in population X by its reliability value, get optimal individual Stud, save the best reliability value as R_best and the worst reliability value as R_worst
Calculate crossover rate as C_r by e.q (20)
for i = 1 to taskNumber do
    r ← rand(0,1)
    if r < C_r then
        Cv[i] ← 1
    else
        Cv[i] ← 0
    end if
end for
for i = 1 to taskNumber do
    X_i[i] ← X_i ∧ (1 − Cv[i]) + Stud ∧ Cv[i]
end for
return X_i
```

As shown in 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 reliability value) to mating. As shown in Fig. 7, the characteristics from individual *Stud* are copied to individual X_i according to crossover vector Cv.

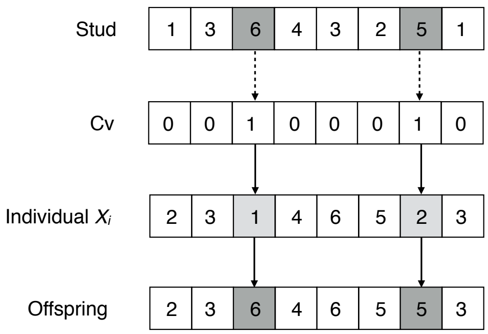

Fig. 7. An example of crossover operator

D. Position update

The offspring generated by crossover operations should be evaluated and updated to current evolutionary sequence. According to the three motion actions proposed earlier, the time-dependent position from time t and Δt can be formulated by the following equation:

$$X_{i+1} = X_i + \Delta t \frac{dX_i}{dt} \tag{22}$$

where

$$\Delta t = C_t \sum_{j=1}^{m} (UB_j - LB_j) \tag{23}$$

where 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 (usually $1/2n$). Based on the above discussions, the improved KH algorithm is given in Algorithm 2.

E. Complexity Analysis

可靠性感知的磷虾群算法（可不可以称为 RAKH 啊?, Reliability Aware Krill Herd Algorithm) 的时间复杂度的计算基于它的：初始化阶段、运动操作、交叉操作和 Rank 操作。我们先分别计算他们的时间复杂度，最后再组合成 RAKH 的时间复杂度。假设用户请求的组合服务 W=（T, E）中有 m 个任务，CEGA 算法的种群数量为 n，此时周围有 k 个用户可以提供服务调用。

按照编码过程生成一个个体所需的时间复杂度为 O(mk)，因为对于每一个任务我们需要从 k 个潜在提供者找出提供这项服务的真正提供者，并随机选取一个。所以整个初始化阶段的时间复杂度为 O(mkn)。运动操作中的三项小操作包括：movement induced by presence of other individuals, foraging motion, and random diffusion, movement induced by presence of other individuals 和 foraging motion 的时间复杂度为 O(n), random diffusion 子操作的时间复杂度为 O(1), 因此运动算子的时间复杂度为 O(n)+O(n)+O(1) = O(n)。

交叉操作的时间复杂度为 O(m), 每一轮操作后都要对当前种群进行排序，而 rank 操作的时间复杂度为 O(nlogn)。因此总的时间复杂度为 O(mk + M(n+m+nlogn))，其中 M 为最大迭代次数。

The computational complexity of the Reliability-aware Krill-Herd Algorithm (RAKH), is calculated based on their initialization, motion, crossover, update and fitness evaluation operations. Initially we calculate the time complexity of each method, then combined to complexity of each method to compute the overall complexity of the proposed RAKH algorithm.

Suppose the mobile user has submitted a composite service request with n tasks, and there are k available service provider within transmission range. S and M is the population size and max iteration of the proposed RAKH algorithm. The time complexity of generating an individual is $O(nk)$, because for each task we need traverse these k potential providers to find the set of actual providers (not all available providers provide a certain kind of service). So, the initial population has the total time complexity of $O(nkS)$. The motion operator consists three sub-operator: 1) motion influenced by other krill; 2) foraging action; 3) physical diffusion, and their time complexity is $O(S)$, $O(S)$, $O(1)$, respectively. Therefor, The time complexity of motion operator is $O(S) + O(S) + O(1) \approx O(S)$. The time complexity for crossover operation is $O(n)$. The time complexity for crossover operation is $O(1)$. Thus, the total time complexity of motion, crossover, update with M generations is $O(MS) + O(Mn) + O(M) \approx O(M(S + n))$.

Algorithm 2 Improved KH algorithm

Input: Number of population size S; Number of max iteration M; Complement time constrain D;
Output: The best solution;

```
Generate initial population as X ← {X_1, X_2, ..., X_S}
for i = 1 to S do
    τ ← estimateCompletionTime(X_i)
    if τ > D then
        X ← remove(X, X_i)
        Generate a valid individual X_v
        X ← add(X, X_v)
    end if
end for
for i = 1 to M do
    for j = 1 to S do
        Calculate movement induced by other krill individuals by e.q (15)
        Calculate foraging motion by e.q (17)
        Calculate random diffusion motion by e.q (19)
        Update position by e.q (22) and save as X'_j
        X''_j ← crossoverOperator(X'_j)
        R'_j ← estimateReliability(X'_j)
        R''_j ← estimateReliability(X''_j)
        τ ← estimateCompletionTime(X''_j)
        if R''_j > R'_j ∧ τ < D then
            accept X''_j as X_{j+1}
        else
            accept X'_j as X_{j+1}
        end if
    end for
end for
Rank all individuals by reliability and return the best one
```

The fitness evaluation method for each individual has the time complexity of $O(n^2)$. The fitness evaluation method for initial population of size S, with M generations has the time complexity of $O(SMn^2)$. The total time complexity of motion, crossover, update and fitness evaluation is $O(M(S+n)) + O(MSn^2) \approx O(MSn^2)$, where $n^2 > S$. Thus the total time complexity of RAKH algorithm is $O(nkS) + O(MSn^2) \approx O(MSn^2)$

V. Case Study and Evaluation

In this section, we present a case study based on some well-known service composition templates and a real-world dataset of measured completion time of D2D contact traces, We apply compare traditional scheduling approaches and our proposed to schedule such service composition templates and show the advantage of our proposed one.

The case study is based on a personal computer with an Intel Core i5 CPU with 2.4 GHz, 4 GB RAM, macOS and Matlab R2015b Edition.

We consider MIT Reality dataset [37] as D2D contact traces of mobile users in a opportunistic network, where user location, Bluetooth devices in proximity, application usage, and phone status (i.e., charging and idle) are collected from 100 mobile users within several months. This dataset can well depicts mobility, time-varying connectivity, and changing reliability of mobile devices and services. We use data from the QWS dataset [38] as the QoS of candidate mobile services. QWS consists of QoS data (response time and throughput) of 4500 real-world Web services observed by 142 users measured every 15 minutes.

Table 1 shows a part of opportunistic contact traces of a mobile user in the MIT Reality dataset, it can be seen that there are 5 nearby devices within D2D transmission distance at time T_1 and thus they are potential service providers. Table 2 shows an example of service providers and the services they exposed to nearby devices. Table 3 shows the Cartesian product of two tables, which identifies the sets of available candidate services at different times, $s_1@p_2$ indicates candidate service s_1 provided by p_2. As shown in Table 3, 5 kinds of services $s_1/s_4/s_5/s_6/s_7$ are available at T_1 and they have 3/3/2/2/1 candidate services, respectively. This table clearly shows that mobile services are with a time-varying set of supporting mobile services. A smart algorithm able to exploit the rum-time dynamicness of time-varying availability of mobile services is thus in high need. The run-time contact time of a mobile service is decided by its corresponding record in the MIT reality dataset. Consider that a composition schedule is applied at T_1 and its estimated completion time is T_3, then the corresponding contact data, which suggests availability status of candidate services, is used decide whether an unsuccessful composition occurs due to unavailability of required services between T_1 and T_3.

We employ representative existing algorithms [12] [13] which consider time-invariant availability and our proposed one to schedule mobile service compositions based on 3 well-known composition templates given in Fig. 8. The templates represent composite services and business processes for travel-arranging, TensorFlow-based data processing, and astronomical image processing applications. As usually done by various existing works [39], [40], data transmission time can be calculated as the throughput (20Mbps as suggested by the MIT Reality dataset) divided by the average bandwidth. The test is based on 50 people’ s opportunistic contract data from MIT Reality dataset. The test includes 50 results of scheduling service compositions based our algorithms and baseline ones in consecutive periods of 5 minutes.

As shown by Fig. 9(a), Fig. 10(a) and Fig. 11(a), our proposed method achieves higher success rate (99.9% vs. 97.3% for Case I in average, 92.1% vs. 75.7% for Case II in average, and 89% vs. 54.9% for Case III in average) compared with baseline algorithms. Fig. 9(b), Fig. 10(b) and Fig. 11(b) shows the comparisons of process completion time where our proposed algorithm achieves 2.2%, 17.7%, and 38.2% time savings in each case. Intuitively, our algorithm outperforms traditional ones because traditional ones consider fully available mobile services and thus they tend to choose services with low response time, and thus fast services with high risks unavailability due to high mobility are selected.

VI. Conclusion

In this paper, we propose a comprehensive framework for optimal mobile service composition in mobile environment. We present a mobile service composition framework for opportunistic network that fully leverages service mobility and introduce an reliability-aware and deadline-constrained mobile service composition model. Then we formulate the developed model as an optimization problem aiming at maximizing the quality of composite service and propose a Krill-Herd-based algorithm to solve it. We also carry out a case study based on a real-world opportunistic network and some well-known mobile service composition templates and shows that our proposed approach outperforms traditional ones, especially those who consider constant/time-invariant availability of mobile services, in terms of success rate and completion time.

We consider the following topics for future work as well: 1) Some prediction methods, e.g., time-series model, hidden Markov models and neural networks can be used to predict user’s future movement and help to achieve further improved performance of composite services; 2) more metrics, e.g., service scalability and service reputation are supposed to be modeled and analyzed; 3) this work consider hard constraints. We intend to consider soft constraints (where completion time of composite services is allowed to exceed a threshold value with a bounded given rate) and introduce corresponding algorithms for optimal run-time scheduling.

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.

TABLE I
User D2D contract traces

Time	Available service providers
T_1	Rabbit, Tony, S10, BlueRadios, NORTHOLT
T_2	Tony, S10, Rabbit, NORTHOLT, BlueRadios
T_3	Rabbit, NORTHOLT, BlueRadios, S10, Tony, Henrymobile, S4
T_4	Tony, NORTHOLT, BlueRadios, S10, Rabbit, S4
T_5	BlueRadios, S4, AliKatz, NORTHOLT, Rabbit, S25, S10
T_6	S25, S10, NORTHOLT, BlueRadios, Rabbit
T_7	AliKatz, S10, NORTHOLT, BlueRadios, S25
T_8	BlueRadios, NORTHOLT, AliKatz, S25, S4, Henrymobile
T_9	AliKatz, BlueRadios, S25, NORTHOLT, Rabbit
...	...

TABLE II
Services exposed by providers

Service Provider	Provider Name	Exposed Services
p_1	AliKatz	s_1, s_2, s_3, s_4
p_2	BlueRadios	s_1, s_5
p_3	Henrymobile	s_2, s_4
p_4	NORTHOLT	s_4, s_5, s_6
p_5	Rabbit	s_1, s_4
p_6	S4	s_1, s_2
p_7	S10	s_6, s_7
p_8	S25	s_4, s_5
p_9	Tony	s_1, s_4
...	...	

TABLE III
Available candidate services

Time	Available candidate services
T_1	$s_1@p_2$, $s_1@p_5$, $s_1@p_9$, $s_4@p_5$, $s_4@p_9$, $s_4@p_4$, $s_5@p_2$, $s_5@p_4$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
T_2	$s_1@p_2$, $s_1@p_5$, $s_1@p_9$, $s_4@p_4$, $s_4@p_5$, $s_4@p_7$, $s_5@p_2$, $s_5@p_4$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
T_3	$s_1@p_2$, $s_1@p_5$, $s_1@p_6$, $s_1@p_9$, $s_2@p_3$, $s_2@p_6$, $s_4@p_3$, $s_4@p_4$, $s_4@p_5$, $s_4@p_9$, $s_5@p_2$, $s_5@p_4$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
T_4	$s_1@p_2$, $s_1@p_5$, $s_1@p_7$, $s_1@p_9$, $s_2@p_6$, $s_4@p_4$, $s_4@p_5$, $s_4@p_9$, $s_5@p_2$, $s_5@p_4$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
T_5	$s_1@p_1$, $s_1@p_2$, $s_1@p_5$, $s_1@p_6$, $s_2@p_1$, $s_2@p_6$, $s_3@p_1$, $s_4@p_1$, $s_4@p_4$, $s_4@p_5$, $s_4@p_8$, $s_5@p_2$, $s_5@p_4$, $s_5@p_8$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
T_6	$s_1@p_2$, $s_1@p_5$, $s_4@p_4$, $s_4@p_5$, $s_4@p_8$, $s_5@p_2$, $s_5@p_4$, $s_5@p_8$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
...	...

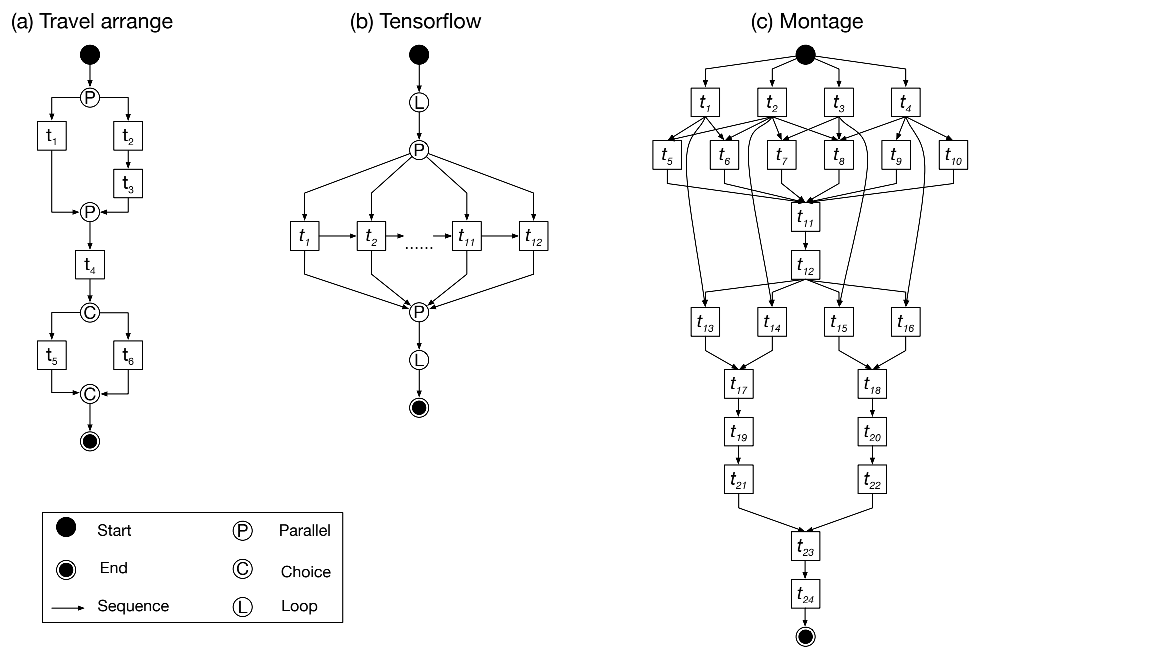

Fig. 8. The mobile service composition templates for the case study

[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 Edufwdocumentsre-*

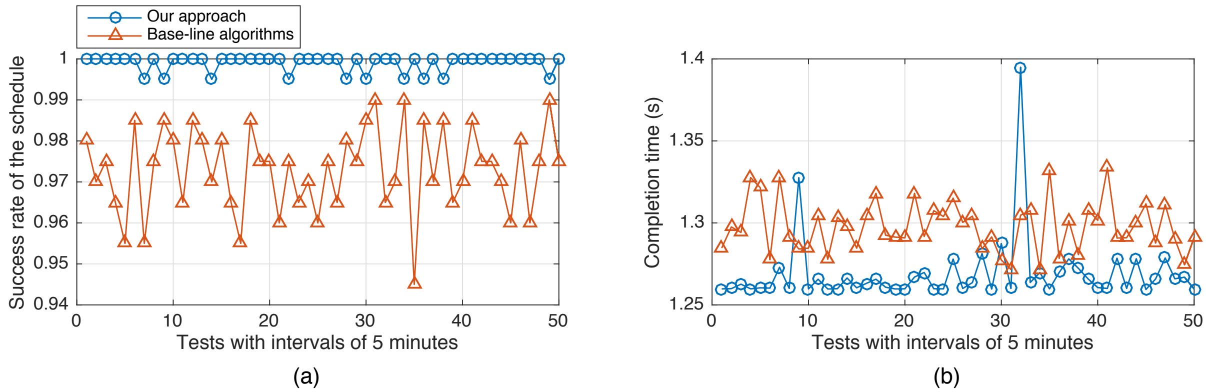

Fig. 9. Case study for travel arrange

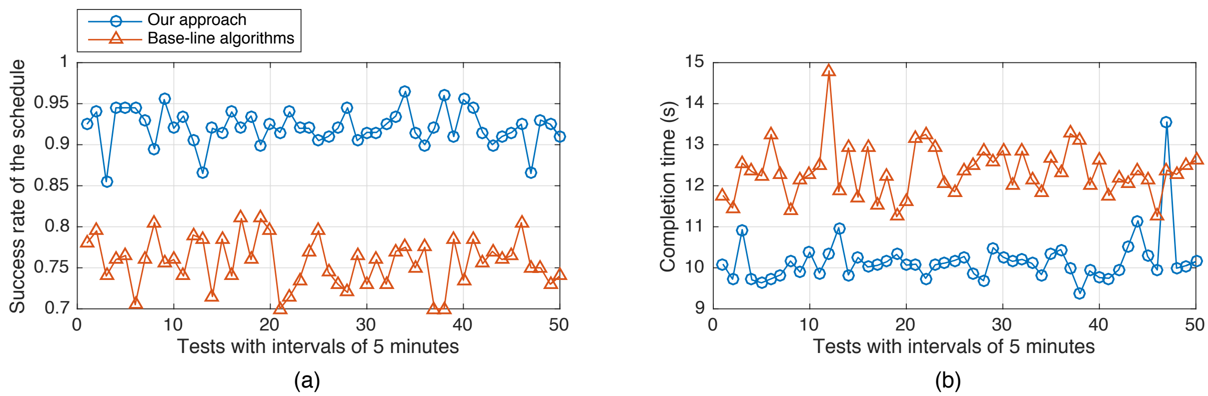

Fig. 10. Case study for tensorflow

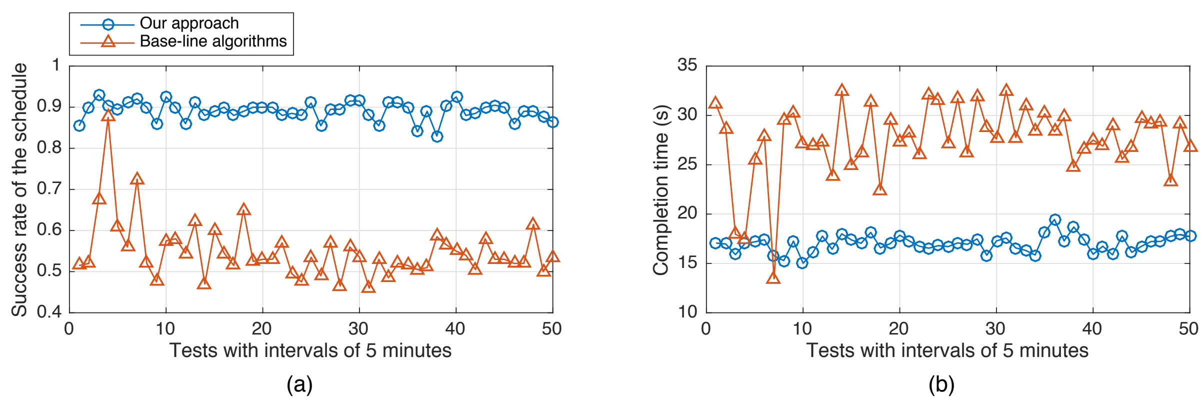

Fig. 11. Case study for montage

ports2007PowerAnalysis Pdf, 2007.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[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] H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke, "Personal paging area design based on mobile's moving behaviors," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2001, pp. 21–30.

[30] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," in *Systems, signals and image processing, 2008. IWSSIP 2008. 15th international conference on*. IEEE, 2008, pp. 303–306.

[31] J. El Hadad, M. Manouvrier, and M. Rukoz, "Tqos: Transactional and qos-aware selection algorithm for automatic web service composition," *IEEE Transactions on Services Computing*, vol. 3, no. 1, pp. 73–85, 2010.

[32] 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.

[33] Y. Xia, Q. Zhu, Y. Huang, and Z. Wang, "A novel reduction approach to analyzing qos of workflow processes," *Concurrency and Computation: Practice and Experience*, vol. 21, no. 2, pp. 205–223, 2009.

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

[35] 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.

[36] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, "Stud krill herd algorithm," *Neurocomputing*, vol. 128, pp. 363–370, 2014.

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

[38] 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.

[39] J. Meena, M. Kumar, and M. Vardhan, "Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint," *IEEE Access*, vol. 4, pp. 5065–5082, 2016.

[40] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds," *Cloud Computing IEEE Transactions on*, vol. 2, no. 2, pp. 222–235, 2014.