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 can be constructed when they are within each other's communication. Typically, cyber-physical environments consist of 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 reliability-aware 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 a mobile service opportunistic network, and a reliability-aware and deadline-constrained scheduling algorithm for service composition. We then formulate the problem into an optimization problem and propose a Krill-Herd-based 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 considering stable and fully available mobile services in their models and algorithms.**

***Note to Practitioners*—Recently, the rapidly development of mobile devices and mobile communication leads to the prosperity of mobile service computing. Services running on mobile devices within limited range are allowed to be composed to coordinate together through wireless communication technologies and perform complex tasks and business processes. Despite its great potential, mobile service compositions remains a challenge due to the fact that mobility of users and devices in mobile environment imposes high unpredictability on the execution of the tasks. A careful investigation into existing methods has found their various limitations, especially consideration of time-invariant availability of mobile services in performance models and scheduling algorithms. In this paper, we introduce a novel reliability-aware and deadline-constrained service composition method for mobile opportunistic networks. Instead of considering time-invariant availability of mobile nodes, our proposed method is capable of estimating service availability at run-time and leveraging a Krill-Herd-based algorithm to yield deadline-constrained and reliability-aware service composition schedules. Extensive case studies based on well-known service composition templates and real-world QoS datasets clearly suggest that our proposed method outperform traditional ones in terms of increased success rates. The proposed method can aid the design and optimization of mobile service compositions in a mobile environment and help practitioners better manage reliability and performance of real-world applications built upon mobile services.**

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

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. Yunni Xia is the corresponding author of this work.

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).

List of abbreviations:

C2M	Cloud to Mobile pattern
DE	Differential Evolution
D2D	Device to Device communications
GA	Genetic Algorithm
KH	Krill-Herd algorithm
M2M	Mobile to Mobile pattern
PSO	Particle Swarm Optimization
RWP	Random way point mobility model
QoS	Quality of Service
SGA	Stud Genetic algorithm
SLA	Service-level-agreement
SLAW	Self-similar Least Action Walk model

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 completion time of a service composition
ω	The number of generations
$A(s)$	The function to identify availability of mobile service s
b_i	The estimated start time of t_i
C_r	The crossover rate of a KH algorithm
C_v	The crossover vector of a KH algorithm

d_i	The estimated end 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
G	The number of max iteration generations of a KH algorithm
m	The number of tasks
N_i	The motion influenced by other krills
P	The resources 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
t_{entry}	The dummy start task of a service composition
t_{exit}	The dummy end 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 krills
X_i	The i-th individual in a krill population
y	The size of initial population

I. Introduction

RECENT YEARS have seen the rocketing development of mobile devices (e.g., smart phones, tablet computers, wearable devices, etc.) and mobile communication. Mobile devices are changing the way people getting the information and their daily lives. These devices allow them to enjoy multiple ways of communicating almost anywhere at anytime [1].

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

Opportunistic computing is a promising complementary to traditional mobile cloud computing. As illustrated in Fig. 1, the core idea of opportunistic computing is sharing. In this paradigm, mobile users are allow to utilize resources and service shared by other users nearby, by exploiting direct physical contacts among users. These available resources and services can be shared directly among users in an elastic and on-demand way without time-consuming and energy-requiring interactions with pre-existing infrastructure, for example, cellular networks in networking level, and mobile computing cloud in computing/service level. Note that, mobile tasks

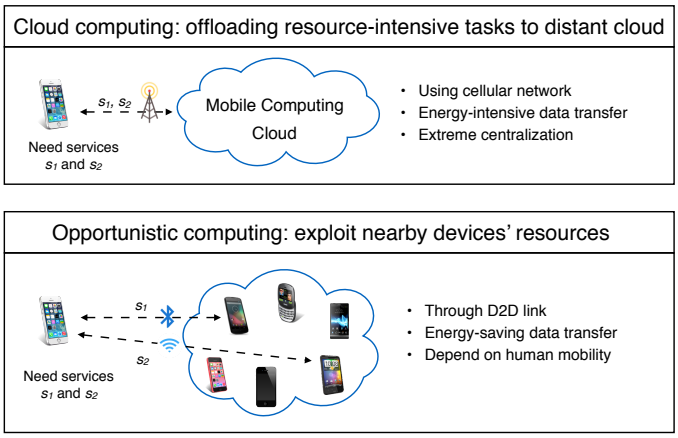


Fig. 1. Opportunistic computing

usually require huge computational resources or data transfer (e.g., TensorFlow Lite, 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 with extensively-reduced data transfer delays and required energy than the 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 and which differs from traditional service computing, service computing in mobile environment faces two inherent challenges.

1) Unguaranteed reliability: Mobile users may change their locations very frequently in a mobile environment and thus service availability tends to be fluctuating and time-varying. It is therefore difficult to guarantee high reliability of service composition when their underlying mobile services are of time-varying availability. However, most existing studies considers fully-available services of constant connectivity in a mobile environment when designing service composition models and algorithms. As a result, such models and algorithms easily lead to low reliability of resulting composite services when applied to real-world mobile applications.

2) Neglect of deadline constraint: Real-world mobile applications may well be time-critical and thus composite services are usually supposed to accomplish required tasks within certain stipulated deadlines. However, most existing studies for mobile service composition consider response-time minimization as the major optimization objective and deadline constraints are rarely taken into account.

To address the above-described challenges and concerns, in this work, we propose a new approach for service selections and compositions in mobile environments. We propose a reliability-aware and deadline-constrained mobile service composition approach, where mobile users in a 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 the QWS dataset).

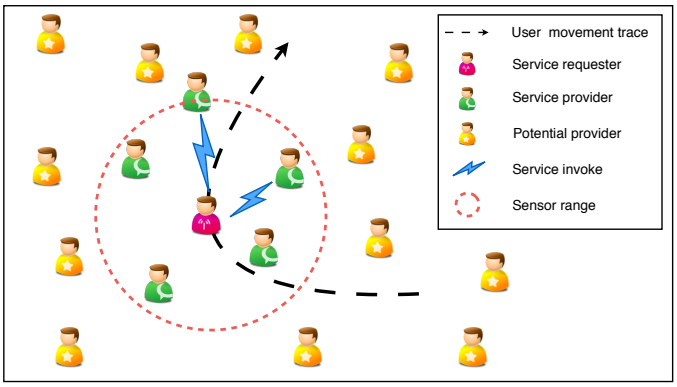


Fig. 2. Mobile services composition over opportunistic network

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 regard node mobility as a problem but useful opportunity. Extensive efforts have been made into this direction. For example, Marco *et al*. [6] give a review of opportunistic networks and regard 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 a 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 a 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 a crowdsourcing problem. Zhan *el al*. [10] propose a incentive-aware and time-sensitive framework for mobile crowdsensing. they formulate the problem of interaction between data carrier and relay users into a two-player game, and utilize an asymmetric Nash bargain strategy to generate optimal schedule.

B. Mobile service composition

Mobile service composition technique allows users to issue composite services with the help of fine-grained services over mobile networks. Recent technological advances made in the hardware and software of mobile devices, especially wireless networking, materialize a vision where devices all around a user, either carried by nearby users, or embedded as a part of smart spaces, are enabled to present services probably useful. Users sometimes demand novel services which are not pre-existent on any devices. Otherwise, they dynamically build new services by appropriately combining already existing ones. Extensive studies 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 schedule. They consider services shared in a community are fully available all the time. Sadiq *et al*. [13] use a Levy walk model and Self-similar Least Action Walk (SLAW) model to generate user traces, where each node is equally likely to meet any other one and the connectivity between devices is implemented using multi-hop paths. Christin Groba *et al*. [14] present a schedule protocol for service composition. Their protocol allocates service providers opportunistically to service consumers to minimize the impact of change on topology. However, their protocol only support a sequences and parallel service flows. Yang *et al*. [15] propose a QoS model for mobile service selection. 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 a 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 that services are able to tolerate a certain level of the mobility of service providers. However, for simplicity, they consider the sequential pattern of service compositions only.

It can be seen that, existing studies are limited in several ways: 1) for simplicity, most of them consider fully-available services in their QoS determination models and scheduling algorithms. However, real-world mobile services are usually of unreliable and time-varying availability, as suggested by our test data given later. 2) Various studies consider RWP and Levy models where mobile devices 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. Different mobility model must be applied 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 is unrealistic. 3) Some, e.g., [17], [22] use a probabilistic model to characterize the uncertainty of composite services and consider that unavailability of composing concrete services leads 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 adopted as an optimization objective to improve success rate of mobile service composition.

III. Modeling Reliability-Aware Mobile Service Composition

A. Mobile Service Composition over Opportunistic Network Framework

Mobile service composition over an opportunistic network has three distinctive properties:

1) Locality: In a mobile opportunistic network, mobile users are allowed to perceive nearby services and establish self-organized local communications within permitted transmission distance. Thus, locality of services can well be exploited and utilized.

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 is time-varying because it is decided by the relative distance between service providers and consumers and such distance is changing.

Fig. 2 illustrates how mobile service 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 can be implemented and deployed on mobile devices, and it allow mobile devices to discover available mobile services nearby, select appropriate concrete services, and compose selected services into a service composition. All concrete services interact with the composer directly.

We consider only one-hop D2D links for both service requesters and providers since multi-hop D2D communications generally incur unacceptable network overhead [23]. Some existing researches, e.g., [24]–[28], as well, show that mobile users can usually find sufficient one-hop neighbors to support their tasks and thus multi-hop ones are rarely considered.

We use a simple example to explain how services are composed over opportunistic networks. Consider Tom situated

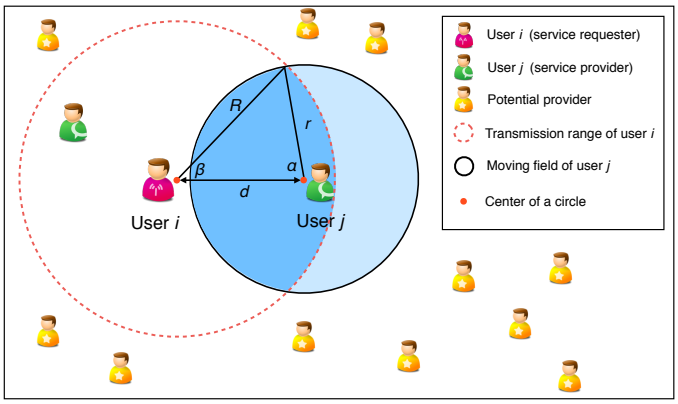


Fig. 3. Mobile service availability

in a crowded subway whose mobile phone has low battery. He now wants to edit some videos, add some visual effects, and share these video clips to his friends. Because of low battery of his mobile phone, he gives up local editing and uploads original videos to a cloud and uses cloud services to carry out editing jobs. However, uploading tasks into a cloud requires heavy cellular traffic, which requires much energy consumption and expensive communication overhead. Luckily, several video processing services available on nearby mobile devices. Tom thus decides to choose from and compose and invoke such mobile services to get jobs done through D2D communications. Since both the user and candidate services are of high mobility, the composition plan can thus be dynamically decided based on the availability of services. It is clear to see that traditional service composition strategies based on a static model formulation are insufficient and a novel composition strategy considering dynamic availability is required.

B. Service Availability Model

The availability of mobile services is varying with time and dynamically decided by user mobility. As illustrated by an example in Fig. 3, mobile users i and j are with in 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 their moving area is a circle with a radius r (note that this assumption is widely used in related work, e.g., [12], [15], [29]). d is the distance between i and j, and it can be deduced from RSSI (Received Signal Strength Indicator) easily [30]. If j moves out of the transmission range of i, then j is unreachable for i and consequently the services of j become unavailable to i. Consider that a mobile user i issues a composite service which contains task t. A mobile service s shared by user j is a one of the candidate service for task t, 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 j. $A(s)$ serves

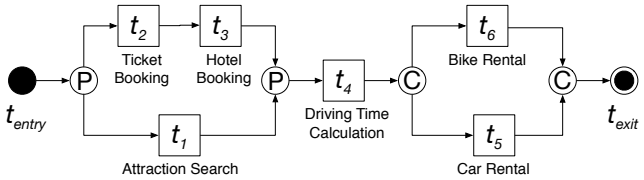


Fig. 4. A sample composite mobile service for arranging travel

as an input into a reliability-aware composition model and scheduling algorithm to be proposed later.

The radius of user moving area r is decided by the product of its moving speed v and the average service time $\bar{t}$. The speed of a mobile user v can be measured and obtained through GPS data or mobile sensors (e.g., Gyro-sensor). The average service time $\bar{t}$ can be statistically calculated form the recent n trials. 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 angles, respectively as shown in Fig. 3, and they 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 can thus be estimated as:

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

C. Service Composition Model

A mobile service composition can be described by a two-tuple $SC = (T, E)$, where $T = \{t_1, t_2, ..., t_m\}$ 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 only 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 completion time is bounded by D. A sample composite mobile service for arranging a travel plan [31] is illustrated in Fig. 4. We denote the composition patterns (i.e., sequence, choice, parallel and loop) by symbols $\rightarrow$, Ⓒ, Ⓟ and Ⓛ, respectively.

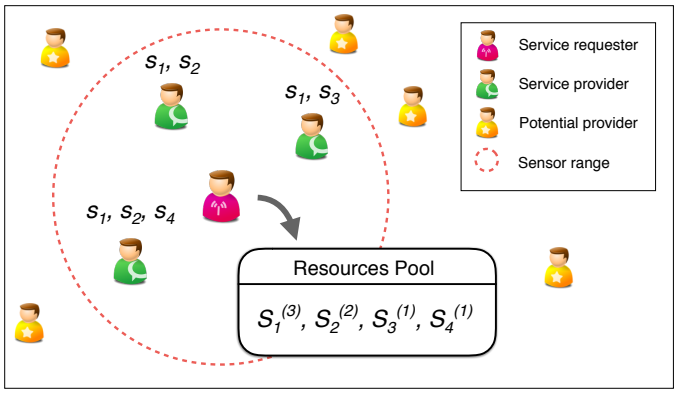


Fig. 5. An example of resource pool

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 are 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, $D_{i,k} = 0$.

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 run-time. 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, and can be obtained by aggregating all services availability thorough a reduction method presented in our earlier work [32]. τ is the estimated time of a service composition.

The derivation of τ requires some efforts. τ can be calculated as the estimated end time of the last task t_{exit} in a service composition template:

$$\tau = d_{exit} \tag{7}$$

where d_{exit} denotes the estimated end time of task t_{exit}. We use d_i to denote the estimated end time of task t_i, it can be iteratively calculated as:

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

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

b_i is decided by the estimated end time of its immediately preceding tasks and the time required for data transfer. Let γ_i denote the estimated time when all earlier tasks scheduled to the same provider to t_i are 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 into which t_i is scheduled. $w(t_i) = w(t_j)$ 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 immediately preceding ones successfully terminate and transfer data. We use y_i to denote the estimated earliest time when 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 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 task 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 issuing a service composition and generating a corresponding schedule.

IV. KH-BASED ALGORITHM FOR MOBILE SERVICE COMPOSITION

The optimization problem proposed in the previous section can be reduced to a knapsack problem. It is widely known that the latter is NP-hard [33]. Hence the proposed 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.

Krill-Herd algorithm [34], called KH for short, is a light-wight meta-heuristic generic stochastic optimization algorithm for global optimization problems. Inspired by predatory behaviors, communication behaviors and random diffusion behaviors of krills. These behaviors are in parallel and thus resulting algorithms can be parallel executed for high efficiency. These distinctive features make KH algorithms high suitable for solving the problem described in this paper.

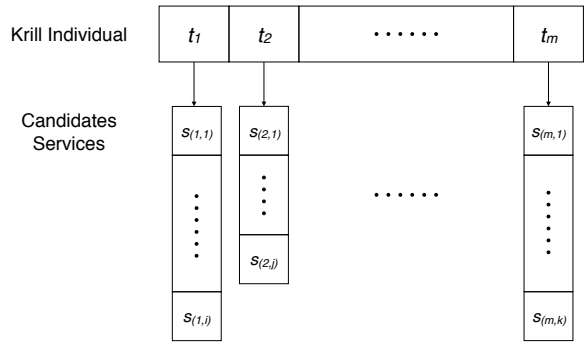


Fig. 6. An example of krill encoding

Based on our problem descriptions introduced earlier, we present definitions of motion and crossover operations of KH 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 so-far-obtained best schedule in terms of its corresponding estimated reliability of a composite service. The target of KH is to find an individual with the best position, which corresponds to a composition schedule with the highest estimated reliability. Therefore, once the optimal individual is found, the best mobile service composition is obtained.

The position vector of each individual can be represented by an integer array, whose length equal to the number of tasks in a service composition request. The i-th entry of the array, in turn, refers to the concrete service selected by task t_i . In other words, 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.

B. Motion operator

A motion operator is a key component of KH. The change in position of each krill individual has three primary contributors: 1) motion influenced by other krills; 2) foraging motion; 3) random physical diffusion. Therefore, the Lagrangian model of a KH's motion operator can be described as follows:

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

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

1) Movement influenced by other krills

The motion influenced by other individuals 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

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

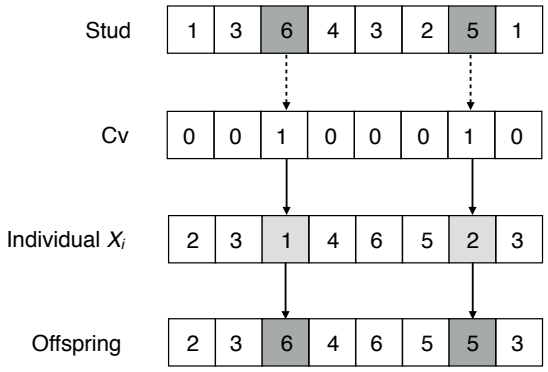


Fig. 7. An example of crossover operator

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

2) Foraging Motion

Similarly, the movement caused by foraging F_i refers to learning from the highest estimated process reliability so far. It has two parts: the information about current food location and previous food 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 motion, and F_i^{old} the last foraging motion, and β_i the direction of the foraging motion.

3) Random physical diffusion

The physical diffusion of individual X_i can be considered as a random process. It includes two components: a random direction vector and a maximum physical diffusion speed.

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

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

C. Crossover operator

Crossover operators aim at enhancing the search capability. Inspired by SGA (Stud Genetic algorithm) [35] that 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:

$$C_r = r + (1-r) \times \frac{R_h - R_i}{R_h - R_l} \tag{20}$$

where r is a preset parameter to control crossover rate baseline, R_i the i-th individual's reliability, R_h the current highest reliability value, and R_l lowest reliability value obtained 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 m;

Output: A new individual after a crossover operation;

```
rank all krill individuals in population X by its reliability value, get the best individual Stud, save the highest reliability value as R_h and the lowest reliability value as R_l
calculate crossover rate as C_r by (20)
for i = 1 to m 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 m 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 the highest reliability value) to mate. As shown in Fig. 7, the characteristics from individual $Stud$ are copied to individual X_i according to crossover vector Cv.

D. Position update

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

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

where

$$\Delta t = C_T \sum_{j=1}^{m} (U_j - L_j) \tag{23}$$

where C_T is a constant to scale the search space, U_j the upper bound of candidate services for the j-th task and L_j the lower bound.

E. Population Initialization

The search space of solutions usually grows exponentially with the scale of a composite service. To accelerate the search speed and convergence of the KH algorithm, the initial population is defined and generated in such a way that each individual of the population complies with the structural constraint of the

service composition, and its position vector is generated by randomly choosing concrete services from the resource pool. The pseudocode of generating initial population is given in Algorithm 2. This procedure starts with constructing a resource pool through summarizing services exposed by service providers nearby. Then we choose concrete services randomly from the resource pool for each individual.

Algorithm 2 Initialize population

Input: Population size y; The set of available service provider P; Composite service SC requested;
Output: Population X;

initialize Resource Pool P with empty
initialize Population X with empty
for each service provider p in P **do**
  for each task t in SC **do**
    if provider p provide concrete service s for t **then**
      add service s to Resource Pool $P[t]$
    end if
  end for
end for
for $i = 1$ **to** y **do**
  initialize individual $X_i \leftarrow \{0, 0, ..., 0\}$
  for each task t in SC **do**
    $X_i[j] \leftarrow$ RandomChoose($P[t]$)
  end for
  add individual X_i to population X
end for
return population X

F. Fitness evaluation and selection

The fitness of an individual solution is decided by its estimated reliability on condition that the deadline constraint is met. The selection strategy is therefore based on the fitness estimation and implemented through a tournament-based method [36]. To be specific, we employ a constraint-handling strategy described below to select individuals for new generations: 1) if the τ values of two solutions meet the user-defined deadline, then discard the one with lower estimated reliability, 2) if the τ value of only one solution meets the deadline, then discard the unsatisfied one; and 3) if the τ values of both solutions exceed the deadline, then discard the one with lower estimated reliability. Based on the above discussions, the KH-based composition algorithm is given in Algorithm 3.

G. Complexity Analysis

The overall computational complexity of our proposed framework can be analyzed by examining its initialization, fitness evaluation, motion, crossover and update operations. Suppose there are k available service providers within transmission range, the time complexity of resource pool initialization is $O(m \times k)$. The time complexity of initializing an individual is $O(m)$, and thus population initialization requires $O(m \times y)$ where y is the size of initial population. Therefore, the time complexity of initialization operation is $O(mk + my)$. The fitness evaluation for each individual has the time complexity

Algorithm 3 KH-based composition algorithm

Input: Number of population size y; Number of max iteration generations ω; The set of available providers P; Composite service SC requested;
Output: The best solution;

$X \leftarrow InitializePopulation(y, P, SC)$
for $i = 1$ **to** ω **do**
  evaluate the fitness of all individuals
  for each individual X_j in population X **do**
    calculate movement induced by other krill individuals by (15)
    calculate foraging motion by (17)
    calculate random diffusion motion by (19)
    update position by (22) and save as $offspring_1$
    $offspring_2 \leftarrow$ CrossoverOperator($offspring_1$)
    $offspring \leftarrow$ Tournament($offspring_1, offspring_2$)
    mark $offspring$ as X_j
  end for
end for
return the best individual

of $O(m^2)$ and thus fitness evaluation for initial population of size y with ω generations has the time complexity of $O(y\omega m^2)$. The time complexity for motion, crossover, and update operations are $O(y)$, $O(m)$, and $O(1)$, respectively. Consequently, the total time complexity of motion, crossover, update with ω generations is $O(\omega y) + O(\omega m) + O(\omega)$. Finally, the total time complexity of initialization, fitness evaluation, motion, crossover and update is thus $O(mk + my) + O(y\omega m^2) + O(\omega y) + O(\omega m) + O(\omega) = O(y\omega m^2)$.

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 and compare traditional scheduling approaches and our proposed to schedule such service composition templates.

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

We consider MIT Reality dataset [37] as D2D contact traces of mobile users in an opportunistic network, where user location, application usage, Bluetooth devices in proximity, and hardware status (i.e., charging or 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 I 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 II shows an example of service providers and the services they expose to nearby devices. Table III shows the Cartesian product of

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	$s_1@p_2$, $s_1@p_5$, $s_4@p_4$, $s_4@p_5$, $s_4@p_8$, $s_5@p_2$, $s_5@p_4$, $s_5@p_8$, $s_6@p_4$, $s_6@p_7$, $s_7@p_7$
...	...

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 III, five 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 time-varying availability of mobile services is thus needed. 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 to decide whether an unsuccessful composition occurs as caused by the unavailability of required services between T_1 and T_3.

For the comparison purpose, we also apply a representative algorithm proposed in [12] [13], which consider time-invariant availability, to schedule four composition templates given in Fig. 8. Note that we consider these two as the baseline algorithms because: 1) their physical model formulation is identical to ours; 2) their test results suggest that they outperform all other earlier methods; and 3) their proposed algorithms are different in details but actually equally effective.

The four composition templates include three well-known templates and a random generated template. The three well-known templates represent composite services and business processes for travel-arranging, TensorFlow-based data processing, and astronomical image processing applications. As usually done in various existing studies [39], [40], data transmission time can be calculated as the throughput divided by the average bandwidth (20Mbps as suggested by the MIT Reality dataset). The test is based on 50-people opportunistic contract data from the MIT Reality dataset. The test includes 50 results of scheduling service compositions with our algorithms and its peers in consecutive periods of 5 minutes.

As shown in Fig. 9(a), Fig. 10(a), Fig. 11(a) and Fig. 12(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, and 91.8% vs. 75.6% for Case IV in average) compared with its peers. Fig. 9(b), Fig. 10(b), Fig. 11(b) and Fig. 12(b) shows the comparisons of process completion time where our proposed algorithm achieves 2.2%, 17.7%, 38.2%, and 18.8% time savings in each case over its peer. Intuitively, our algorithm outperforms traditional ones because traditional ones consider fully available mobile services and therefore ignore the reliability of a composite service, thus they tend to choose services with low response time. As a result, the service composition constructed by these low response time services

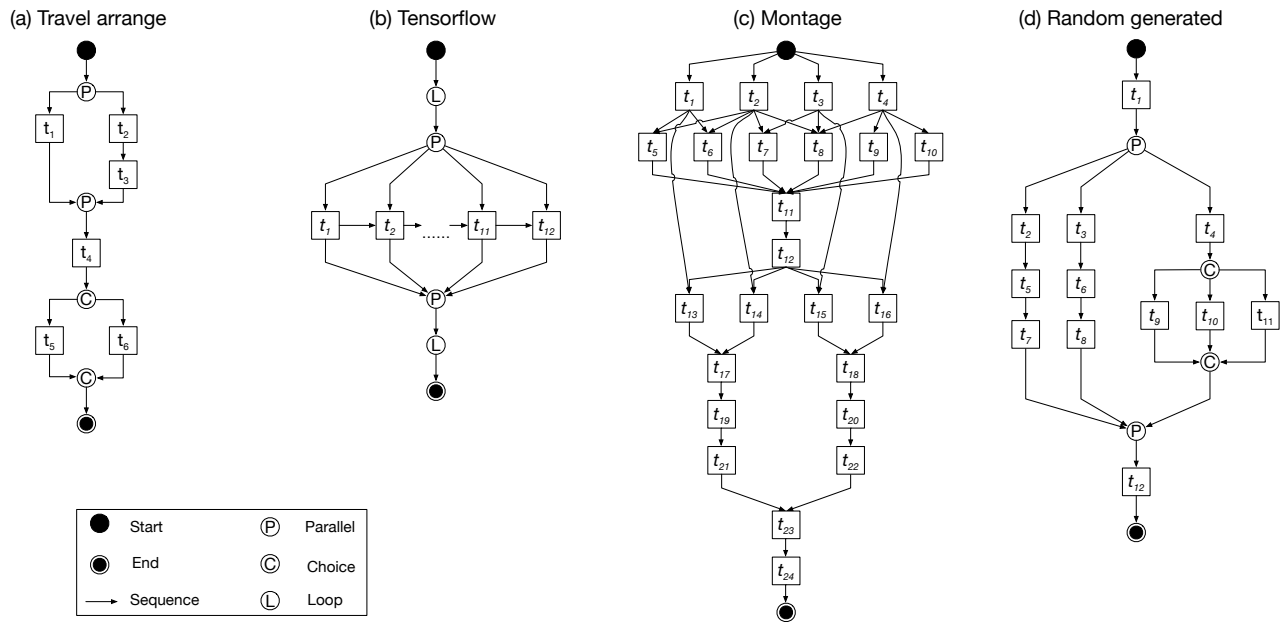


Fig. 8. The mobile service composition templates for the case study

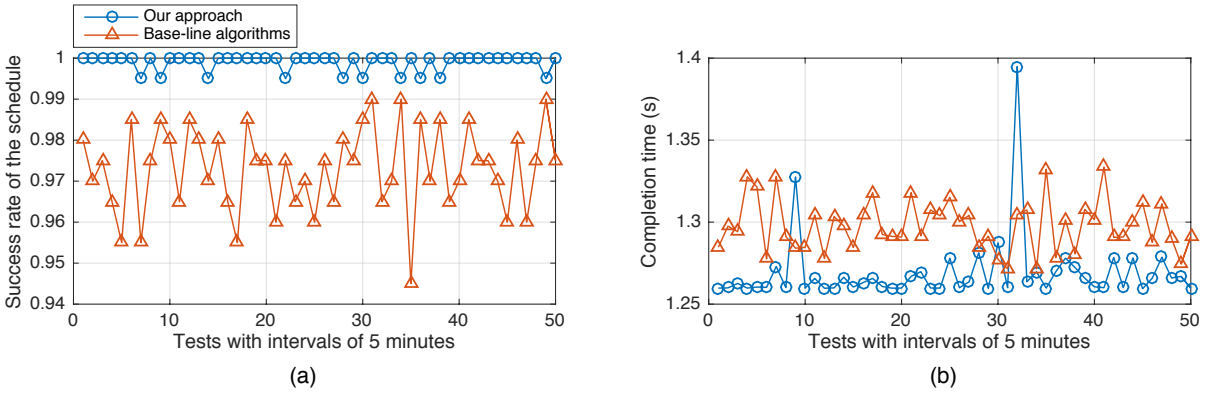


Fig. 9. Case study for travel arrange

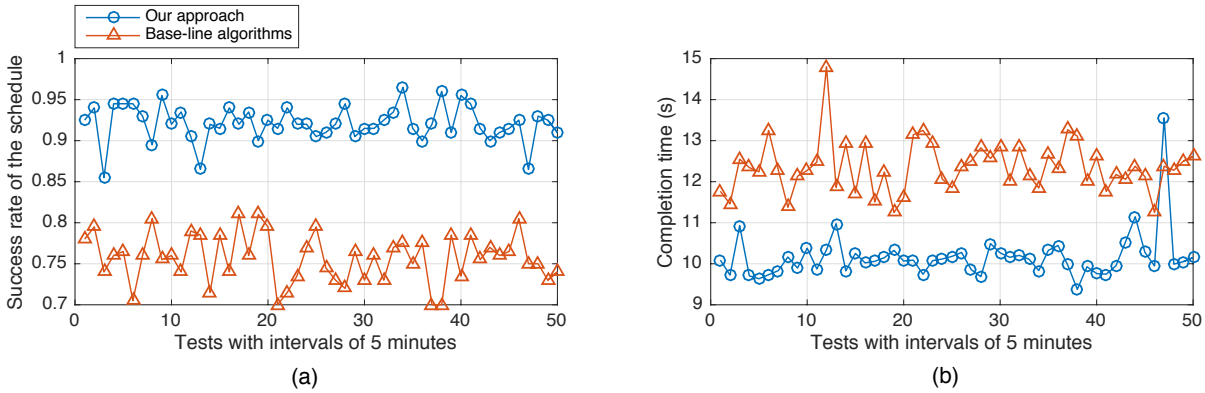


Fig. 10. Case study for tensorflow

may have a poor reliability and would eventually lead to a failure.

VI. Conclusion

This paper presents a comprehensive framework for optimal mobile service composition in a mobile environment. It

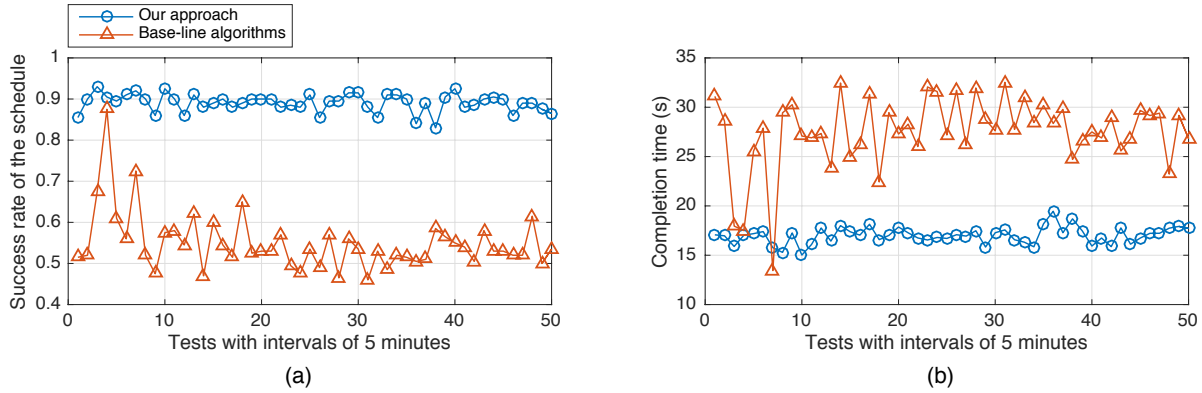


Fig. 11. Case study for montage

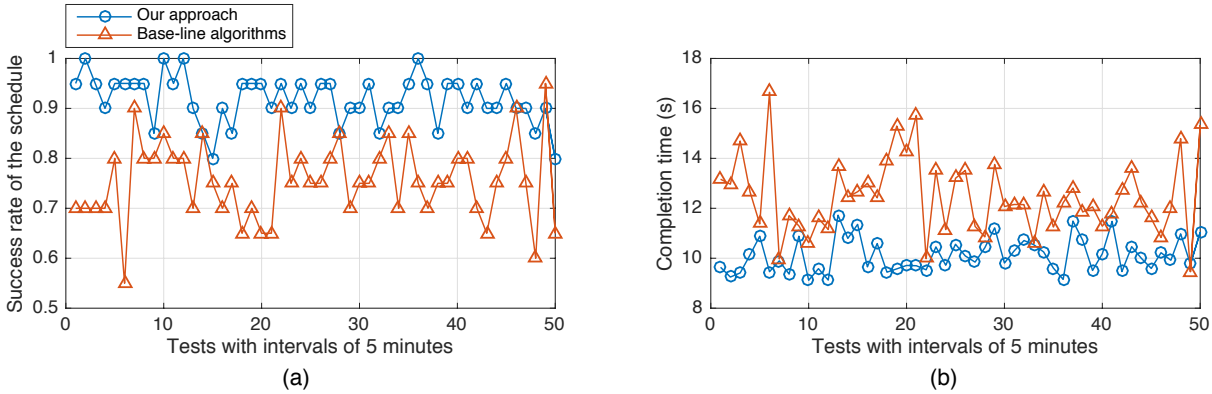


Fig. 12. Case study for random generated composition template

fully leverages service mobility with a 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 approaches, especially those who consider constant/time-invariant availability of mobile services, in terms of success rate and completion time.

The following issues should be addressed as future work: 1) Some prediction methods, e.g., time-series model, hidden Markov models and neural networks can be used to predict a user's future movement and to achieve further improved performance of composite services; 2) more metrics, e.g., service scalability and service reputation should be modeled and analyzed; 3) soft constraints should be considered. They allow to exceed a threshold value with a bounded given rate, while the corresponding algorithms for optimal run-time scheduling should be developed.

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] 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] 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.
[32] 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.
[33] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.
[34] 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.
[35] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, "Stud krill herd algorithm," *Neurocomputing*, vol. 128, pp. 363–370, 2014.
[36] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.
[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.

Qinglan Peng received the B.S. degree in software engineering from the Xinjiang University, Urumchi, China, in 2016, the M.S. degree in software engineering from the Zhejiang University, Hangzhou, China, in 2018. He is currently a Ph.D candidate in Chongqing University, Chongqing, China. His research interests include cloud computing and service computing.

MengChu Zhou (S'88-M'90-SM'93-F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of Electrical and Computer Engineering. He has over 680 publications, including 12 books, 360+ journal papers (over 260 in the IEEE TRANSACTIONS), and 28 book-chapters. His research interests include Petri nets, big data, Internet of things, semiconductor manufacturing, transportation, and energy systems.

Dr. Zhou is the Founding Editor of the IEEE Press Book Series on Systems Science and Engineering. He is a Fellow of the International Federation of Automatic Control and The American Association for the Advancement of Science.

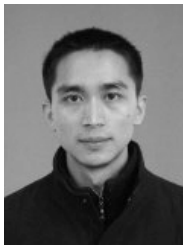
Yunni Xia (SM'14) received the B.S. degree in computer science from Chongqing University, China, 2003, and the Ph.D. degrees in computer science from Peking University (PKU), China, 2008. Since August 2008, he has been an associate professor in school of Computer Science at Chongqing University, China. He is the author or co-author of more than 50 research publications. His research interests are in Petri nets, software quality, performance evaluation, and cloud computing system dependability.

Shuiguang Deng (M'14) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

He is currently a Professor with the College of Computer Science and Technology, Zhejiang University. He was a Visiting Scholar with the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2014, and Stanford University, Stanford, CA, USA, in 2015. His research interests include service computing, business process management, and data management.

Xin Luo (M'14-SM'17) received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2016, and is currently a Professor of computer science and engineering. His research interests include big data analysis and intelligent control. He has published more than 70 papers (including 18 IEEE Transactions papers) in his related areas.

Qingsheng Zhu received his B.S., M.S., and Ph.D. degrees in computer science from Chongqing University, Chongqing, China, in 1982, 1985, and 2000. He was a Visiting Scholar at the University of London, London, U.K., from 1993 to 1994, and a Visiting Professor at the University of Illinois at Chicago, Chicago, USA, from 2001 to 2002. Currently, he is a Professor of Computer College, and the Director of the Key Laboratory of Software Theory and Technology in Chongqing University.



Wanbo Zheng received the B.S. degree in electrical and information engineering from the Chongqing University of Technology, China, in 2005, and the M.S. degree in mining engineering from the China Coal Research Institute, China, in 2009, and Ph.D degrees in computer science from Chongqing University, Chongqing, China, in 2017. His current research interests include Petri nets, performance evaluation, and trustworthy computing.