Mobility-Enabled Service Selection for Composite Services

Shuiguang Deng, Longtao Huang, Daning Hu, J. Leon Zhao, Senior Member, IEEE, and Zhaohui Wu, Senior Member, IEEE

Abstract—Mobile business is becoming a reality due to ubiquitous Internet connectivity, popular mobile devices, and widely available cloud services. However, characteristics of the mobile environment, such as mobility, unpredictability, and variation of mobile network's signal strength, present challenges in selecting optimal services for composition. Traditional QoS-aware methods that select individual services with the best QoS may not always result in the best composite service because constant mobility makes the performance of service invocation unpredictable and location-based. This paper discusses the challenges of this problem and defines it in a formal way. To solve this new research problem, we propose a mobility model, a mobility-aware QoS computation rule, and a mobility-enabled selection algorithm with teaching-learning-based optimization. The experimental simulation results demonstrate that our approach can obtain better solutions than current standard composition methods in mobile environments. The approach can obtain near-optimal solutions and has a nearly linear algorithmic complexity with respect to the problem size.

Index Terms—service selection; composition; mobile networks

1 INTRODUCTION

The development of cloud computing and mobile networks has enabled people to access Internet services using their smart mobile devices to address business anytime from anywhere [1]. As increasing numbers of enterprises are willing to provide services from the cloud, mobile business will become an important part of people’s daily lives.

In recent years, mobile networks have become service consumption and delivery platforms for many industries and government organizations worldwide. Mobile service computing is a new form of technology that combines service computing, cloud computing, and mobile networks. Millions of mobile services are downloaded around the world, generating large yearly revenues. The number of new mobile service offerings available in different fields continues to increase dramatically. With the help of mobile services, using mobile devices to access Internet services has become a life-enhancing and indispensable experience of modern life.

As the number of mobile service applications increases, more problems arise such as service reliability, service response time, and cost. These problems present new challenges to traditional service computing technologies (e.g., service discovery, service selection, and service composition). The most critical challenge is service selection [2]. Specifically, people need to determine how to select the correct service from among thousands of applicable services in the cloud and have their quality of service (QoS) met.

The problem of service selection for service composition is even more challenging. As users’ requirements become more complicated, single services barely satisfy them. Service composition technology is used to compose a complex application by integrating several distributed services that are provided by different service providers [3]. Specifically, each service composition request may include multiple tasks, and there are multiple services that can provide the same functionality for each task. Thus, we need to choose one of these services to perform the composite service with the best QoS. Traditional service selection methods for composition select the service that claims the best QoS for each task. In a mobile environment, however, it becomes more difficult for mobile users to select candidates. One consideration for invoking service(s) is context information [42]. Context information such as location, time, user identity and profile, and device capabilities impacts mobile users. Thus, one consideration for service composition is the variation of the mobile network’s signal strength in different places when the user is moving. The website Opensignal1 has recorded mobile signal coverage in the real world. From the snapshot shown in Figure 1, we observe that the distribution of signal base stations in a city is not uniform, thus the signal strength varies within kilometers. Therefore, it is

1 http://opensignal.com/
intuitive that the mobility of mobile users would impact the performance of invoked services. This paper focuses on the problem of service selection for complex service composition in a mobile environment.

Furthermore, during execution of a mobile service composition, unexpected variations in the network signal may lead to failure, making it necessary to include adaptation mechanisms for the mobile service composition. In this paper our focus is on the selection phase; in later work we will focus on adapting the mobile service composition during its execution.

In this paper, we propose a new approach for service selection for service composition in a mobile environment, consisting of the following five contributions.

1. We identify the new research problem of service selection for mobile service composition. We also provide formal definitions to describe this problem.

2. We propose a mobility model that formally models service invocations in a mobile environment. The model consists of two parts: the paths of moving mobile users when they are invoking service compositions and the quality of the mobile network when transmitting data for component services.

3. We specify a mobility-aware QoS computation rule that allows computing mobility-aware QoS for service composition. Our mobility-aware QoS computation can also handle the QoS of structured service compositions with data dependence.

4. We develop a mobility-aware service selection algorithm by utilizing the teaching-learning-based optimization (TLBO) algorithm [5]. Based on the proposed mobility model, we tailor the operations of TLBO to the problem of service selection for mobile service composition to improve the solution quality and scalability of our algorithm.

5. We conduct a series of evaluations to validate that the global QoS of service compositions computed by our algorithm is approximately optimal and much better than that of current standard composition approaches. Furthermore, we compare our methods with other population-based optimization methods on both optimality and scalability.

The remainder of this paper is organized as follows. Section 2 details two examples that illustrate our motivation. Section 3 introduces definitions related to the problem. Section 4 describes our approach to selecting services for mobile service composition. Section 5 presents the experimental simulation evaluation and analysis. Section 6 reviews related work. Section 7 presents the paper’s conclusions and explores possible future work.

2 Motivation Scenarios

Because of the mobility of mobile users and the dynamics of mobile networks, service selection in a mobile environment is notably different from that in the traditional Internet environment. We will outline the differences using examples for single service selection and composite service selection.

2.1 Single Service Selection

Figure 2 illustrates an example of a single service selection in a mobile network.

Assume a mobile user Tom wants to invoke a hotel booking service when he is walking from base station A to base station B. Assume that the signal strength of B is stronger than A, the average data transmission rate between Tom’s cellphone and A is 10 Kbps, and the data transmission rate between Tom’s cellphone and B is 20 Kbps. A virtual service provider sp is responsible for selecting the service with the best response time for Tom. sp can get the functional and non-functional attributes of service candidates from UDDI. It can also obtain knowledge of the mobile network capability directly from the telecom service provider or from some third parties that monitor signals such as Opensignal2. Thus, sp can make a selection for Tom based on the acquired information. Suppose that sp finds two candidates that can provide hotel booking service, Ctrip and Elong, which are well-known hotel booking services in China. sp would make the selection decision for Tom depending on the QoS of each service. The booking confirmation wait time is 100 s for Ctrip and 120 s for Elong.

In the traditional Internet environment, it is intuitive

2 http://opensignal.com/
that \( sp \) would select \( Ctrip \) for Tom because \( Ctrip \) performs faster than \( Elong \). However, Tom is moving when he invokes the services. He sends the hotel booking request while at location \( a \) (we assume the time of sending a request is 1 s). If \( sp \) selects \( Ctrip \) for Tom, he would obtain the response when at location \( b \) and start receiving confirmation with 1000Kb of data. Because of the handover principles of cellular networks [4], Tom would not switch his connection to station \( B \) as soon as he gets into its coverage area. He would continue the connection with station \( A \) until its signal strength is lower than a threshold (we set it to zero for simplification). Finally, Tom finishes receiving the confirmation at location \( c \). Thus, the total service time of \( Ctrip \) is \( 1 \text{s} + 100 \text{s} + 1000/10 = 201 \text{s} \). If Tom selects the service \( Elong \), he would have moved to location \( b' \) before he begins to receive the confirmation. Next, he would establish a connection with station \( B \) because of its stronger signal. Finally, Tom finishes receiving the confirmation at location \( c' \). The total service time of \( Elong \) is \( 1 \text{s} + 120 \text{s} + 1000/20 = 171 \text{s} \). Thus, it is faster for Tom if \( sp \) selects \( Elong \) even though its response time is longer.

Hence, we can see that service selection in a mobile environment is quite different from that in a traditional environment. It is essential to take users’ mobility into consideration when selecting services in mobile networks.

### 2.2 Composite Service Selection

Figure 3 illustrates a more complicated example of service selection for service composition.

This time Tom wants to arrange business travel to Beijing. He needs to know the weather conditions, book a flight, book a hotel, and pay. To this end, the virtual service provider \( sp \) will compose multiple mobile services from different providers for Tom. Figure 3(a) shows the candidates for each task found by \( sp \). Figure 3(b) shows that Tom will invoke the composed services when he goes to work by subway.

If \( sp \) utilized a traditional approach for service selection, the service composition would be composed of the fastest candidate for each task. However, data transmission time may vary as Tom travels to work. The data transmission time cannot be guaranteed by traditional methods. As Figure 3(b) shows, the total response time of the whole service composition is 850 s using traditional methods. If \( sp \) considers Tom’s mobility when selecting services, \( sp \) may choose some suboptimal candidates but the data transmission time can be reduced. The total response time of the whole service composition is now 830 s.

Hence, it is important to consider users’ mobility when selecting services for service composition in mobile networks. This is more complicated than the individual service selection problem because a different selection for one task may result in issuing the following task from a different place, which could affect its data transmission time.

---

**3 PREREQUISITE DEFINITIONS**

In this section, we give clear definitions of the key concepts in the scope of service selection for service composition in mobile networks. The formal framework for service composition in [6] is adopted and extended.

**Definition 1. (Service).** A service is modeled as a triple \( s = (I, O, Q) \) where:

1. \( I \) is the set of input parameters;
2. \( O \) is the set of output parameters;
3. \( Q \) is an \( n \)-tuple \( <q_1, q_2, ..., q_n> \), where each \( q_i \) denotes a QoS property of \( s \) such as cost, response time, throughput, or availability. In this paper, we only consider one QoS property (response time). This is because users’ mobility only affects the variation of the signal strength during the invocation of services, which would only affect the time spent on data transmission. So this may affect the QoS properties related to response time. For example, if the service is charged for time, then the cost would also be affected by user mobility. But the af-
fection is consistent with the response time. That is, the gains obtained by optimizing response time would not negatively affect other QoS properties. In the rest of the paper, the QoS of a service is equivalent to the response time of a service.

Definition 2. (Service Invocation). A service invocation is modeled as a triple \( inv = (s, d_i, d_o) \), where:
1. \( s \) is the invoked service;
2. \( d_i \) is the volume of the input data to \( s \);
3. \( d_o \) is the volume of the output data from \( s \).

Note that a service \( s \) may have different service invocations because the input data may be different. Similarly, services with the same \( I \) and \( O \) may also have different service invocations for the same user request. For example, given two image processing services \( s_1 \) and \( s_2 \), their inputs and outputs are all jpg files. However, \( s_1 \) requires that the image not exceed 200 KB, while \( s_2 \) requires that the image must be larger than 300 KB. Thus, it becomes challenging to select proper services in mobile networks because the quality of the mobile network may affect the data transmission for different service invocations.

Definition 3. (Service Composition Plan). A service composition plan is modeled as a 2-tuple \( scp = (T, R) \), where:
1. \( T = \{t_1, t_2, \ldots, t_n\} \) is a set of tasks; each task \( t_i \) can be implemented by a set of candidate services \( C_i \);
2. \( R = \{r(t_i, t_j) | t_i \in T, t_j \in T\} \) is a set of relations between tasks in \( T \). \( r(t_i, t_j) \) represents that the inputs of \( t_j \) depend on the outputs of \( t_i \). \( R \) is used to describe the structure of the service composition plan.

Definition 4. (Service Composition). A service composition is modeled as a triple \( sc = (p, S, Q) \), where:
1. \( p \) is the service composition plan to which \( sc \) corresponds;
2. \( S \) is the set of services in \( sc \). Each \( s_i \in S \) corresponds to a \( t_i \in T \);
3. \( Q \) is the global QoS of \( sc \). There exist many computation methods to achieve the global QoS by integrating the QoS of each service [7-10].

Generally, the process of service composition has three steps as shown in Figure 4:

- **Composition Planning.** The first step is to build the composition plan with multiple tasks (abstract services). Unlike a concrete service, each task is the symbol representing a group of candidate services with similar functions and interfaces. The tasks are composed together by some control statements (such as assignment, switch, and loop).

- **Candidate Discovery.** After the composition plan is generated, it should discover suitable candidates for each task. These candidate services have the same functions and interfaces as the task. This step can be implemented through service discovery or service recommendation mechanisms, which discover/recommend the first \( k \) services for each task in the service repository with some non-functional constraints or personal preferences. The service discovery/recommendation mechanisms can be provided by the virtual service provider [11]. But this is not the focus of this paper.

- **Service Selection.** Service selection is a key part of service composition. The optimal service from each candidate service group is selected or recommended and replaces the corresponding task in the composition plan [40]. In this study, we focus on how to select concrete services for each task to obtain the optimal service composition. However, it is also important to guarantee that the services for different tasks are compatible. This issue is beyond the scope of this paper, and there is considerable research on this topic [12-13]. In this study, we assume that all candidates for different tasks can interact with each other.

4 MOBILITY-AWARE APPROACH

In this section we introduce the details of our approach. We present the mobility model and describe our mobility-aware QoS computation rule based on this model. Then we present our proposed mobility-aware selection method based on the teaching-learning-based optimization algorithm.

4.1 Mobility Model

In a mobile scenario, we face the challenge of managing users’ mobility; that is, users are moving when they invoke a service composition. While users are moving, the mobile network latency for transmitting input/output data for services varies depending on their location. Thus, our mobility model consists of two parts: the user’s path and the quality of the mobile network.

Definition 5. (User’s Path). A user’s path is modeled as a triple \( mp = (Time, Location, M) \), where:
1. \( Time \) is the time span during which the user is moving. It includes a set of continuous time points;
2. \( Location \) is the set of the user’s locations corresponding to all time points in \( Time \);
3. \( M \) is a function that maps time points to the user’s locations on the motion path. \( M: Time \rightarrow Location \). The function \( M \) can be implemented by the random waypoint mobility model [41].

Definition 6. (Quality of the Mobile Network). The...
quality of the mobile network (QoMN) usually describes the mobile signal strength at a specific location. In this paper, we mainly consider the data transmission rate as the quality of the mobile network. The function \( L \) is used to map locations to QoMN, \( L: \text{Location} \rightarrow \text{QoMN} \).

![Fig. 5. Mobility Model](image)

Figure 5 shows an example that represents the mobility model in Figure 3(b). First we draw Tom’s path in a two-dimensional space; we obtain the location of Tom at any specific time point via function \( M \). We can find the location where Tom starts to send or receive data for a service invocation by calculating the time points of service invocations. For example, Tom starts invoking the weather forecast service at the time point \( t_0 \) and receives a response at time point \( t_1 \). Meanwhile, given a location, we can obtain the quality of the mobile network corresponding to this location. Thus, the mobile network latency for transmitting input/output data can be calculated. Lastly, the final response time of the whole service composition can be calculated.

![Fig. 6. Mobility Model Grid](image)

To make the mobility model computable, we build an overlay grid on the two-dimensional space, as in Figure 6. Each cell of the grid corresponds to an area of constant QoMN. This is equivalent to the practical situation when the cells are infinitely small. We can approximately measure the QoMN of the area covering the user’s path and build this grid by associating QoMN values with each cell. Based on the mobility model, we can define the problem that is the focus of this paper.

We now discuss how users’ mobility information can be acquired. There are three ways to obtain the moving path of a mobile user. First, the routine path of a mobile user can be recorded by some applications/services (e.g., Apple Location Service) based on the historic behavior of the mobile user. For example, a user’s path to work every day may not change much. Then such mobility information (geographic path, duration, speed, etc.) can be estimated based on recording his historic performance. Alternatively, the moving path of a mobile user can be obtained by some mobility prediction methods [43, 44]. For example, a Wi-Fi access point service can log the time instances that mobile nodes associate/disassociate with it. Such information can be used to track the mobility of a particular mobile node. In addition, a service requester may volunteer to report his/her estimated moving path to the virtual service provider. The incentive for a service requester to disclose such information is the potential to save time and money on mobile connection charges. We note that similar incentive mechanisms have been studied in mobile ad hoc networks to achieve cooperative routing and forwarding [45].

Another challenge of adding a mobility model to service selection in composite services is the question of who is responsible for maintaining the model and generating service response time. Telecom service providers have sufficient capability to maintain the proposed mobility model. Firstly, the location information of mobile users can be monitored by GPS or other location-based services provided by the telecom service providers [46]. Additionally, the QoMN knowledge can be obtained through monitoring the mobile network. For example, Opensignal\(^3\) can provide the coverage and quality of cellular networks from different telecom service providers. Hence, telecom service providers are able to implement the path function \( M \) and the mobile network quality function \( L \). Therefore, they are ideal maintainers of the proposed mobility model, which makes the model practical.

### 4.2 Mobility-aware QoS Computation

In this section, we first introduce the concept of mobility-aware QoS (MQoS) based on the mobility model. Then we present how to compute the global QoS of the mobile service composition.

**Assumption 1.** The QoMN remains constant during the time each task is being transmitted.

The QoMN actually does vary during data transmission. But the variation is not obvious and has a limited impact on the time spent on data transmission. Specifically, the data volume for a mobile service is normally within several KB and the data transmission rate is usually in Kbps. So a complete data transmission takes several seconds. A mobile user moves at most several meters in seconds. The coverage of a signal base station can reach several kilometers. Thus, there isn’t obvious variation of signal strength across several meters. Therefore, we propose Assumption 1 for ease of presentation.

**Definition 7. (Mobility-aware QoS).** Mobility-aware QoS (MQoS) describes the performance of a component service in a mobile service composition consideration. In this paper, we consider only one property of QoS (response time). The MQoS of a component service \( s \) can be calculated as follows:

\[ \text{MQoS} = \text{QoMN} \times \text{response time} \]

\(^3\)http://opensignal.com/
where $i = d_i + Q_{O_M} + t_{d_i}$

where $d_i$ is the volume of input data and $Q_{O_M}$ is the QoM at the location from which the user starts sending the input data. Based on **Assumption 1**, $Q_{O_M}$ will not change during $t_{d_i}$, $d_o$ and $Q_{O_M}$ are the corresponding variables for the output data.

Consider the example in Figure 3(b). To calculate the mobility-aware QoS of a weather forecast service $s$, we first need the time point $tp$ when the user starts sending the input data; the user’s location at the time point $tp$ can be found through the function $M: location=M(tp)$. The quality of the mobile network at that location is derived through the function $L: Q_{O_M}=L(location)$. Next, $t_{d_i}$ is calculated with Equation (2). $t_{d_i}$ is computed similarly. Finally, the $QoM$ of $s$ can be found with Equation (1).

**Definition 8. (Global QoS).** Global QoS describes the performance of the entire service composition. The global response time of a service composition can be calculated as follows:

\[ GQoS = \psi \sum_{i=1}^{n} MQoS_i \]

where $\psi$ is an operator that integrates the values of local QoS. We adopt the QoS integration rules in [10] to implement $\psi$, as shown in Table 1, where $\psi_1$ is the integration function for QoS of services in a sequential execution path. $\psi_2$ is the integration for QoS of multiple parallel paths. For the notations in the table, we only use their intuitive mathematical meanings. For example, "\(\sum\)" means summation, "\(\prod\)" means product, "max" means maximum, and "\(\min\)" means minimum. The optimal QoS of a composition is the best value obtained from the integration rules. For simplicity, we consider only a 1-dimensional QoS value (response time) in this paper. It is not difficult to extend to other criteria by aggregating the overall QoS value of the service composition through the computation rules. If an efficient aggregating function of multiple QoS properties is provided, our proposal can also handle QoS values of multiple dimensions.

<table>
<thead>
<tr>
<th>QoS property</th>
<th>$\psi_1$</th>
<th>$\psi_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost</td>
<td>$\sum$</td>
<td>$\sum$</td>
</tr>
<tr>
<td>response time</td>
<td>$\sum$</td>
<td>$\text{Max}$</td>
</tr>
</tbody>
</table>

Based on the above mobility model and the mobility-aware QoS computation, we can define the problem in this paper as:

**Definition 9. (Service selection for service composition in mobile networks).** Given a user’s path and the QoM within the area covering the path, for a service composition required by the mobile user, select concrete services from service candidates to obtain the optimal global QoS (GQoS).

### 4.3 Mobility-enabled Service Selection

Our selection algorithm is based on the teaching-learning-based optimization algorithm (TLBO), which belongs to the “swarm intelligence” optimization methods. First, we illustrate how our problem is transformed to an optimization problem. Then, we introduce our service selection algorithm based on TLBO.

**a) Optimization Problem**

An optimization problem is to find the smallest $F(\Theta)$ with a feasible parameter vector $\Theta$, which can be modeled as follows [14]:

\[
\begin{align*}
\text{Min} & \quad F(\Theta) \\
\text{Subject to} & \quad \theta_i \in [1, N] \\
& \quad \theta_i \in Z
\end{align*}
\]

This means the feasible set of parameter vectors is constrained by $\theta_i \in [1, N]$ and is an integer. The optimal solution $\Theta$ satisfies the following conditions:

1. $\Theta$ belongs to the feasible set;
2. $\forall \Theta, F(\Theta) \leq F(\Theta)$

The following theorem presents the relationship between this optimization problem and our mobility-aware service selection problem.

**Theorem 1. (Optimization Problem).** A mobility-aware service selection problem with a user’s given path and a given quality of mobile network in this area is equivalent to the optimization problem described in (4).

**Proof** For the problem of selecting optimal services with shortest response time while considering mobility, the vector $\Theta = (\theta_1, \ldots, \theta_m)$ can describe a possible solution as a service composition with $m$ tasks. An element $\theta_i$ in $\Theta$ corresponds to a selected service from the candidates for the $i$-th task. The evaluation function for the parameter vector $\Theta$ can be implemented by using Equation (3):

\[ F(\Theta) = \psi \sum_{i=1}^{n} MQoS_{\theta_i} \]

The target of the mobility-aware service selection problem is to find a $\Theta$ to obtain the smallest $F(\Theta)$. Thus the problem is equivalent to the optimization problem described in Equation (4).

The problem in Equation (4) is an integer programming problem (IP), which is a famous NP problem. Generally, there is no known algorithm with Non-deterministic Polynomial time complexity to solve such a problem. Thus, we propose a solution method based on the TLBO algorithm,
which can achieve an approximately optimal solution in polynomial time.

b) Service Selection Algorithm

In this section, we give a basic overview of the TLBO algorithm. We then introduce our customized algorithm to solve the problem of service selection for mobile service composition.

(1) Overview of TLBO

The teaching-learning-based optimization algorithm was first proposed by Rao and Kalyankar [5]. Like other nature-inspired algorithms, TLBO is a population-based method that uses a population of solutions to proceed to the global solution. For TLBO, the population is considered to be a group or class of learners. For our mobile service composition problem, each learner in the population corresponds to a feasible service composition. Moreover, different tasks in the service composition plan are analogous to different subjects offered to learners; the learners’ results are analogous to ‘fitness’, as in other population-based optimization techniques. The teacher is considered to be the best solution obtained so far. Table 2 shows the analogous term matches between the TLBO and the service composition domains.

<table>
<thead>
<tr>
<th>Terms in TLBO</th>
<th>Terms in Service Composition Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>The optimal service composition</td>
</tr>
<tr>
<td>Learner</td>
<td>One feasible service composition</td>
</tr>
<tr>
<td>Class</td>
<td>The feasible set of service compositions</td>
</tr>
<tr>
<td>Subjects</td>
<td>Tasks in the service composition plan</td>
</tr>
<tr>
<td>Grade</td>
<td>Fitness (QoS) of a service composition</td>
</tr>
</tbody>
</table>

TLBO consists of two parts: ‘Teacher Phase’ and ‘Student Phase’. The ‘Teacher Phase’ means learning from the teacher, and the ‘Student Phase’ means learning through interactions between learners (Figure 7).

(2) Initialization Phase

One advantage of TLBO is that there are not as many parameters to be tuned as in other population-based methods. Only two basic parameters need be decided in the initialization phase. One is the population size $P$, and the other is the maximum iteration number $I$. Next, the initial population is generated randomly.

For each learner in the class, $X' = (x'_1, x'_2, ..., x'_{d})$ is generated randomly, where:

$$i = (1, 2, 3, ..., P)$$

and $d$ is the number of tasks in the service composition plan. $x'_i$ is the selected candidate for the $j$-th task in solution $X'$, an integer that represents the selected candidate.

(3) Teacher Phase

In the teacher phase of TLBO, every learner $X'$ ($i = 1, 2, 3, ..., P$) in the class learns from the teacher $X_{teacher}$ through the difference between the teacher $X_{teacher}$ and the mean value of the learners, $Mean$:

$$X_{new} = X_{old} + \text{difference}$$

where $X_{old}$ and $X_{new}$ is the $i$-th learner before and after learning from the teacher, $r_i$ is the learning step-length, $TF_i$ is the teaching factor, and $Mean$ is the average of all learners:

$$Mean = \frac{1}{P} \sum_{i = 1}^{P} X'_i$$

In our mobile service composition problem, each variable in a solution vector $X$ must be an integer. We therefore add a refine operation for TLBO after each vector operation:

$$def \text{refine}(X') :$$

$$x_j = round(x_j)$$

if $x_j > up : x_j = up$

if $x_j < low : x_j = low$

where $up$ and $low$ are the upper and lower bounds of the candidates, respectively.

After learning from the teacher, all learners update themselves with the learned result:

$$\text{if } F(X_{new}) < F(X_{old}) \text{ : }$$

$$X_{old} = X_{new}$$

where the function $F$ is used to calculate the fitness of the learner according to Equation (5).

(4) Student Phase

Learners increase their knowledge through interaction between themselves in the student phase. A learner learns...
something new if another learner has more knowledge than he or she. This keeps the population diverse, which can avoid the algorithm converging too early to obtain a good result.

For each learner in the class, \( X' = (x'_1, x'_2, ..., x'_d) \) will randomly choose a learning target \( X^* = (x^*_1, x^*_2, ..., x^*_d) \) \( i \neq j \). \( X' \) will analyze how it is different from \( X^* \), then make the learning decision:

\[
X'_{\text{new}} = \begin{cases} X'_{\text{old}} + r \cdot (X^* - X') & F(X') > F(X^*) \\ X'_{\text{old}} + r \cdot (X^* - X') & F(X') < F(X^*) \end{cases}
\]

(11)

where \( r = \text{rand}(0,1) \) is the learning step-length.

After learning between learners themselves, learners then update themselves as in (10).

c) Time Complexity Analysis

Suppose the number of learners in the population is \( P \), the number of tasks in a service composition plan is \( d \), and the maximum iteration number is \( I \). The process of the proposed method can be summarized as follows:

**Algorithm: TLBO**

**01:** (1) Initialization: randomly generate \( P \) service compositions.
**02:** (2) Calculate the fitness of every composition
**03:** For \( i = 1:P \)
**04:** (3) Decide which is the teacher in the class;
**07:** For \( j = 1:d \)
**12:** | (4) Update the candidate for each task by learning from the teacher through Equations (6) and (10);
**13:** | (5) Update the candidate for each task by teaching from other learners following Equations (9) and (10);

**14:** EndFor
**15:** EndFor
**16:** (6) If the iteration number reaches \( I \), the algorithm terminates. Otherwise, return to step (2)

We can see that the main computation time is spent in steps (4) and (5): computing the difference between learners and the teacher/learners and then updating the candidates for each task. The time complexity of each step is \( O(d) \). Thus, for each learner in the population, the time complexity of the teacher and learner phases is \( O(d) \). Next, the time complexity for the whole population is \( O(Pd) \). Finally, the overall time complexity with \( I \) iterations is \( O(IP^2d) \).

## 5 Simulation Experiments and Analysis

In this section, we describe a series of simulations we conducted to evaluate and validate our proposed approach. The experiments were designed to answer the following questions:

1) Why is it necessary to consider user mobility when selecting services for service composition in a mobile environment? What is the impact of mobility?

2) Compared with other metaheuristic algorithm-based methods, can our approach find more optimal results?

3) Compared with other metaheuristic algorithm-based methods, how does our approach perform as to scalability?

### 5.1 Setup

The evaluation was run on a machine with an Intel Core 2.3 GHz i7 CPU. All algorithms were implemented in Python 2.7 and evaluated sequentially; they were given up to a maximum of 8 GB of memory if needed. We generated our service compositions with randomly inserted tasks and control structures. Note that for each generated composition process, we only consider selection of a candidate service for each task so that all candidate composite services have the same structure. For each task, we randomly created a number of candidate services, each with a different QoS. To avoid other factors affecting the evaluation results, we set it up so that the candidates for the same task require the same amount of input/output data. The response time of each service was generated from a uniform distribution. Figure 8 depicts an example of a generated service composition with 5 tasks.

![Fig. 8. Example Service Composition with 5 tasks](image)

For each implemented algorithm the population size was \( P = 10 \), the maximum iteration number was 100, and all the algorithms were run independently 50 times for unprejudiced statistcal results.

### 5.2 Impact of Mobility

To validate the necessity of considering mobility when making selections for mobile service composition, we compared our method with the standard composition method that considers only services’ individual QoS. This method selects the candidate with the optimal response time for each task without considering mobile network latency. That is, for each task, it only considers \( Q_i \) for Equation (1) and selects the candidate service with the best \( Q_i \). For the experiments in this section, we generated service compositions with 50 tasks and 100 candidates available per task.

#### a) Impact of Variation of Signal Strength

In the proposed mobility model, the two functions \( M \) and \( L \) determine the relationship between users’ movements and the variation of signal strength. For our experiments, we combined the user movement function \( location=M(time) \) and the mobile signal-strength function \( Q0MN=L(location) \) into one function of signal strength: \( Q0MN=G(time)=L(M(time)) \), which gives the signal strength with increasing time. Because there are many factors affecting the variation of signal strength (such as a user’s speed, distance from signal stations, properties of each signal station, etc.), the variation of signal strength in...
practice cannot be defined by any functions, and it is difficult to acquire the real data of signal strength. Therefore, we generated four different functions of signal strength to simulate the variation of signal strength during users’ movements:

\( G_1: \) Constant function. We set \( G_1 = a \). This function simulates a traditional environment and was used to evaluate how our method compares to the standard method for traditional service composition.

\( G_2: \) Cosine function. We set \( G_2 = a \cdot (\cos(b \cdot \text{time}) + 1) + 1 \), which makes the signal strength vary between 1 and \( a + 1 \) with a variation cycle of \( \frac{2\pi}{b} \). This function simulates a user moving with constant velocity and regularly distributed base stations.

\( G_3: \) Piecewise function. We set \( G_3 \) to be a piecewise function: \( G_3 = \begin{cases} a \cdot \text{time} & \text{if } \text{time} \leq t_1 \\ b \cdot \text{time} & \text{if } t_1 < \text{time} \leq t_2 \\ c \cdot \text{time} & \text{if } t_2 < \text{time} \leq t_3 \\ \vdots \end{cases} \). This function simulates the user in different periods and that the signal strength changes in each period.

\( G_4: \) Random function. We set \( G_4 = \text{random}(a, b) \). This function simulates unpredictable situations.

Figure 9 shows the comparison results of the four signal-strength functions; the x-axes are the iteration number and the y-axes are the response times of the service compositions found by the two methods. Figure 9(a) shows that the standard method can find better service compositions (with shorter response times) than our method. The standard method can guarantee to find the optimal solution with the shortest response time because the mobile network latency is always the same with function \( G_1 \). However, our method approaches the optimal result closely after a sufficient number of iterations, which validates that our method can find near-optimal solutions.

Figure 9(b-d) shows that our proposed method outperforms the standard method with the three variations of the signal-strength function. This indicates that the standard method is not effective at finding compositions when signal strength varies in a mobile environment. We also find that our method performs better whether the signal strength varies regularly or randomly. We can also observe that the response time of composition result obtained by the standard method varies with different variations of the signal-strength function. Actually, the composition result obtained by the standard method is the same one with different variations of the signal-strength.
function. But as the signal strength changes during the execution of the composition, the time spent on data transmission would change. Thus, this may affect the response time of the composition result.

b) Impact of Amplitude of Variation of Signal Strength

In this experiment, we aim to evaluate the impact of the amplitude of the variation of signal strength (that is, how much the signal strength varied) on the improvement of our approach. To this end, we selected the cosine function $G_2$ as the variation function of signal strength because of the ease of adjusting the amplitude of variation through tuning parameter $a$ in $G_2$. We set the range of $a$ from 60 to 120, and $b=1$. The population size was $P=10$; the maximum iteration number was 100. We used the metric $improve_{rate}$ to evaluate how much better our method is compared to the standard method.

$$improve_{rate} = \frac{r_s - r_{sm}}{r_s}$$

(12)

where $r_s$ is the optimal result achieved by the standard method, and $r_{sm}$ is the optimal result from our method.

Figure 10 shows the improvement of our method with different values of the amplitude of the variation of signal strength. The average improvement using our method is approximately 20% compared to the standard composition method. We observe that as the amplitude of variation increases, there is no obvious regularity in how $improve_{rate}$ varies. This indicates that our method outperforms the standard method no matter how the amplitude of variation of signal strength varies because the values of $improve_{rate}$ are always positive. Furthermore, we note that the improvement fluctuates with different amplitude values; this is because the improvement of the mobile network latency cannot remain fixed under all conditions.

c) Impact of Frequency of Variation of Signal Strength

We also evaluated the impact of the frequency of the variation of signal strength on the improvement of our approach using the cosine function $G_2$ as the variation function of signal strength. The frequency was adjusted by tuning parameter $b$ in $G_2$. We set $b$ to range from 1 to 50, and $a=100$. The same $improve_{rate}$ metric was used to evaluate the improvement of our method compared to the standard method.

Figure 11 shows the improvement of our method with different values of the frequency of the variation of signal strength. The results show that as the frequency of variation increases, the $improve_{rate}$ value increases initially. When the frequency passes over a certain threshold, the $improve_{rate}$ value begins to decrease. Thus, we can conclude that our method cannot continue to improve as signal strength changes more quickly. There are two reasons for this: 1) the initial increase of $improve_{rate}$ confirms the intuitive notion that a relatively obvious variation of signal strength leads to better improvement; 2) When the variation frequency surpasses a certain threshold, it may cause the signal strength to vary so quickly that it returns to its previous value when the user re-issues a data request; thus there is less change in signal strength from the user’s perspective.

5.3 Optimality Evaluation

Because few studies have investigated the problem of service selection for mobile service composition, we have not found many existing methods to compare with ours. Because TLBO is a fundamental population-based algorithm, we chose several other population-based algorithms and compared their optimality with our method.

Genetic Algorithm (GA). A search heuristic algorithm that mimics the process of natural selection [15]. This has been used by existing service composition approaches [16-18]. We extended it by adding mobility.

Particle Swarm Optimization (PSO). A computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality [19]. PSO has also been widely utilized in service composition research [20-22].

Negative Selection Algorithm (NSA). NSA belongs to the field of Artificial Immune Systems inspired by theoretical immunology and observed immune functions, principles, and models [23]. It has recently been used in solving service composition problems and proved to have high efficiency [24-25].
our method outperforms the others no matter what the number of candidate services is.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameter Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>cross rate=0.7; mutate rate=0.3</td>
</tr>
<tr>
<td>PSO</td>
<td>$c_1=c_2=2$; weight=0.8</td>
</tr>
<tr>
<td>NSA</td>
<td>$\alpha=0.1$; $\beta=3$; $\rho=1$</td>
</tr>
</tbody>
</table>

To compare the above algorithms with our approach, we tuned the parameters for each algorithm to achieve their best performance. The most suitable parameters are shown in Table 3. We generated service compositions with sizes between 10 and 100 (in steps of 10). We varied the number of candidate services available per task between 100 and 1,000 (in steps of 100), which is considerably more than most of the previous studies used.

To evaluate the optimality of the algorithms, we plotted the total response time of the service compositions found by all algorithms versus increasing problem size.

**a) Impact of Different Task Number**

Figure 12 plots the response time of the optimal service composition achieved by different algorithms against an increasing number of tasks with a fixed number (100) of services available per task.

From the comparison results in Figure 12, we observe that our method has outstanding performance of solution optimality with all different task numbers. The response time of the optimal service composition returned by TLBO is at least 10% lower than others, the advantage becoming more obvious with the increasing numbers of tasks. Thus, we can conclude that our approach manages to achieve a good approximation ratio of the optimal solution regardless of the composition size.

**b) Impact of Different Numbers of Candidates**

Figure 13 shows that the optimal response time decreases slightly as the number of services per task increases. This is because there are more choices available as the number of services increases. We also observe that

From the comparisons in this and the previous sections, we observe that our approach maintains good performance with large-scale datasets of both tasks and candidates. This is because TLBO keeps the population diverse through the learning phase, which efficiently avoids converging to a suboptimal value too early. PSO performs better than GA because it has an evolution target in each generation, but this can also result in early convergence. For the problem we address, NSA performs worst, which is not consistent with previous research in [24]. This is because NSA-based methods focus more on local fitness, which is efficient in traditional service composition problems but not suitable in a mobile environment. In summary, our proposed TLBO-based method has the best performance for optimality.

**5.4 Scalability Evaluation**

In this section, we compared the scalability of our approach with the other algorithms using the same settings as for optimality.

**a) Impact of Different Numbers of Tasks**
Figure 14 shows that GA runs faster than the other three algorithms. However, the qualities of its solutions are much worse than those of PSO and TLBO, even though it takes much less than 100 ms to compute. It seems that GA is faster only because it fails to improve the quality of its solutions significantly, thus converging more quickly to a bad local optimum. Similarly, NSA also runs faster than PSO and TLBO, but the quality of NSA solutions is even worse than those of GA. PSO and TLBO obtain much better qualities of solutions, but they sacrifice some runtime to improve the optimality.

Furthermore, TLBO takes longer than PSO. This is because TLBO adds a learning phase to avoid early convergence that takes considerable runtime. Although TLBO uses a little more runtime than the other algorithms, it has a low algorithmic complexity, which is roughly linear with regard to the composition size. This observation validates the time complexity analysis in Section 4.3.

b) Impact of Different Numbers of Candidates

In Figure 15 we note that the runtime of NSA increases significantly as the number of services per task increases. Thus, NSA quickly becomes infeasible for practical purposes; it takes 6 times longer compared to the other algorithms for 1,000 services per task. On the other hand, the runtimes of the other three algorithms do not change as much with an increasing number of services per task. Hence, all three algorithms scale well in this regard in our scenario.

![Image of runtime per iteration with different numbers of services per task]

**Fig. 15.** Runtime per iteration with different numbers of services per task

6 RELATED WORK

Because our proposal targets the problem of service selection for mobile service composition, we review related work from the following two aspects: 1) service selection for service composition on the traditional Internet; 2) service selection for service composition in a mobile environment.

6.1 Traditional Service Composition

With the increasing popularity of employing service composition technology for distributed systems, more researchers have focused in recent years on the problem of quality of service (QoS)-aware web service selection. Because this paper focuses on the selection for service compositions, we mainly review the work on this topic. There have been many algorithms and models adapted to solving this problem, such as integer programming, the multi-dimensional multi-choice 0-1 knapsack problem, the genetic algorithm, and particle swarm optimization.

Zeng et al. [26] addressed the QoS-aware composition problem by formalizing and solving it with a Linear Integer Programming (LIP) approach, which is still widely applied to obtain optimal solutions for the composition problem. Tsesmetzis et al. [27] introduced the “Selective Multiple Choice Knapsack Problem” that selects services to maximize the provider’s profit. Wang et al. [28] proposed a fuzzy linear programming method to identify the dissimilarities of service candidates. This method can help service consumers select the most suitable services by considering their requirements and preferences. Sun and Zhao [29] presented a constraint decomposition-based approach for service composition. These researchers computed the utility of a composite service from the utilities of component services. Conversely, the constraints of component services can also be decomposed from the constraints of a composite service. Ai et al. [30] proposed a penalty-based genetic algorithm for the problem of decentralized composing of composite web services. They covered issues ranging from code partitioning for decentralization to detailed discussion of the servers that participate in decentralized execution. Luo et al. [31] developed a heuristic algorithm for web service composition with end-to-end QoS constraints. This method is based on a generic cross-entropy algorithm adopted for the combinatorial optimization problems. Ma et al. [32] extended the genetic algorithm for service composition by providing strategies to keep population diversity and enhance the initial population. Jiang et al. [33] proposed a PSO-based algorithm to solve the web service selection problem. The basic PSO method was enhanced with adaptive weight adjustment and non-uniform mutation strategies. Wang et al. [34] proposed a comprehensive evaluation model for service composition based on the generic QoS and domain QoS of the composite web services. They built a hybrid culture max-min ant system to solve the service selection problem.

This previous work on traditional service composition forms the foundation for our research in this paper. We re-cast the problem into an optimization problem and utilize a population-based method to solve it. The difference is that we modeled the service composition problem in a mobile environment in which it is not easy to apply the traditional methods.

6.2 Mobile Service Composition

As mobile devices become more popular, future service composition will need to be more flexible and complex [35, 36]. Because the topic of mobile service composition has considerable potential, several researchers have begun working on mobile service composition.

Fortuna and Mohorcic [37] presented an overview of dynamic service composition in both wired and wireless environments.
networks. They also discussed other issues related to service composition such as service discovery and service negotiation. However, they did not present technical details as to how the service composition is carried out. Furthermore, the methods proposed in their paper are not specifically for mobile networks. Wang et al. [38] introduced a mobility prediction method for the problem of dependable service composition in wireless mobile ad hoc networks. Their goal was to compose a service that can tolerate the uncertain mobility of service providers. To this end, the authors proposed two criteria to model the dependability of the service composition, with and without using probability distribution information to characterize the service provider mobility. However, this work does not seek the optimal QoS service compositions. Furthermore, the structure of service composition in this paper is limited to sequential workflows. Luo et al. [39] proposed a network-aware algorithm for service composition. This method considers network features such as network availability and delay during the service composing process. The overall QoS of the service composition factors in network metrics and other non-network QoS parameters such as price and reputation. Next, the QoS constraint to be satisfied by a successful service composition is formulated as a multi-dimensional multi-choice 0–1 knapsack problem (MMKP). This work did not consider user mobility.

The approach proposed in this paper goes beyond existing approaches by modeling the mobility in mobile service composition. Based on the mobility model, we can efficiently compute the mobility-aware QoS of a service composition when the user is moving while invoking services. Additionally, we aimed to find a service composition with an optimal QoS for users in a mobile environment.

7 CONCLUSIONS AND FUTURE WORK
This paper addresses the problem of service selection for mobile composite services. We propose a mobility-enabled approach for service composition in a mobile environment, consisting of a mobility model, a mobility-aware QoS computation rule, and a mobility-enabled selection algorithm based on TLBO. The experimental simulation results demonstrate that our approach can obtain better solutions in a mobile environment than current standard composition methods. Furthermore, our approach achieves near-optimal solutions and has roughly linear algorithmic complexity with regard to the problem size.

In the future, we would like to extend this work to include energy consumption concerns. We will also relax certain conditions in this paper, such as assuming that the user’s path is unknown; in this case, we will need to make a mobility prediction before making a service selection.

ACKNOWLEDGMENTS
This research work was conducted while Shiguang Deng and Longtai Huang were working at MIT as visiting scholars. We thank Prof. Stuart Madnick for his valuable comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant No. 61170033 and the National High-Tech Research and Development Plan of China under Grant No. 2013BAD19B10.

REFERENCES
Shuiguang Deng received his BS and PhD in Computer Science from Zhejiang University in 2002 and 2007, respectively. He is an associate professor in the College of Computer Science at Zhejiang University. He is presently a visiting scholar at MIT. His research interests include Service Computing, Business Process Management and Data Management.

Longtao Huang received his BS degree in Software Engineering from Zhejiang University, Hangzhou, China, in 2010. Currently, he is working towards a PhD in computer science and technology at Zhejiang University. His research interests include service computing and cloud computing. He is a visiting student at MIT since January 2014.

Daning Hu is an Assistant Professor in the Department of Informatics at the University of Zurich and Head of the Business Intelligence Research Group. He received his PhD in Management Information Systems (minor in Finance) from the University of Arizona and BS degree in Computer Science from Zhejiang University.

J. Leon Zhao is Chair Professor and Head of the Department of Information Systems at City University of Hong Kong. He was Interim Head and Eller Professor in MIS at the University of Arizona and taught previously at HKUST and the College of William and Mary. He holds PhD and MS degrees from the Haas School of Business, UC Berkeley, an MS from UC Davis, and a BS from Beijing Institute of Agricultural Mechanization.

Zhaohui Wu received a PhD in computer science from Zhejiang University in 1988. From 1991 to 1993 he was a member of a Sino-Germany jointly trained Ph.D. program. Currently he is a Professor in the College of Computer Science, Zhejiang University.