

## Research Article

# QoS-Aware Mobile Service Selection Algorithm

Chengwen Zhang,<sup>1</sup> Lei Zhang,<sup>2</sup> and Guanhua Zhang<sup>2</sup>

<sup>1</sup>Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts & Telecommunications, Beijing 100876, China

<sup>2</sup>Beijing University of Posts & Telecommunications, Beijing 100876, China

Correspondence should be addressed to Chengwen Zhang; [paperbupt@126.com](mailto:paperbupt@126.com)

Received 6 July 2016; Revised 19 September 2016; Accepted 10 October 2016

Academic Editor: Juan C. Cano

Copyright © 2016 Chengwen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the problem of mobile service selection, this paper gives a context-aware service selection algorithm based on Genetic Algorithm. In this algorithm, a tree encoding method, a fitness function, and a fitness-better strategy were proposed. The tree encoding mode made Genetic Algorithm support selection of various types of service combinations, for example, sequence composition, concurrence composition, probability composition, and loop composition. According to the encoding method, a fitness function was designed specially. The fitness-better strategy gives the direction of population evolution and avoids the degradation of population fitness. Some experiments analyses show that the provided service selection algorithm can get better service composition.

## 1. Introduction

At present, research on service selection is popular [1, 2]. Popularity of service and homogeneity of service functions lead to the same nonfunctional properties. QoS will be more and more important for service composition selection [3]. To provide appropriate services for user, it is necessary to select service based on QoS [4–6].

With quick development of mobile computing, the computing capability can be used as mobile service [7, 8]. Composing service according to user's context information and service's context is important. User's context information includes user's preferences and current location. Service's context includes geographical location and QoS.

Mobile service composition is one of the motive forces of the development of services [9, 10]. A composite service has specific functions which can be divided into several components (component services). Component functions can be, respectively, accomplished by several concrete candidate services. Inevitably, service providers would provide services with same functions and different context. These services can combine tens of thousands of combination plans with the same functions and different context. Therefore, we need to choose from massive services based on user's requirements.

Context-based service selection plays an important role in the area of service combination.

Current service selections are usually based on QoS of services, ignoring the geographical location of services and the location of users. Context information of user, service, and personalized demands are not considered.

When user arrives somewhere, some suitable service compositions are chosen according to user's own preferences, context information of user's current position, context information of services' positions, and QoS.

Therefore, it is necessary to put forward mobile service selection algorithm based on the context of user and service. It introduces context information to the mobile service selection system. And it has higher selection accuracy than the traditional ones.

This paper presents an Improved Genetic Algorithm (IGA) for mobile service composition selection based on context information. Firstly, an improved chromosome coding method is described. Secondly, a fitness function is proposed on the basis of the improved coding method. At last, a fitness-better method is provided to quicken GA's convergence.

Section 2 gives research about mobile service selection computing. IGA is displayed in Section 3. Section 4 is about

simulation. Section 5 points out conclusion and the next step.

## 2. Service Selection Based on Context

QoS is a part of context data. Service selection based on QoS is an attractive search area [11–13]. Meeting global constraints (user's preferences) and finding optimal combination services are QoS-based service combination selection and are in the scope of combinatorial optimization. QoS-based service combination selection is a NP-hard problem [14].

Therefore, approximate algorithm can better select QoS-based service combination. Genetic Algorithm (GA) is a kind of approximate algorithm searching randomly based on probability. It can overcome limitation of exhaustive algorithm and optimize combination [15].

Reference [16] used GA and gained suboptimal solutions. Reference [17] adopted GA and built services combination model by using a one-dimensional chromosome encoding means. Reference [18] used GA to optimize service composition based on QoS. In [19], Girardello and Michahelles selected the most frequent applications after analyzing these mobile applications based on user's current position context. But the paper only considered location context and applications usage, ignoring user's personal preferences. Literature [20] took distance between mobile users and businesses into consideration. It believed that users preferred nearby services. Distance between users and services might have big influence on users.

Service composition based on the context information has many types, such as serial type, parallel type, probability type, and circle type. So, an appropriate encoding of GA needs to be designed to satisfy various types of service compositions. Stochastic calculation of GA ensures that GA has the ability to search all the solution space, but it also brings the problem of slow convergence. Some methods should be adopted to speed up convergence of GA.

## 3. Improved Genetic Algorithm

We provide an Improved Genetic Algorithm (IGA) for mobile service composition selection. It includes a tree ergodic encoding, a fitness function, and a fitness-better strategy.

**3.1. Tree Encoding Method.** This section shows a tree encoding method. The encoding is from a services combination tree stencil.

**3.1.1. Tree Stencil of Services Combination.** Some basic service composition types, such as probability composition, series composition, parallel composition, and circulation composition are included in mobile service combination process.

A special tree structure can present the service combination process. It is called tree combination stencil. An entire service combination process, the logical relationship among components, can be presented with root node, nonleaf nodes, and leaf nodes in this tree combination stencil. Figure 1 describes a service combination.

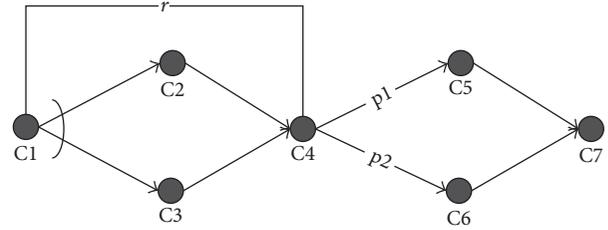


FIGURE 1: An example about service combination.

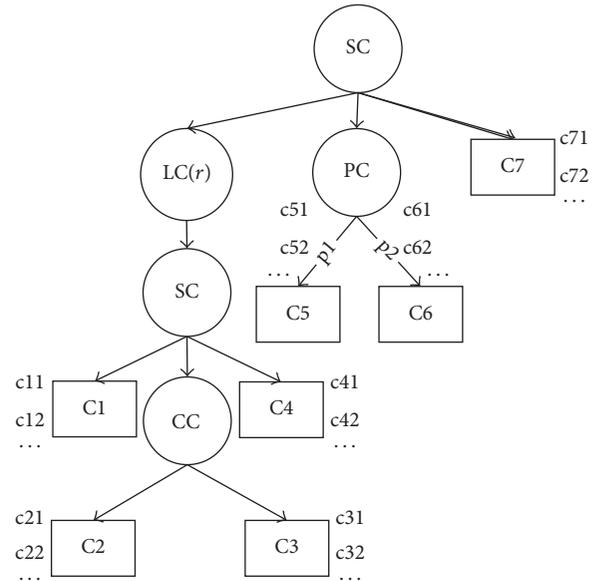


FIGURE 2: Tree combination stencil of Figure 1.

Figure 1 is the logical plan of a service composition example. It is built according to logical functions of service components. There are four service combination types (sequence composition, concurrence composition, probability composition, and loop composition) in Figure 1.

In Figure 1, C1~C7 are seven service components. C1~C4 are loop composition and  $r$  is repeat number. C2~C3 are concurrence composition. C5~C6 are probability composition.  $p1$  and  $p2$  are probability value and the sum of them is 1.

Figure 2 is the tree combination stencil of Figure 1.

In Figure 2, circle icons are nonleaf nodes; composition type and structure parameters are saved there. Rectangle icons are leaf nodes; service components and logical function definitions are there. Nodes CC, LC, PC, and SC are, respectively, concurrence composition, loop composition, probability composition, and series composition. They represent composition type of their subtree. Nodes C1~C7 are the same as those in Figure 1. They are service components of service combination. Each service component has a number of candidate mobile services. For example,  $c11$  and  $c12$  are candidate mobile services of C1, and specific context values (QoS values and geographical positions) are saved in  $c11$  and  $c12$ .

**3.1.2. Tree Ergodic Encoding Method.** The encoding design of GA has a direct impact on capacity and efficiency of GA. A simple GA has demonstrated its inadequacies in its encoding while it is used to solve selection of mobile service. Some literatures put forward their own improvement methods.

The following is the chromosome encoding process where the tree ergodic encoding method is used.

The first step is to initialize a pointer that points to the first gene locus of a chromosome on the left.

The second one is to traverse tree combination stencil using preorder ergodic based on the order from left to right.

The third one is that when traversing to a nonleaf node, the composition type values and structure parameters will be used to compute.

The fourth one is that when traversing to a leaf node, the service component represented by the leaf node will be put into the gene locus that the current pointer points to. Then the pointer will be shifted to the right gene locus.

The last one is to repeat the above steps until the tree stencil traversing is finished.

Through the above steps, chromosome can be obtained from the tree combination stencil. The following are the chromosomes of Figure 2:

(C1, C2 + C3, C4)<sup>r</sup>, C5, and C7

(C1, C2 + C3, C4)<sup>r</sup>, C6, and C7

C2 + C3 simultaneously runs C2 and C3. The exponent  $r$  is loop number of C1~C4. Each gene indicates a service composition.

After the two chromosomes are built, some individuals (individuals are instances of service combination) can be generated. Many candidate services for each component make a great deal of service combination instances (instances are individuals in GA).

Some individuals are selected to form an initial population of GA. User's personal preferences will be the global constraints of service composition. Through some genetic computing, next successive generations of population are obtained until getting a service combination solution.

**3.2. Fitness Calculation.** In GA, there are a lot of individual instances. Every individual is an instance of service combination. The fitness value of individual will impact on the probability of being selected. The fitness function is a core technology of GA. The user's personal preferences is the global QoS constraints during selecting individuals. The global QoS constraints are parts of the fitness function. The following is about how to build the fitness function.

**3.2.1. Geography Position Data Process.** Firstly, according to geography distance limits provided by user, some incongruent candidate services will be cancelled. If user has special geography path requirements, some incongruent candidate services will also be quitted. But the algorithm to compute position data does not belong to the paper.

**3.2.2. Fitness Function.** A selection rule and user's personal preferences need to be considered when building a fitness

function. Here, the design rule of the fitness function is that the fitness value of individual is bigger and the selected probability of the individual is bigger.

Formula (1) is a fitness function, and it has two parts divided by a minus sign. The front part is an objective function. It integrates all QoS attributes. The rest is a penalty technique for constrained optimization problems. It selects proportionally the individual disobeying constraints into the next generation population. It is about user's personal preferences.

$$\text{Fitness} = \sum_{j=1}^N (W_j \times Q'_j) - \beta \sum_{k=1}^M (W_k \times \Delta Q_k). \quad (1)$$

In formula (1),  $N$  is the number of QoS properties.  $M$  is the number of QoS constraints.  $\beta$  is a coefficient and an experience value.  $W_j$  and  $W_k$  are the weight values of relative QoS attributes and are importance of relative QoS properties.  $Q'_j$  is a part of the selection rule that integrates all QoS attribute values.  $\Delta Q_k$  is about user's personal preferences or global constraints. The minus sign in formula (1) is penalty function method that can offer survival probability for those individuals that violate user's personal preferences.

There are a lot of kinds of QoS whose measurement magnitudes are different. If these different QoS attributes are in a mathematical fitness function, their difference needs to be eliminated to guarantee their fair impact on services choice. Moreover, QoS attributes are other two kinds: one is positive-kind (the bigger the QoS value is, the better the impact is, such as reputation), and another one is negative-kind (the less the QoS value is, the better the impact is, such as price). Because there are positive-kind QoS and negative-kind one at the same time in the mathematical fitness function, it needs to employ some methods to synthesize the two opposite types. Because of the above various situations, the formula of  $Q'_j$  is built as

$$Q'_j = \begin{cases} \frac{Q_j - Q_{j\text{Min}}}{Q_{j\text{Mean}}} & Q_j \text{ is positive-kind} \\ \frac{Q_{j\text{Max}} - Q_j}{Q_{j\text{Mean}}} & Q_j \text{ is negative-kind.} \end{cases} \quad (2)$$

In formula (2),  $Q_{j\text{Min}}$ ,  $Q_{j\text{Mean}}$ , and  $Q_{j\text{Max}}$  are, respectively, minimum, mean value, and maximum of the  $j$ th QoS attribute among all of service composition instances. The division in formula (2) is to balance different magnitude of QoS attributes.

When the  $k$ th QoS value conforms to the relative QoS limit, the value of  $\Delta Q_k$  is 0. Or, formula (3) is used for  $\Delta Q_k$ .

$$\Delta Q_k = \begin{cases} \frac{Q_k - C_{k\text{Max}}}{C_{k\text{Max}}} & \text{if } Q_k > C_{k\text{Max}} \\ \frac{C_{k\text{Min}} - Q_k}{C_{k\text{Min}}} & \text{if } Q_k < C_{k\text{Min}}. \end{cases} \quad (3)$$

In formula (3),  $C_{k\text{Min}}$  and  $C_{k\text{Max}}$  are minimum and maximum in the  $k$ th restrictive condition, respectively.  $Q_k$  is the  $k$ th QoS value.

When using the tree encoding method, its calculation process for individual fitness value is as follows.

The first step is to initialize a pointer that points to the first gene locus from left to right in the chromosome.

The second is to traverse tree combination stencil according to preorder ergodic order.

The third is the represented component being replaced by an atom service that the current pointer points to at gene locus when a leaf node is traversed. Then the pointer will be shifted to the next gene locus.

The fourth one is that when traversing to a nonleaf node, if all its child leaf nodes have been replaced with the specific atomic services and include the QoS properties vectors of themselves, and if its child nonleaf nodes have stored integration value vectors of QoS properties of their child nodes, the current nonleaf node will be able to use QoS properties vectors contained in its child nodes as well as its stored combination logic information to calculate integrated value of QoS attributes. Finally, the calculated integration values of QoS properties will be stored in the nonleaf node.

The last is to repeat the above steps until all nodes of tree stencil are traversed.

Through the above steps, the global values of all QoS properties of an individual are obtained and stored in the root node of the tree combination stencil. On this basis, the fitness values of all individuals can be calculated using the fitness function.

**3.3. Fitness-Better Strategy.** GA is a random search algorithm. Random calculation enables GA to search all solution areas. But random search in the scope of massive solutions causes slow convergence situation. It is difficult for GA to obtain an optimal solution. Thus, some methods should be used to reduce the random running of GA. It is a good way to direct GA using some evolutionary means.

In GA, crossover and mutation are two evolution operations generating new individuals. It is possible that the result(s) of fitness of these two operations is (are) worse than their parent chromosome(s). To avoid the above degeneration situation, a fitness-better method is provided and it can direct every operation. The following is the fitness-better way. Meanwhile, to avoid local optimum, the individual with the best fitness can survive with the highest possibility. During every evolution calculation period, the best fitness will be obtained.

(1) During mutation calculation period of each generation, the calculation process is the following. Before the first mutation operation, the fitness value of mother-chromosome is saved as the best fitness value of mutation (its abbreviation is BestFM). For each mutation of every gene locus in the chromosome, the fitness of mother-chromosome and its offspring chromosomes is compared and a better fitness (BetterFM) is obtained. Then, BetterFM is compared with the old BestFM. The better fitness between BetterFM and old BestFM is saved as new BestFM. After mutation operations of all gene loci in the mother-chromosome, the newest BestFM is the best fitness value among these mutation operations. The second mutation operation will use the BestFM as its initial BestFM, and so on. After all of mutation calculations,

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| c11 | c21 | c31 | c41 | c51 | c61 | c71 |
| c12 | c22 | c32 | c42 | c52 | c62 | c72 |
| ... | ... | ... | ... | ... | ... | ... |
| C1  | C2  | C3  | C4  | C5  | C6  | C7  |

FIGURE 3: One-dimensional encoding means.

the BestFM will become the best fitness during this mutation calculation period. The relative chromosome will enter into the next generation.

(2) During crossover calculation period of each generation, the calculation process is the following. Before the first crossover operation, the two mother chromosomes are saved as two best fitness values of crossover (their abbreviations are BestFC-1 and BestFC-2).

After each crossover operation, the fitness values of parent chromosomes and their offspring chromosomes are compared to one another. The best two fitness values will be BetterFC-1 and BetterFC-2. The top two fitness values among BetterFC-1, BetterFC-2, BestFC-1, and BestFC-2 will be new BestFC-1 and BestFC-2 to replace the old ones. After all of crossover calculations during this crossover period, the BestFC-1 and BestFC-2 will become the best two fitness values during this crossover calculation period. The relative chromosomes will enter into the next generation.

Through the above ways, the individuals with better fitness can be got. The new generation can have better fitness than old ones. Thus, population fitness will be improved. To some extent, the convergence speed can be quickened.

## 4. Experiments Analyses

In this paper, an improved GA is proposed. It includes the tree encoding method, the fitness function, and the fitness-better method. Through the above improvements, faster convergence speed and better search ability will be possible. The following are some experiments and analysis which show that IGA has better efficiency and capacity.

For mobile service composition selection, some experiments between Simple Genetic Algorithm (SGA) and IGA were set in order to testify the effect of IGA.

IGA used the tree coding, fitness function like formula (1), and the fitness-better strategy.

And, SGA adopted one-dimensional encoding means in Figure 3 and fitness function like formula (4).

In Figure 3, C1~C7 and c11, c12, . . . are the same as those in Figure 2. The one-dimensional encoding means that it does not have the ability to represent many service composition scenarios, such as probabilistic invocation, parallel invocation, sequential activation.

$$\text{Fitness} = \sum_{j=1}^N (W_j \times Q'_j). \quad (4)$$

Formula (4) used absolute rejection technology to reject the individual whose QoS violates the constraints asserted by user.

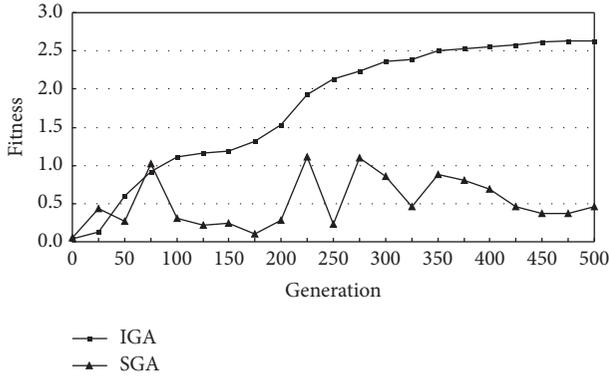


FIGURE 4: Convergence speed comparison.

Same parameters were for both algorithms, including same population size, a single-point mutation and mutation probability 0.1, and a single-point crossover and crossover probability 0.7.

The encoding means, fitness function, and the evolution policy are the different points among the compared GAs.

The two compared algorithms confronted the mobile service selection with the same combination data, including the same number of components and same QoS attributes for every candidate service. All QoS values and global limits for some QoS attributes were randomly made.

The same data include 10–30 candidate services for each service component and 6 QoS data for each service. A simplified representation of service includes ID number and some QoS data.

**4.1. Experiment Analyses about Convergence.** Convergence rate is related to generation number where GA obtains a final solution. The less the halt generation number is, the quicker the convergence rate is.

In order to testify whether IGA decreases the halt generation number, SGA and IGA were run for 100 times, respectively. Some result data are displayed in Figure 4.

From Figure 4, SGA was not convergent before 500 generations and IGA was stable before about 450 generations. The random search method of SGA is the reason. In contrast, IGA improves convergence rate through the above fitness-better ways that reduce the randomness of GA. IGA gets the final result of mobile service selection in a shorter period of time and has faster convergence speed than SGA. Moreover, IGA can obtain higher fitness than SGA. The reason is that individuals with better fitness can survive during crossover and mutation operation.

So, one of the contributions of IGA is that it can rapidly and steadily select the optimal composition plan from a great deal of candidate plans.

**4.2. Experiments Analysis about Search.** Search capability is the better solution that can be found. It can be measured through the fitness value of the final selected individual. On the basis of the above fitness formulas, the fitness value is proportionate to search ability of GA.

IGA adopted the tree coding way and the special fitness function as well as the fitness-better means to improve its

TABLE 1: Maximum fitness value (SGA : IGA).

| Component number | Maximum fitness value |
|------------------|-----------------------|
| 10               | 1.16 : 2.63           |
| 25               | 0.93 : 1.85           |
| 30               | 0.36 : 0.86           |

search ability. In order to test them, SGA and IGA both run for 100 times. Some maximum values of the final individuals were saved. Table 1 is part of experiment data.

In Table 1, IGA obtains larger final fitness value than SGA. That is, IGA may get much better solutions than SGA. It is shown that IGA has better search capability than SGA. Using the fitness-better methods, the individuals with better fitness will have bigger opportunity to enter into the next generation. The individuals with lower fitness have less opportunity to take part in the next generation evolution. In comparison, the individuals with higher fitness value are unsure about surviving in SGA. It is difficult for SGA to get better and better fitness.

## 5. Conclusion

It is inevitable that a large number of candidate services with the same function and different nonfunctional attributes (mainly referring to QoS attributes) exist. How to quickly select a mobile service composition from massive candidate ones to meet user's requirements has become urgent.

This paper presents a mobile services selection algorithm based on GA. The improvement of GA includes a tree encoding method and a fitness calculation way based on the new encoding method. In order to compensate for the lack of evolution direction of GA, a fitness-better way is used during crossover and mutation operations. In the result, individuals with better fitness can be selected and convergence rate can be raised.

Through the above-mentioned experiments, capacity and efficiency of IGA were confirmed. It is a good solution for mobile services composition selection.

One deficiency of GA is its local convergence capability. It is necessary to combine GA and a local search algorithm that can compensate for GA's local search ability [21]. Therefore, the next study will present a hybrid algorithm of GA and a local optimization algorithm.

## Competing Interests

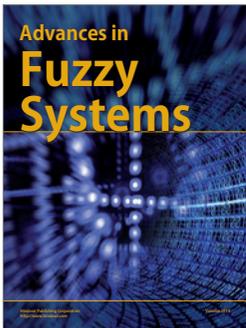
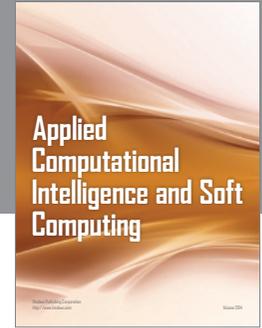
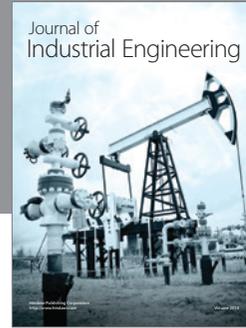
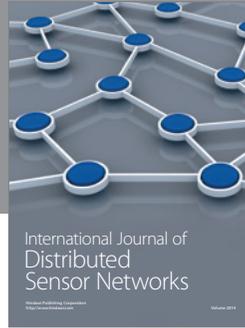
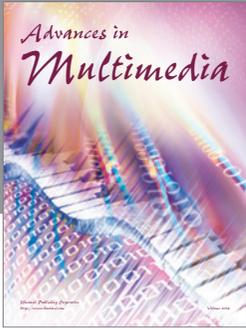
The authors declare that there are no competing interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61132001); National 973 Programs (no. 2013CB329102); and Program of National High-tech R&D Program of China (863 Program) under Grant no. 2013AA102301.

## References

- [1] S. Soltani, P. Martin, and K. Elgazzar, "QuARAM recommender: case-based reasoning for IaaS service selection," in *Proceedings of the International Conference on Cloud and Autonomic Computing (ICCAC '14)*, pp. 220–226, London, UK, September 2014.
- [2] X. G. Wang, J. Cao, and Y. Xiang, "Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing," *Journal of Systems and Software*, vol. 100, pp. 195–210, 2015.
- [3] M. Rajeswari, G. Sambasivam, N. Balaji, M. Saleem Basha, T. Vengattaraman, and P. Dhavachelvan, "Appraisal and analysis on various web service composition approaches based on QoS factors," *Journal of King Saud University—Computer and Information Sciences*, vol. 26, no. 1, pp. 143–152, 2014.
- [4] L. Wu, Y.-L. Liu, and S.-J. Liu, "Multi-strategy QoS-aware service selection and ranking model," *Tongxin Xuebao/Journal on Communications*, vol. 33, no. 1, pp. 141–147, 2012.
- [5] H. Y. Zheng, J. Yang, W. L. Zhao et al., "QoS analysis for web service compositions based on probabilistic QoS," in *Proceedings of the 9th International Conference on Service Oriented Computing*, pp. 47–61, Paphos, Cyprus, December 2011.
- [6] L. Chen, J. Yang, and L. Zhang, "Time based QoS modeling and prediction for web services," in *Service-Oriented Computing: 9th International Conference, ICSOC 2011, Paphos, Cyprus, December 5–8, 2011 Proceedings*, vol. 7084 of *Lecture Notes in Computer Science*, pp. 532–540, Springer, Berlin, Germany, 2011.
- [7] S. D. Xue, B. D. Wu, B. Cheng, X. Qiao, and J. Chen, "Mobile business process personalization for end users," *China Communications*, vol. 12, no. 12, pp. 1–12, 2015.
- [8] S. S. Tu and Y. F. Huang, "Towards efficient and secure access control system for mobile cloud computing," *China Communications*, vol. 12, no. 12, pp. 43–52, 2015.
- [9] G. A. G. Llinas and R. Nagi, "Network and QoS-based selection of complementary services," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 79–91, 2015.
- [10] X. K. Wu, Y. Tian, J. Y. Wu, B. Cheng, and J. Chen, "A composite service provision method based on novel node model in mobile ad hoc networks," *China Communications*, vol. 11, no. 4, pp. 130–142, 2014.
- [11] Z. U. Rehman, O. K. Hussain, and F. K. Hussain, "Parallel cloud service selection and ranking based on QoS history," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 820–852, 2014.
- [12] Q. Wu, Z. Li, Y. Yin, and H. Zeng, "Adaptive service selection method in mobile cloud computing," *China Communications*, vol. 9, no. 12, pp. 46–55, 2012.
- [13] Q. B. Sun, W. B. Wang, and H. Zou, "A service selection approach based on availability-aware in mobile ad hoc networks," *China Communications*, vol. 8, no. 1, pp. 87–94, 2011.
- [14] G. Canfora, M. D. Penta, R. Esposito et al., "A lightweight approach for QoS-aware service composition," in *Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC '04)*, pp. 36–47, New York, NY, USA, 2004.
- [15] M. Srinivas and L. M. Patnaik, "Genetic algorithms: a survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994.
- [16] T. Yu, Y. Zhang, and K. J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Transactions on the Web*, vol. 1, no. 1, pp. 1–26, 2007.
- [17] G. Canfora, M. Dipenta, and R. Esposito, "An approach for QoS-aware service composition based on genetic algorithms," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 1069–1075, Washington, DC, USA, 2005.
- [18] C. W. Zhang, J. Kuang, B. Cheng, L. C. Zhang, and Z. Z. Zhang, "Genetic algorithm with dynamic stopping criteria for global QoS-driven web service replanning selection," *Information*, vol. 15, no. 2, pp. 689–698, 2012.
- [19] A. Girardello and F. Michahelles, "AppAware: which mobile applications are hot," in *Proceedings of the Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*, pp. 2010–431, ACM Press, New York, NY, USA, 2010.
- [20] W.-S. Yang, H.-C. Cheng, and J.-B. Dia, "A location-aware recommender system for mobile shopping environments," *Expert Systems with Applications*, vol. 34, no. 1, pp. 437–445, 2008.
- [21] S. X. Sun and J. Zhao, "A decomposition-based approach for service composition with global QoS guarantees," *Information Sciences*, vol. 199, pp. 138–153, 2012.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

