


RESEARCH

Open Access



A hybrid service selection optimization algorithm in internet of things

Xiaofei Zhang^{*} , Juncheng Geng, Jianwei Ma, Hao Liu, Shuangxia Niu and Wandeng Mao

^{*}Correspondence:
zzdw_zxf@tom.com
Electric Power Research
Institute of State Grid Henan
Electric Power Company,
Zhengzhou 450052, China

Abstract

With the scale of Internet of Things (IoT) continues to increase, it brings big challenges for service selection in a large-scale IoT. For solving this problem, a service selection method based on the enhanced genetic algorithm is proposed in this paper. To decrease the scale of service selection, this paper uses the lexicographic optimization approach and quality of service (QoS) constraint relaxation technique to find the candidate service with height QoS. Then, the IoT service selection problem is transformed into a single-objective optimization problem adopting a simple weighting method, and the final composite service meeting the user's QoS needs are obtained from the candidate service. The simulation results show that the proposed algorithm can efficiently and quickly achieve a composite service satisfying user's QoS needs, and is more suitable for solving the service composite problem in large-scale IoT services.

Keywords: Internet of things, Hybrid optimization algorithm, Service selection, Quality of service

1 Introduction

Internet of Things is an emerging technology, which is widely used in many fields such as smart homes, intelligent logistics, and so on. It also changes people's lifestyles and promotes a series of industrial revolutions. The essence of IoT is allowing smart components, such as smartphones, with certain computing and storage capabilities, to interconnect. Meanwhile, these smart devices can communicate, interact, and cooperate with other components to generate new smart services [1, 2]. Moreover, with the popularization of IoT technology, the number of smart devices deployed on IoT is increasing exponentially every year [3]. In IoT, each smart device provides its own functional services, which are called and provided to users through standard web services. Therefore, service-oriented computing (SoC) and service-oriented architecture (SoA) are regarded as the key to the development of IoT technology.

At present, due to the changing of user needs, a single service, which provided by smart devices, cannot meet the user's needs in IoT. IoT services composition based on SoC and SoA has successfully solved this problem. Compared with traditional service selection, the characteristic of today's IoT service selection is that all IoT services run directly in smart devices with limited power, storage capacity, and computing resources [4]. Therefore, the non-functional attributes of IoT composite services are crucial factors,

which will directly affect the user experience. However, as the scale of the Internet of Things continues to increase, there are many services with the same function and different QoS. How to quickly integrate these services into comprehensive composite services and applications satisfying user QoS requirements has become a major challenge.

In order to solve the service composition problem concerning QoS needs in the large-scale IoT, we propose a hybrid service selection method based on the enhanced genetic algorithm in this paper. This algorithm mainly consists of two parts: First, for reducing the scale of service selection, the hybrid service selection problem is modeled as a multi-attribute decision-making problem, and we use the lexicographic optimization method and QoS constraint relaxation technology to select candidate services with higher QoS within users' QoS preferences [5, 6]. Finally, the simple linear weighting method [7] is used to convert the service composite problem into a single-objective optimization problem, and the enhanced genetic algorithm [8] is used to search the service selection with the greatest fitness from the candidate services.

Simulation experiments are carried with matlab2014a on a 64-bit Windows operating system. We compared our proposed EGS~QoS algorithm with related algorithms. The simulation results indicate that the proposed algorithm performs better. It can efficiently and accurately find the optimal composite service with higher QoS.

The remaining parts are organized as follows. Section 2 summarizes related work for IoT service selection. Section 3 introduces the model of IoT service. Section 4 details the proposed algorithm. Section 5 presents the simulation results and evaluation. Section 6 gives conclusions in this work.

2 Related work

Now, SOA and SOC are some of the most commonly used structures to solve service selection. Based on this structure, the multiple services with similar functions and different QoS can be combined into a complex and comprehensive composite service within QoS constraints. Thus, the essence of hybrid service selection problem is NP-hard problem. Many scholars have devoted themselves to solve this problem in the past.

Kouicem et al. [9] and Yachir et al. [7] used the simple additive weighting (SAW) technique to calculate the QoS utility of the service, then select the service with the largest utility value as the final composite service. L. Zeng Et al. [11] used integer linear programming (ILP) to solve the problem of service selection successfully, then proposed a two-step QoS-driven services composition method. First, regardless of global QoS constraints, a local selection strategy was implemented for each service class, and candidate services that met the requirements were selected. Then, the ILP method was used to select the final composite service form candidate service for each service class based on the global QoS constraints and user's preferences. Y. Ngoko et al. [12] applied a mixed-integer linear programming method to solve service selection problems and took energy consumption as a QoS attribute to optimize the process of composition. However, as the problem scale increased, the efficiency of algorithm selection would decrease, which was not suitable for large-scale service composite problems. X. Jin et al. [13] adopted the concept of Pareto optimality to deal with the problem of service selection and proposed a service model with four QoS attributes to describe the characteristics of IoT services.

Through a single QoS rating function (fitness function), the candidate service was evaluated and the composition service that met the requirements was selected.

Different from the available studies, we combined the lexicographic optimization method [14] and threshold relaxation technology [15] to preselect candidate services with higher QoS, and then use the genetic algorithm to search for the final combination services from the candidate services according to fitness value.

2.1 Models description and optimization goal

In the process of service selection for IoT, there are mainly two types of services: the concrete services and the abstract service classes. The abstract service class, also known as abstract service, is defined as a service function that includes multiple concrete services with the same function and different QoS. The concrete services, also known as atomic services, are provided by the smart device in IoT [16]. Figure 1 is the model of hybrid service selection. As shown in Fig. 1, the hybrid service selection has two parts: First, integrate the existing service classes into a new comprehensive class. Second, find the optimal concrete service form each abstract service to composite service for new service class [17].

In this paper, the concrete service Cs_{ij} , also known as invocable service, is provided by IoT components and can be called through standard web services.

$$Cs_{ij} = \{QoSA(Cs_{ij}), Action(Cs_{ij})\} \tag{1}$$

In which, $QoSA(Cs_{ij})$ is the QoS attribute of the concrete service Cs_{ij} , and $Action(Cs_{ij})$ represents the function of the concrete service Cs_{ij} .

The abstract service class As_i consists of m concrete services which have the same functions but different QoS levels, it can be expressed as follows.

$$As_i = \{Cs_{i1}, Cs_{i2}, \dots, Cs_{ij}\} \quad \text{and } i \in [1, n], j \in [1, m] \tag{2}$$

In the IoT environment, the QoS attributes of services are almost the same as those of web services. The QoS attributes usually have two types: qualitative attributes and quantitative attributes. Qualitative attributes include privacy, security, and so on. quantitative attributes include cost, response time, reliability, and reputation [18, 19].

According to the impact of QoS attributes on services, the attribute is divided into the positive attribute and the negative attribute. The positive attribute means that, as the

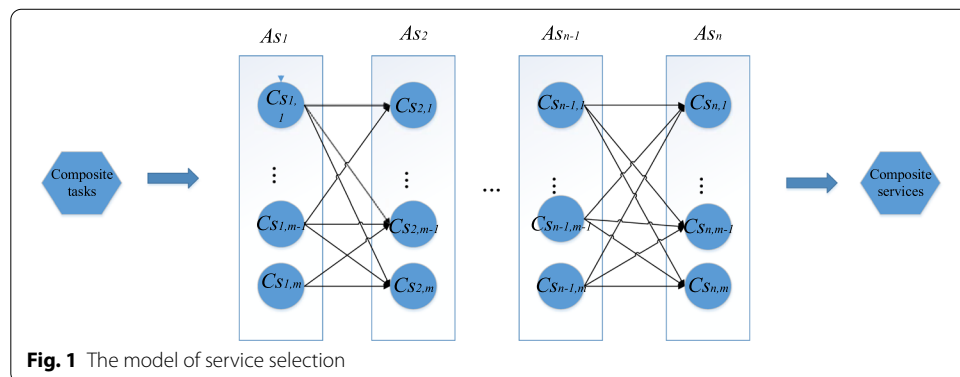


Fig. 1 The model of service selection

value of the QoS attribute increasing, the service QoS increases [20]. The negative attribute means that, with the value of the QoS attribute growing, the service QoS decreases.

The QoS attributes of the concrete service Cs_{ij} are as follows:

$$QoSA(Cs_{ij}) = \{qos_1^{ij}, qos_2^{ij}, \dots, qos_t^{ij}\} \quad (3)$$

In which, qos_t^{ij} represents the t -th QoS attribute in the concrete service Cs_{ij} .

The QoS model of the concrete Cs_{ij} is as follows:

$$\begin{cases} QaggregationP(Cs_{ij}) = \sum_{l=1}^{t_1} w_l qos_l^{ij}, & \text{if } qos_t^{ij} \text{ is postive} \\ QaggregationN(Cs_{ij}) = \sum_{l=1}^{t_2} w_l qos_l^{ij} & \text{otherwise} \\ \sum_{l=1}^t w_l = 1, \text{ and } t_1 + t_2 = t \end{cases} \quad (4)$$

In which, $QaggregationP(Cs_{ij})$ and $QaggregationN(Cs_{ij})$ are the QoS aggregation values for the positive and negative attributes in the specific service Cs_{ij} , respectively, and w_l represents the weight of the user's preference for each QoS attributes [21].

$$Qaggregation(Cs_{ij}) = \frac{1}{QaggregationN(Cs_{ij})} + QaggregationP(Cs_{ij}) \quad (5)$$

In which, $Qaggregation(Cs_{ij})$ is the QoS aggregation value for the concrete service Cs_{ij} . The Cs_{ij} with larger the QoS aggregation value shows that Cs_{ij} with the upper QoS level of the service in abstract service class As_i .

The QoS of the composite service in IoT is affected by its own structure. The commonly used service composite structures are sequential, parallel, conditional, and loop, respectively. All composite services can be combined with the above four basic structures. Since any complex structure can be transformed into a sequential structure through a certain technology, so the composite service in this paper adopts a sequential structure [22, 23].

In this work, the service selection problem is finally transformed into a single-objective optimization problem, and the enhanced genetic algorithm is used to find a composite service meeting the user's QoS requirements. In the composition process, we will choose the final composite service based on the fitness of the composition service. The optimization goal in this paper is as follows.

$$Maximizing \rightarrow ComFitness(x) = \sum_{i=1}^n Qaggregation(x) \quad (6)$$

In which, $ComFitness(x)$ is the fitness value of the composite service, n is the size of the composite service, and x represents the concrete service Cs_{ij} contained in the composite.

3 The proposed algorithm

Here, we propose a hybrid service selection method based on the enhanced Genetic algorithm (EGS-QoS). The EGS-QoS algorithm consists of two steps. First, lexicographic optimization and threshold relaxation technology are used to preselect candidate services with higher QoS. Second, final composition service is selected based on enhanced genetic algorithm.

The lexicographic optimization method is a classic multi-attribute decision-making technique, according to the importance of attributes and constraints, which can sequentially screen out solutions that meet the requirements. Threshold relaxation technology is a variable threshold calculation method. According to constraints, user preferences and relaxation coefficients, the threshold range of variable can be calculated.

Composition service is a NP-hard problem. Bio-inspired algorithm is suitable for solving this kind of problem. Genetic algorithm is one of the most successful bio-inspired algorithms and successful used in many applications. Hence, we use an enhanced genetic algorithm to select final composition service.

3.1 Service preselection concerning QoS levels and user’s QoS needs

In this paper, we model the service preselection concerning QoS levels and user’s QoS needs as a multi-attribute decision-making problem, then use lexicographic optimization technology and threshold relaxation technique to preselect candidates that meet the needs. The traditional lexicographic optimization technology is used to solve multiple subproblems ranking according to the importance of its objective problem.

In this paper, we innovatively regard the aggregate value of the IoT service QoS, the positive QoS or the negative QoS for each abstract service class as the sub-optimization problem to be solved in the lexicographic optimization technology. Therefore, the service pre-screening steps are as follows:

- Step 1* Ranking QoS aggregation value attributes and the positive or the negative QoS aggregation value according to the sum of user’s preference weights for QoS.
- Step 2* For each abstract service class As_i with m concrete service Cs_{ij} , each QoS aggregation value is processed, whether negative or positive, according to the previous order.
- Step 2.1* Discover the optimal QoS aggregation value $Best_Q$ in the abstract service class As_i .
- Step 2.2* Using threshold relaxation technology [6], calculate the threshold of each QoS attribute aggregation value, based on the optimal QoS aggregation value $Best_Q$ and tolerance factor δ_{QoS} .

$$QoS_{Aggeration}^{Threshold} = \begin{cases} BestQ - QoS_{Aggeration}^{Decrease}, & \text{if } QoS_{Aggregate}^{Value} \text{ is positive} \\ BestQ + QoS_{Aggeration}^{Decrease}, & \end{cases} \tag{7}$$

$$\text{with } QoS_{Aggeration}^{Decrease} = BestQ * \delta_{QoS} \tag{8}$$

- Step 3* The concrete service Cs_{ij} that is not within the threshold $QoS_{Aggeration}^{Threshold}$ will be deleted from the candidate service sets of the abstract service class As_i . This reduces the composite scale and search time to a certain extent.

The pseudocode of the service preselection stage based on lexicographic optimization

| |
|---|
| <p>Algorithm 1— service preselection concerning QoS levels and user’s QoS needs</p> <p>Input : abstract service class As_i containing m concrete service.</p> <p>δ_{QoS}: the tolerance factor for QoS aggregation value.</p> <p>Step 1: the QoS aggregation values, positive or negative, are ranked according to the user’s preference weight for QoS Attribute. The rank is 1 or 2. The rank of QoS aggregation values with 1 has the highest priority.</p> <p>Step 2: $CAS=As_i$</p> <p>Step 3: for rank=1 to 2 do</p> <p>Step 4: discover $Best_Q$ from CAS</p> <p>Step 5: compute $QoS_{Aggregation}^{Decrease}$ according to (8)</p> <p>Step 6: compute $QoS_{Aggregation}^{Threshold}$ according to (7)</p> <p>Step 7: Update CAS as follows:</p> <p>Step 8: if current QoS aggregation value is positive then</p> <p>Step 9 : $CAS=\{Cs_{ij} \in CAS \cap QaggregationP(Cs_{ij}) \geq QoS_{Aggregation}^{Threshold} \}$</p> <p>Step 10 : else</p> <p>Step 11 : $CAS=\{Cs_{ij} \in CAS \cap QaggregationN(Cs_{ij}) \leq QoS_{Aggregation}^{Threshold} \}$</p> <p>Step 12 : end if</p> <p>Step 13 : end for</p> <p>Outputs : the pre-selecting candidate service sets CAS for each abstract service class As_i.</p> |
|---|

and threshold relaxation technology is shown in Algorithm 1.

3.2 Final composition service selection based on enhanced genetic algorithm

In this paper, concrete services with relatively low QoS are deleted through service preselection for each abstract service class, which reduces the search domain to a certain extent. The service selection in IoT is an NP-hard problem. The genetic algorithm is a common method to solve this problem. To speed up efficiency, we use an enhanced genetic algorithm to find the final composite service. The enhanced genetic algorithm adopts some novel operators, such as population selection operation, individual coding method, individual crossover and mutation operators.

The flow chart of the enhanced genetic algorithm is shown in Fig. 2.

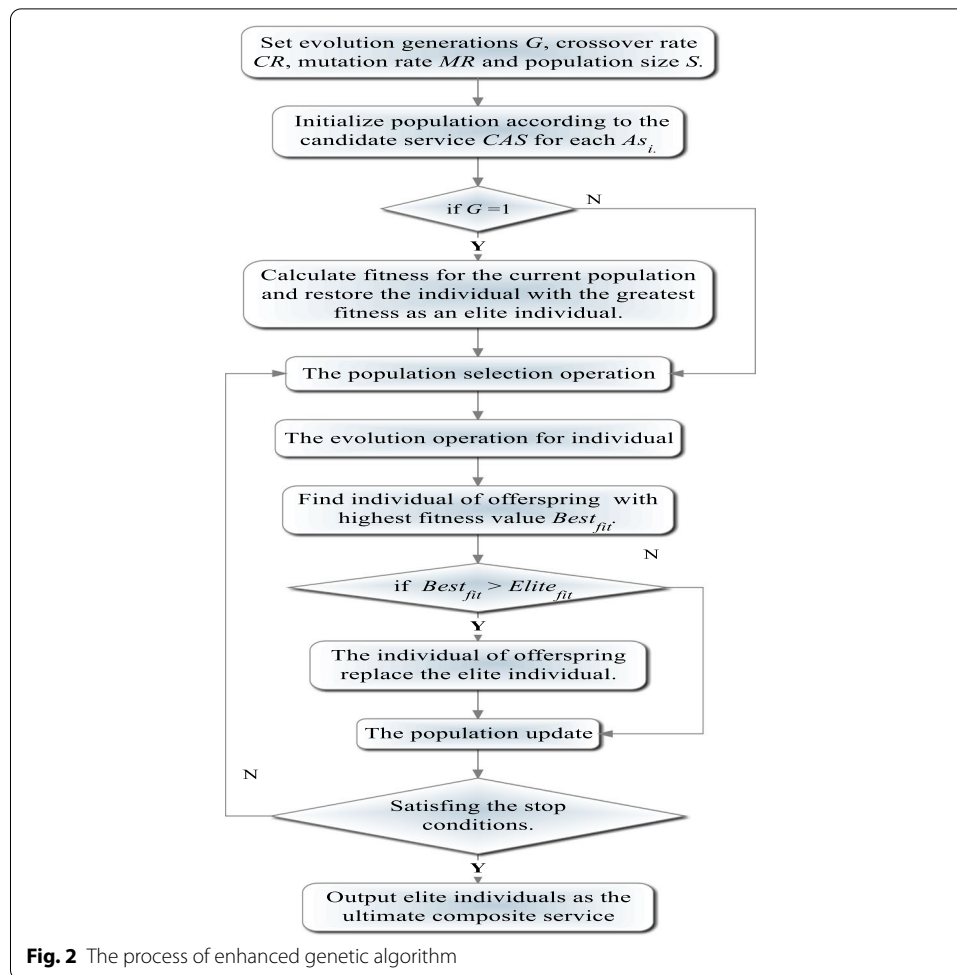
For the enhanced genetic algorithm, the current population is generated according to the number of concrete services contained in the candidate service set of each abstract class. In order to ensure the complete evolution of the population, the evolution iterations change adaptively according to the scale of the composite service, as shown below.

$$Gertaions = \frac{n*m}{10} \tag{9}$$

In which, n is the total number of abstract service classes, and m is the number of concrete services for abstract service classes.

3.2.1 The population selection operation

Commonly used methods of population selection are roulette wheel selection [17], stochastic universal sampling [23], and tournament selection method [24]. To ensure the



population diversity in the search process, this paper adopts the tournament selection method for population selection operations.

3.2.2 The individual coding method in evolutionary operation

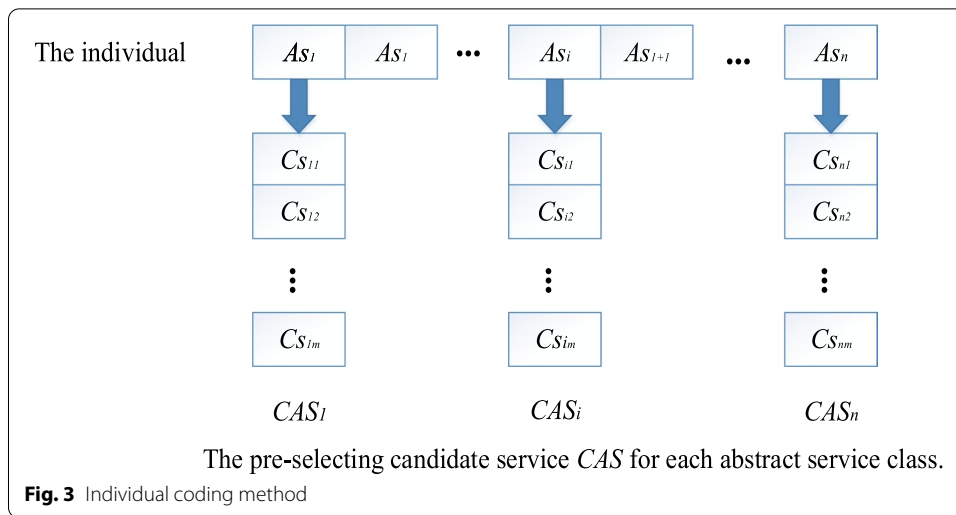
In this paper, the coding method of population individuals adopts integer coding. As shown in Fig. 3, the individual length is the total number n of abstract service classes participating in service selection. The i -th locus of an individual is the serial number of the concrete service included in the preselected candidate service sets CAS for the abstract service class As_i .

3.2.3 The individual crossover method in evolutionary operation

To speed up the evolution, we use the single-point crossover to perform the individual crossover operation.

3.2.4 The individual mutation method in evolutionary operation

For ensuring the integrity of the individual, we use the small disturbance mutation method to perform mutation operations for the individual as shown below:



- Step 1 Select a mutation point M_p to form individual randomly.
- Step 2 Choose a serial number for the candidate service sets for abstract service to replace the serial number in M_p .

3.2.5 The population updating strategy

To ensure that population evolution does not undergo evolutionary recession, we use the coverage method to update the population.

- Step 1 Arrange the individuals of the current population in descending order according to their fitness.
- Step 2 According to the order, the top Re_N individuals in the current population replaced the elitist individual.

$$Re_N = R_{replace} * Size_{pop} \tag{10}$$

In which, $R_{replace}$ is the population replacement rate and $Size_{pop}$ is the current population size.

4 The simulation and result analysis

For verifying the performance of EGS-QoS, we use matlab2014a on a 64-bit windows operating system, Intel Core i3-9100E, 3.60 GHz, and 8 GB RAM to conduct simulation experiments. In the simulation, EGS-QoS is compared with traditional genetic algorithms (TGA) [24] in terms of the maximum fitness of individuals.

4.1 Simulation settings

In this paper, we consider five common QoS attributes for the IoT service, and all QoS attributes are uniform distribution in a certain interval as shown in Table 1. Meanwhile, the user's preference weight for reliability, reputation, security, cost, and response time are 0.3, 0.25, 0.2, 0.15, 0.1, respectively. For the enhanced genetic algorithm, the crossover rate is set to 0.2, the mutation rate is set to 0.35.

Table 1 QoS attributes considered for the simulation

| The positive QoS attributes | | | The negative QoS attributes | |
|-----------------------------|------------|-----------|-----------------------------|---------------|
| Reliability | Reputation | Security | Cost | Response time |
| [0.3,0.7] | [0.2,0.8] | [0.1,0.9] | [100,500] | [0.2,0.4] |

In this work, we set various abstract service class numbers n and concrete service numbers m for each abstract service class A_i to evaluate the performance of EGS-QoS. The setting of abstract service class n and concrete service number m is shown in Table 2.

4.2 The results analysis

In the simulation, we show the performance of the EGS-QoS algorithm in terms of maximum fitness of individuals [22, 23]. We set various number of abstract service classes and concrete service for each abstract service class, which is shown in Table 2.

4.2.1 Maximum fitness of individual versus evolutionary iteration

Figure 4 is maximum fitness of individual versus evolutionary iteration. For the first simulation, the number of abstract service classes and the number of concrete services for each abstract service class are set to 10 and 500, respectively. The number of evolutionary iterations is 500, and the tolerance factor is set to 0.4, 0.5 and 0.6 respectively. From Fig. 4, it is can be seen that, compared with TGA, the EGS-QoS algorithm converges

Table 2 Simulation Setting of abstract service class n and concrete service number m

| The Condition number | The abstract service class n | The concrete service number m |
|----------------------|--------------------------------|---------------------------------|
| 1 | 10 | Changing in [100,500] |
| 2 | Changing in [10,50] | 100 |

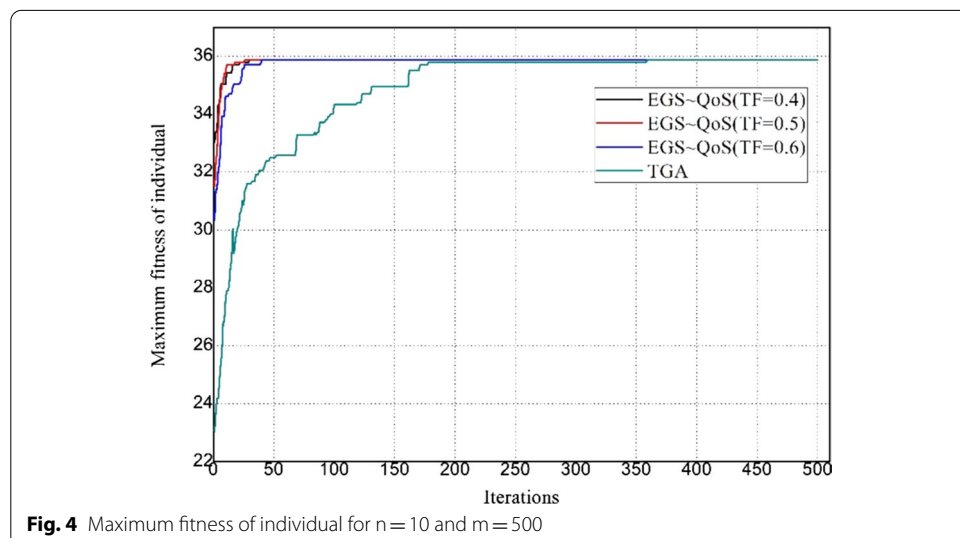


Fig. 4 Maximum fitness of individual for $n = 10$ and $m = 500$

to the global optimal solution before 50 generations on average. Therefore, the convergence speed of EGA-QoS is higher than that of TGA, and its global optimization ability is greater than TGA.

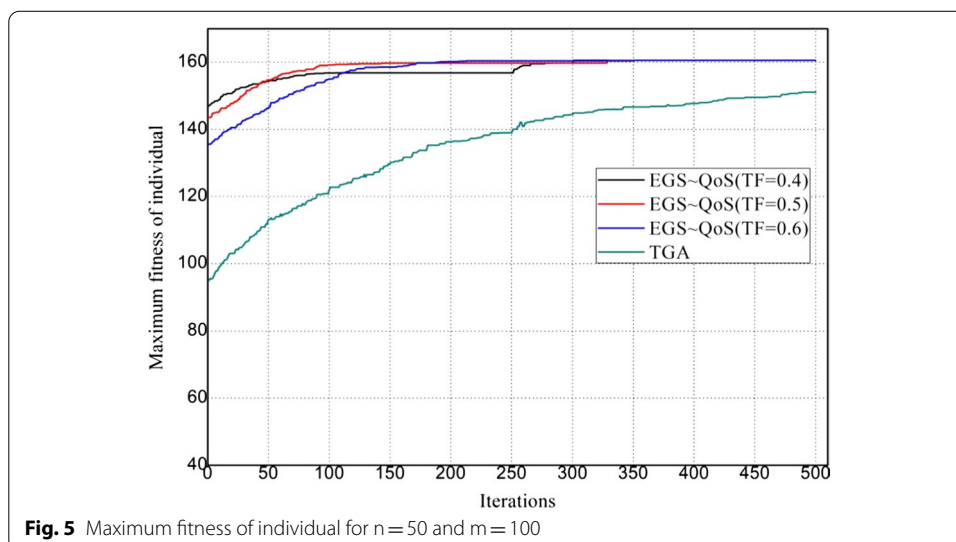
Figure 5 is maximum fitness of individual versus evolutionary iteration. For the second simulation, the number of abstract service classes and the number of concrete services for each abstract service class are set to 50 and 100, respectively. The number of evolutionary iterations is 500, and the tolerance factor is set to 0.4, 0.5 and 0.6 respectively. As shown in Fig. 5, compared with TGA, the EGS-QoS algorithm converges to the global optimal solution before 300 generations on average. Therefore, the convergence speed of EGA-QoS is higher than that of TGA, and its global optimization ability is greater than TGA.

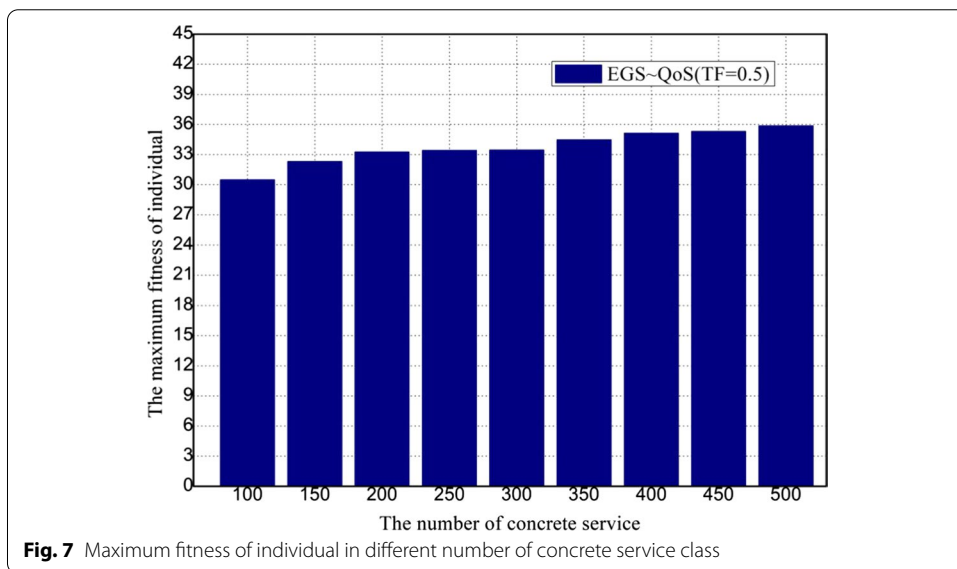
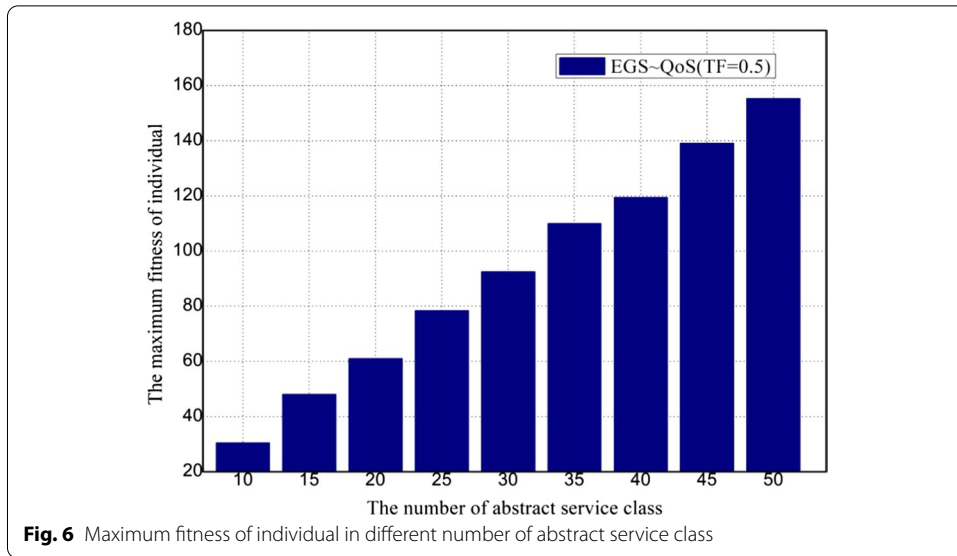
4.2.2 Maximum fitness of individual versus number of abstract service class

Figure 6 is maximum fitness of individual versus number of abstract service class in different seniors. For the third simulation, the number of abstract service classes and the number of specific services is consistent with condition 2 in Table 2, and the tolerant factor is 0.5. From Fig 6, it is very obvious that as the number of abstract classes increases, the maximum individual fitness also increases, which is due to the fact that as the number of abstract service classes increases, the scale of composite services also increases.

4.2.3 Maximum fitness of individual versus number of concrete services

Figure 7 is maximum fitness of individual versus number of concrete service class. For the fourth simulation, the number of abstract service classes and the number of specific services is consistent with condition 1 in Table 2, and the tolerant factor is 0.5. It is clearly evident from the Fig. 7 that as the number of concrete services increases, the maximum individual fitness also increases, which is due to the fact that as the number of concrete services increases, the probability of choosing the service with the highest fitness also increases.





5 Results and discussion

In this paper, we propose a hybrid service selection method based on enhanced genetic algorithm. The simulation results indicate that the global search ability and convergence speed of the proposed algorithm are better than the related algorithm. Meanwhile, the proposed algorithm can efficiently and accurately find the optimal composite service, which has higher QoS and meets the user’s needs. However, energy consumption is not considered in the service selection. In fact, when the size of the service class increases, the energy consumption increases dramatically. Hence, we should take into account both energy consumption and QoS of service selection in internet of things. We will explore this issue in our future work.

Abbreviations

QoS: Quality of service; EGS-QoS: Hybrid service selection method based on the enhanced Genetic algorithm; IoT: Internet of Things; SoA: Service-oriented architecture; SAW: Simple additive weighting; TGA: Traditional genetic algorithms.

Acknowledgements

The authors acknowledged the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Authors' contributions

Xiaofei Zhang is responsible for the idea of the manuscript, Juncheng Geng and Jianwei Ma is responsible for data collection and analysis of the experiment, Hao Liu and Shuangxia Niu are responsible for the writing of the manuscript. All authors read and approved the final manuscript.

Funding

This paper is financially supported by the science and technology project of State Grid Corporation of China(5200-201924303A-0-0-00).

Availability of data and materials

Data sharing is not applicable to this paper as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Received: 29 October 2020 Accepted: 7 December 2020

Published online: 06 January 2021

References

1. I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, D.P. Agrawal, Choices for interaction with things on Internet and underlying issues. *Ad Hoc Netw* **28**(9), 68–90 (2015)
2. Z. Huang, X. Xu, J. Ni, H. Zhu, C. Wang, Multimodal representation learning for recommendation in Internet of Things. *IEEE Internet Things J* **6**(6), 10675–10685 (2019)
3. P. Asghari, A. Rahmani, H. Javadi, Service selection approaches in IoT: a systematic review. *J Netw Comput Appl* **34**(6), 61–77 (2018)
4. C. Jatoth, G. Gangadharan, R. Buyya, Computational intelligence based QoS-aware web service selection: a systematic literature review. *IEEE Trans Serv Comput* **10**(3), 475–492 (2017)
5. M. Khanouche, F. Attal, Y. Amirat, Clustering-based and QoS-aware services composition algorithm for ambient intelligence. *Inf. Sci.* **10**(3), 419–439 (2019)
6. M. Khanouche, Y. Amirat, A. Chibani, Energy-centered and QoS-aware services selection for internet of things. *IEEE Trans Autom Sci Eng* **13**(3), 1256–1269 (2016)
7. A. Yachir, Y. Amirat, A. Chibani, N. Badache, Event-aware framework for dynamic services discovery and selection in the context of ambient intelligence and internet of things. *IEEE Trans Autom Sci Eng* **13**(1), 85–102 (2016)
8. H. Liang, J. Zou, K. Zuo, K. Muhammad Junaid, An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system. *Mech Syst Signal Process* **142**, 106708 (2020)
9. A. Kouicem, A. Chibani, A. Tari, Y. Amirat, and Z. Tari, Dynamic services selection approach for the composition of complex services in the web of objects, in 2014 IEEE World Forum on Internet of Things (WF-IoT), 298–303(2014).
10. Y. Zhang, R. Zhu, Z. Chen, J. Gao, D. Xia, Evaluating and selecting features via information theoretic lower bounds of feature inner correlations for high-dimensional data. *Eur J Oper Res* (2020). <https://doi.org/10.1016/j.ejor.2020.09.028>
11. L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam, H. Chang, QoS-aware middleware for Web services composition. *IEEE Trans Softw Eng* **30**(5), 311–327 (2004)
12. Y. Ngoko, A. Goldman, D. Milojevic, Service selection in web service compositions optimizing energy consumption and service response time. *J Internet Serv Appl* **4**(1), 1–12 (2013)
13. X. Jin, S. Chun, J. Jung, and K.-H. Lee, IoT service selection based on physical service model and absolute dominance relationship, in Proceedings of the 7th IEEE International Conference on Service-Oriented Computing and Application, Matsue, Japan, 2014.
14. L. Li, S. Li, S. Zhao, QoS-aware scheduling of services-oriented internet of things. *IEEE Trans Ind Inf* **10**(2), 1497–1505 (2014)
15. C. Jatoth, G. Gangadharan, R. Buyya, Computational intelligence based QoS-aware web service composition: a systematic literature review. *IEEE Trans Serv Comput* **9**(6), 148–152 (2015)
16. P.C. Fishburn, Exceptional paper-lexicographic orders, utilities and decision rules: a survey. *Manag Sci* **20**(11), 1442–1471 (2014)
17. N. Xiong, W. Han, A. Vandenberg, Green cloud computing schemes based on networks: a survey. *IET Commun* **6**(18), 3294–3300 (2012)
18. D. Stefanoiu, J. Culita, F. Ionescu, Vibration fault diagnosis through genetic matching pursuit optimization. *Soft Comput* **23**(17), 1234–1239 (2019)
19. C.A.C. Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inf* **16**(3), 193–203 (2002)
20. G. Güngör-Demirci, A. Aksoy, Evaluation of the genetic algorithm parameters on the optimization performance: a case study on pump-and-treat remediation design. *TOP* **18**, 2 (2010)
21. H. Liang, A. Xian, M. Min Mao, P. Ni, H. Wu, A research on remote fracturing monitoring and decision-making method supporting smart city. *Sustain City Soc* **62**, 102414 (2020)

22. H. Liang, D. Zou, Z. Li, K. Muhammad Junaid, Y. Lu, Dynamic evaluation of drilling leakage risk based on fuzzy theory and PSO-SVR algorithm. *Future Gener Comput Syst* **95**, 454–466 (2019)
23. H. Zheng, W. Guo, N. Xiong, A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems. *IEEE Trans Syst Man Cybernet Syst* **48**(12), 2315–2327 (2017)
24. J. Yi, L. Xing, G. Wang et al., Behavior of crossover operators in NSGA-III for large-scale optimization problems. *Inf Sci* **509**(3), 470–487 (2020)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
