

A QoS-driven Service Selection Optimization Algorithm for Internet of Things

Xiaofei Zhang (✉ zzdw_zxf@tom.com)

China Electric Power Research Institute <https://orcid.org/0000-0001-7178-7451>

Juncheng Geng

Electric Power Research Institute

Jianwei Ma

Electric Power Research Institute

Hao Liu

Electric Power Research Institute

Shuangxia Niu

Electric Power Research Institute

Research

Keywords: internet of things, hybrid optimization algorithm, service composition, quality of service

Posted Date: September 8th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-69961/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A QoS-driven Service Selection Optimization Algorithm for Internet of Things

Xiaofei Zhang*, Juncheng Geng, Jianwei Ma, Hao Liu, Shuangxia Niu
(Electric Power Research Institute of State Grid Henan Electric Power Company,
450000, China)
E-mail: zzdw_zxf@tom.com

Abstract Currently, the smart devices have been widely deployed in Internet of Things (IoT). With the scale of IoT continues to increase, it brings big challenges for service composition in a large-scale IoT. For solving this problem, a QoS-driven service selection method based on the enhanced Genetic algorithm (EGS~QoS) is proposed in this paper. To decrease the scale of service composition, we use the lexicographic optimization approach and QoS constraint relaxation technique to find the candidate service with height QoS. Then, the IoT service composition 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 is found from the candidate service. Compared with the related algorithm, the simulation results show that EGS~QoS can efficiently and quickly select 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 composition, quality of service

1. Introduction

Internet of Things (IoT) 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 continuous 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 continuous improvement 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 composition, the characteristic of today's IoT service composition is that all IoT services run directly in smart devices with limited power, storage capacity, and computing resources [4]. Therefore, the non-functional attributes, also call QoS, 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 this article, to solve the service composition problem concerning QoS needs in the large-scale IoT, we propose a QoS-driven service selection method based on the

enhanced genetic algorithm (EGS~QoS).

This algorithm mainly consists of two parts: First, for reducing the scale of service composition, the QoS~driven 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 elitist genetic algorithm [8] is used to search the service composition way 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 in this paper. Section 2 summarizes related work for IoT service composition. Section 3 introduces the model of IoT service. Section 4 details the EGS~QoS 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 composition. 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 QoS~driven service composition problem is NP-hard problem. Many scholars have solved this problem in the past.

Kouicem et al. [9] and Yachir et al. [10] use the Simple Additive Weighting (SAW) technique to calculate the QoS utility of the service, then use the service with the largest utility value as the final composite service. L. Zeng Et al.[12] use Integer linear programming to successfully solve the problem of service composition, then propose two-step QoS~driven services composition methods. First, regardless of global QoS constraints, a local selection strategy is implemented for each service class, and candidate services that meet the requirements are selected. Then, using the ILP method finds 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. [13] apply a mixed-integer linear programming method to solve service composition problems and take energy consumption as a QoS attribute to optimize the process of composition. However, as the problem scale increases, the efficiency of algorithm selection will decrease, which is not suitable for large-scale service composite problems. X. Jin et al. [14] adopts the concept of Pareto optimality to deal with the problem of service composition and proposes a service model that uses four QoS attributes to describe the characteristics of IoT services. Through a single QoS rating function (fitness function), the candidate service is evaluated and the composition service that meets the requirements is selected.

Different from previous studies, in this article, we combined the lexicographic optimization method [17] and threshold relaxation technology [6] to preselect candidate services with higher QoS, and then use the genetic algorithm to search for the final combination of services from the candidate service according to fitness value.

3. Models description and optimization goal

In the process of service composition for IoT, there are mainly two types of services: the concrete services C_S and the abstract service classes A_S . 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 [15]. As shown in Figure 1, the essence of QoS-driven service composition is: First, integrate the existing service classes into a new comprehensive class. Then, find the optimal concrete service form each abstract service to composite service for new service class [16].

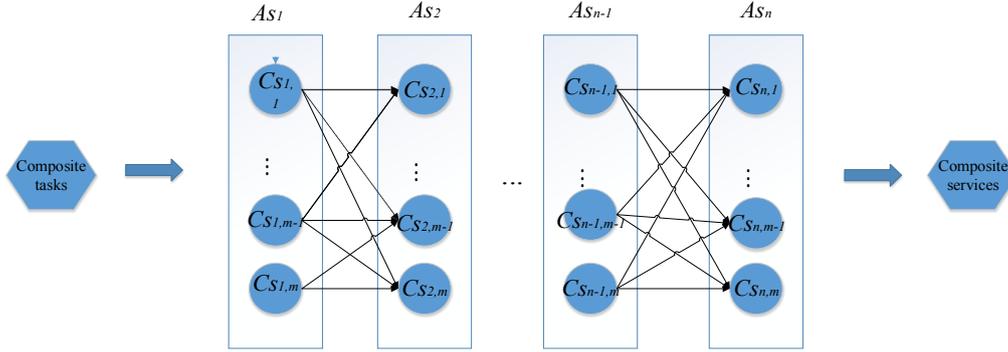


Figure.1 the model of QoS-driven service composition

3.1 The service models in IoT

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

$$C_{S_{ij}} = \{QoSA(C_{S_{ij}}), Action(C_{S_{ij}})\} \quad (1)$$

Where $QoSA(C_{S_{ij}})$ is the QoS attribute of the concrete service $C_{S_{ij}}$, and $Action(C_{S_{ij}})$ represents the function of the concrete service $C_{S_{ij}}$.

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

$$A_{S_i} = \{C_{S_{i1}}, C_{S_{i2}}, \dots, C_{S_{ij}}\} \text{ and } i \in [1, n], j \in [1, m] \quad (2)$$

3.2 the QoS model

3.2.1 the concrete service QoS model

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 [6].

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

The QoS attributes of the concrete service $C_{S_{ij}}$ are as follows:

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

Where qos_t^{ij} represents the t -th QoS attribute in the concrete service $C_{S_{ij}}$.

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

$$\begin{cases} Q_{aggregationP}(Cs_{ij}) = \sum_{l=1}^{t_1} w_l qos_l^{ij}, \text{ if } qos_l^{ij} \text{ is positive} \\ Q_{aggregationN}(Cs_{ij}) = \sum_{l=1}^{t_2} w_l qos_l^{ij}, \text{ otherwise} \\ \sum_{l=1}^t w_l = 1, \text{ and } t_1 + t_2 = t \end{cases} \quad (4)$$

Where $Q_{aggregationP}(Cs_{ij})$ and $Q_{aggregationN}(Cs_{ij})$ 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.

$$Q_{aggregation}(Cs_{ij}) = 1 / (Q_{aggregationN}(Cs_{ij}) + Q_{aggregationP}(Cs_{ij})) \quad (5)$$

Where $Q_{aggregation}(Cs_{ij})$ 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 .

3.2.2 The composite service QoS model

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 article adopts a sequential structure [6].

3.3 The optimization goal

In this work, the service composition 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.

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

Where $\text{ComFitness}(x)$ 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.

4、EGS~QoS : an QoS-driven service selection method based on enhanced Genetic optimization algorithm

The EGS~QoS algorithm proposed in this paper consists of two steps. First, lexicographic optimization and threshold relaxation technology are used to preselect candidate services with higher QoS.

4.1 service preselection concerning QoS levels and user's QoS needs

In this article, 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 solves multiple subproblems ranking according to the importance of its objective problem.

In this article, 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:

- 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.
- 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.
 - A) Discover the optimal QoS aggregation value $BestQ$ in the abstract service class As_i .
 - B) Using threshold relaxation technology [6], calculate the threshold of each QoS attribute aggregation value, based on the optimal QoS aggregation value $BestQ$ and tolerance factor δ_{QoS} .

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

$$\text{with } QoS_{Aggregation}^{Decrease} = BestQ * \delta_{QoS} \quad (8)$$

- 3) The concrete service Cs_{ij} that is not within the threshold $QoS_{Aggregation}^{Threshold}$ 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 and threshold relaxation technology is shown in Algorithm 1.

Algorithm 1— service preselection concerning QoS levels and user's QoS needs
Input :abstract service class As_i containing m concrete service. δ_{QoS} : the tolerance factor for QoS aggregation value.
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.
Step 2: $CAS = As_i$
Step 3: for rank=1 to 2 do
Step 4: discover $BestQ$ from CAS
Step 5: compute $QoS_{Aggregation}^{Decrease}$ according to (8)
Step 6: compute $QoS_{Aggregation}^{Threshold}$ according to (7)
Step 7: Update CAS as follows:
Step 8: if current QoS aggregation value is positive then
Step 9 : $CAS = \{Cs_{ij} \in CAS \cap Q_{aggregationP}(Cs_{ij}) \geq QoS_{Aggregation}^{Threshold}\}$
Step 10 : else
Step 11 : $CAS = \{Cs_{ij} \in CAS \cap Q_{aggregationN}(Cs_{ij}) \leq QoS_{Aggregation}^{Threshold}\}$
Step 12 : end if

Step 13 : end for

Outputs : the pre-selecting candidate service sets CAS for each abstract service class As_i .

4.2 Final composition service selection based on enhanced genetic algorithm

In this paper, concrete services Cs_{ij} with relatively low QoS are deleted through service preselection for each abstract service class As_i , which reduces the search domain to a certain extent. The service composition 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 [8] to find the final composite service. The flow chart of the enhanced genetic algorithm is shown in Figure.2.

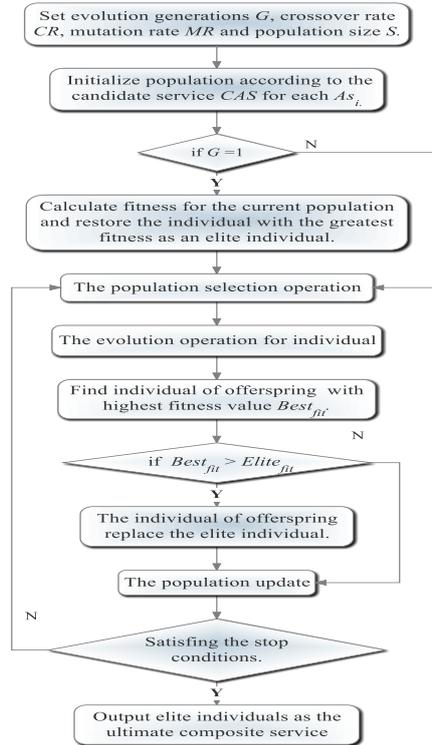


Figure. 2 The process of enhanced genetic algorithm

For the enhanced genetic algorithm, the current population is generated according to the number of concrete services contained in the CAS candidate service set of each abstract class As_i . 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.

$$Gertaions = \frac{n*m}{10} \quad (9)$$

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

4.2.1 the population selection operation

Commonly used methods of population selection are roulette wheel selection [18], stochastic universal sampling [19], and tournament selection method [20]. To ensure

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

4.2.2 the Individual coding method in Evolutionary operation

In this article, the coding method of population individuals adopts integer coding. As shown in Figure.3, the individual length is the total number n of abstract service classes participating in service composition. 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 .

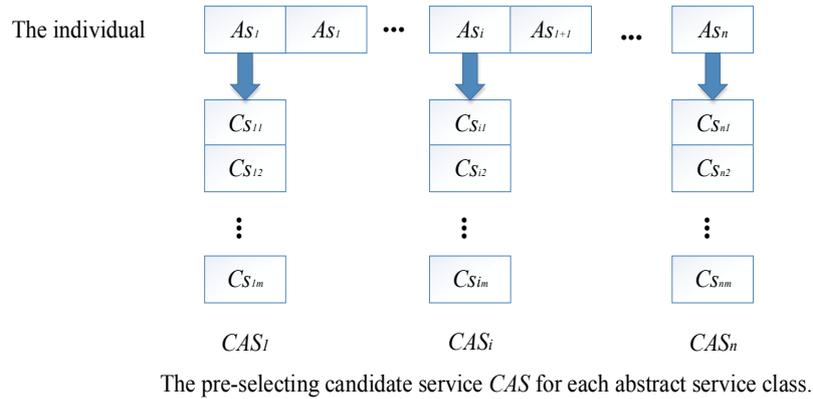


Figure.3 individual coding method

4.2.3 the Individual crossover method in Evolutionary operation

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

4.2.4 the Individual mutation method in Evolutionary operation

for ensuring the integrity of the individual, this paper uses 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 : chose a serial number for the candidate service Sets for abstract service AS_i to replace the serial number in M_p .

4.2.5 the Population updating strategy

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

Step1: Arrange the individuals of the current population in descending order according to their fitness.

Step2: According to the order, the top Re_N individuals in the current population replaced the elitist individual.

$$Re_N = R_{replace} * Size_{pop} \quad (10)$$

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

5. The simulation and result analysis

For verifying the performance of EGS~QoS, we use matlab2014a on a 64-bit Windows operating system, Intel Core i3-9100F, 3.60GHz, and 8GB RAM to conduct

simulation experiments. In the simulation, EGS~QoS will be compared with traditional genetic algorithms(TGA) [22] in terms of the maximum fitness of individuals.

5.1 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.

In this work, we set various abstract service class numbers n and concrete service numbers m for each abstract service class A_{s_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.

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]

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

5.1 The results analysis

In the simulation, we show the performance of the EGS~QoS algorithm in terms of maximum fitness of individuals. We set various number of abstract service classes and concrete service for each abstract service class, which is as shown in table 2.

5.1.2 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, for the EGS~QoS and TGA algorithms, and the tolerance factor is set to 0.4, 0.5 and 0.6 respectively. From Figure.4, it is easy to see that, compared with TGA, the EGS~QoS algorithm converges 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.

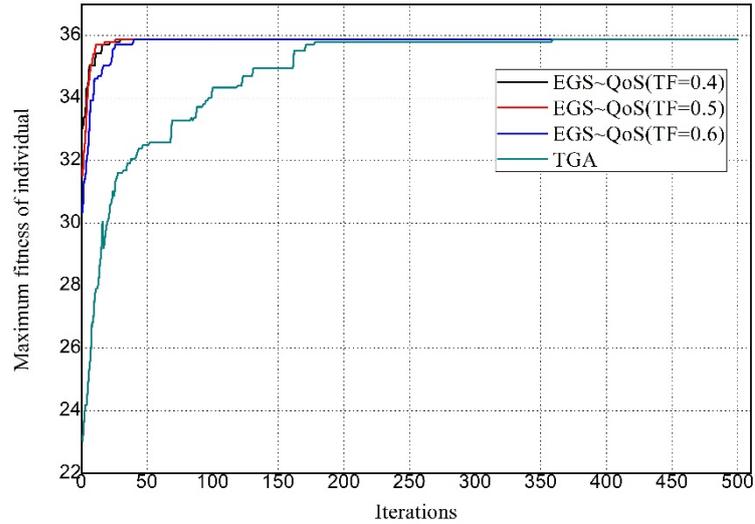


Figure.4 maximum fitness of individual for n=10 and m=500

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. From Figure. 5, it is can be seen that, 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.

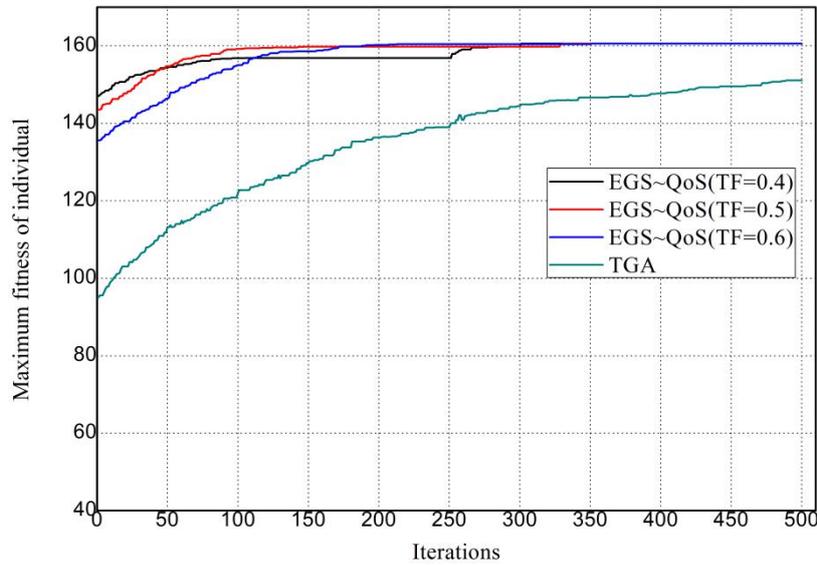


Figure. 5 maximum fitness of individual for n=50 and m=100

5.1.3 Maximum fitness of individual Versus number of abstract service class

For the third simulation, the number of abstract service classes and the number of specific services are consistent with condition 2 in table 2, and the tolerant factor is 0.5. From Figure. 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.

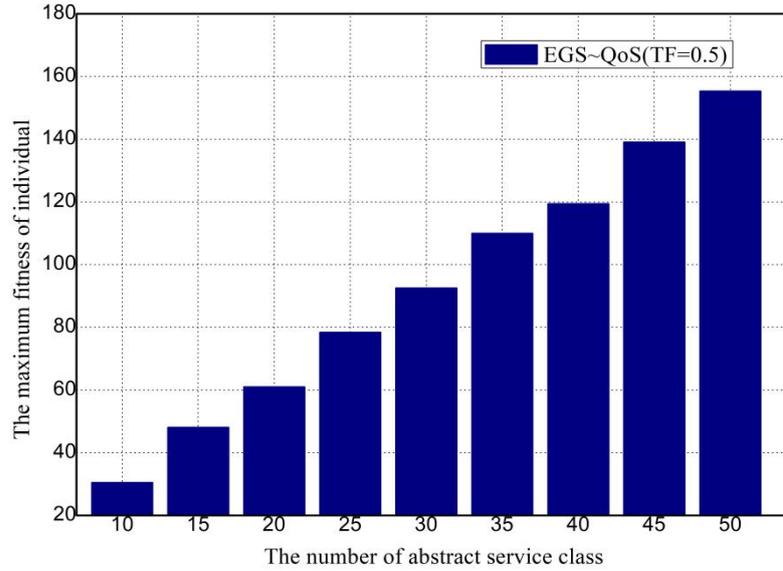


Figure. 6 maximum fitness of individual in different number of abstract service class

5.1.4 Maximum fitness of individual versus number of concrete service

For the fourth simulation, the number of abstract service classes and the number of specific services are consistent with condition 1 in table 2, and the tolerant factor is 0.5. From Figure.7, it is not difficult to find 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.

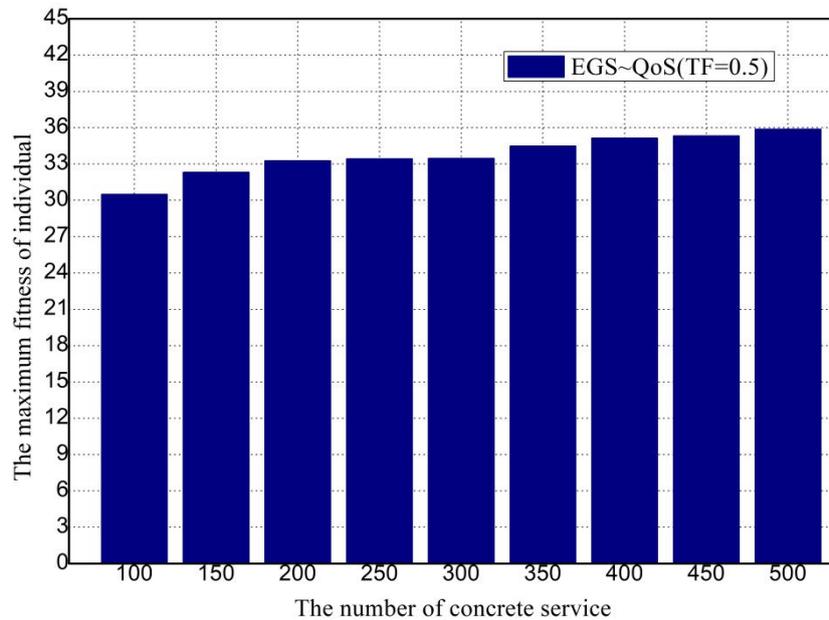


Figure.7 maximum fitness of individual in different number of concrete service class

6. Results and Discussion

In this paper, for dealing with the service composition problem in the IoT

environment within QoS constraints and user's needs for QoS, we propose an QoS-driven 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, in this study, we donot take the energy computation into account. It is should be futher studied that how to quickly combine the services provided by IoT components into a comprehensive composition service to balance energy and quality of service(QoS). Also, how to design more efficient algorithm is a key issue.

Declarations

Availability of data and materials

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

Competing interests

The authors declare that they have no competing interests.

Funding

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

Authors Contribution

Xiaofei Zhang is responsible for the idea of the manuscript, Juncheng Geng and Jianwei Ma is 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.

Acknowledgements

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

Abbreviations

QoS: quality of service; EGS~QoS: QoS-driven 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

Reference

- [1] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, pp. 68-90, 2015/05/01/ 2015.
- [2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things:

- A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.
- [3] I. Gartner. (2017, February 12). Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. Available: show the performance of the EGS-QoS algorithm in terms of maximum fitness of individual.
- [4] Alsaryrah O , Mashal I , Chung T Y D . Energy || ware Services Composition for Internet of Things[C]// IEEE 4th World Forum on Internet of Things. IEEE, 2018.
- [5]P. C. Fishburn, "Exceptional paper-lexicographic orders, utilities and decision rules: A survey," *Manag. Sci.*, vol. 20, no. 11, pp. 1442–1471, Jul. 1974.
- [6]Energy-Centered and QoS-Aware Services Selection for Internet of Things
- [7] A. Yachir, Y. Amirat, A. Chibani, and N. Badache: Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things, *IEEE Transactions on Automation Science and Engineering*, 13(2016) No. 1, p. 85-102, 2016.
- [8] Cheng W , Shi H , Yin X , et al. An Elitism Strategy Based Genetic Algorithm for Streaming Pattern Discovery in Wireless Sensor Networks[J]. *IEEE Communications Letters*, 2011, 15(4):419-421.
- [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), 2014, pp. 298-303.
- [10] A. Yachir, Y. Amirat, A. Chibani, and N. Badache, "Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 85-102, 2016.
- [12]] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for Web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.
- [13] Y. Ngoko, A. Goldman, and D. Milojicic, "Service selection in web service compositions optimizing energy consumption and service response time," *J. Internet Services and Applicat.*, vol. 4, no. 1, pp. 1–12, Dec. 2013.
- [14] X. Jin, S. Chun, J. Jung, and K.-H. Lee, "IoT service selection based on physical service model and absolute dominance relationship," in *Proc. 7th IEEE Int. Conf. Service-Oriented Computing and Applicat.*, Matsue, Japan, 2014, pp. 65–72.
- [15] L. Li, S. Li, and S. Zhao, "QoS-aware scheduling of services-oriented internet of things," *IEEE Trans. Ind. Inf.*, vol. 10, no. 2, pp. 1497–1505, May 2014.
- [16] C. Jatoth, G. Gangadharan, and R. Buyya, "Computational intelligence based QoS-aware web service composition: A systematic literature review," *IEEE Trans. Services Computing*, vol. PP, no. 99, 2015, DOI: 10.1109/TSC.2015.2473840
- [17] P. C. Fishburn, "Exceptional paper-lexicographic orders, utilities and decision rules: A survey," *Manag. Sci.*, vol. 20, no. 11, pp. 1442–1471, Jul. 1974.
- [18] Yu-Hong L , Xin Z . An improved method for roulette wheel selection of genetic algorithm[J]. *Information Technology*, 2009.
- [19] Dan Stefanoiu,Janetta Culita,Florin Ionescu. Vibration fault diagnosis through genetic matching pursuit optimization[J]. *Soft Computing*,2019,23(17).
- [20] Coello C A C , Efrén Mezura Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection[J]. *Advanced Engineering Informatics*, 2002, 16(3):193-203.
- [21]. Gamze Güngör-Demirci,Ayşegül Aksoy. Evaluation of the genetic algorithm parameters on the optimization performance: a case study on pump-and-treat remediation design[J]. *TOP*,2010,18(2).
- [22] Yi J, Xing L, Wang G, et al. Behavior of crossover operators in NSGA-III for large-scale optimization problems[J]. *Information Sciences*, 2020,509: 470-487 (online)

Figure Title and Legend Section

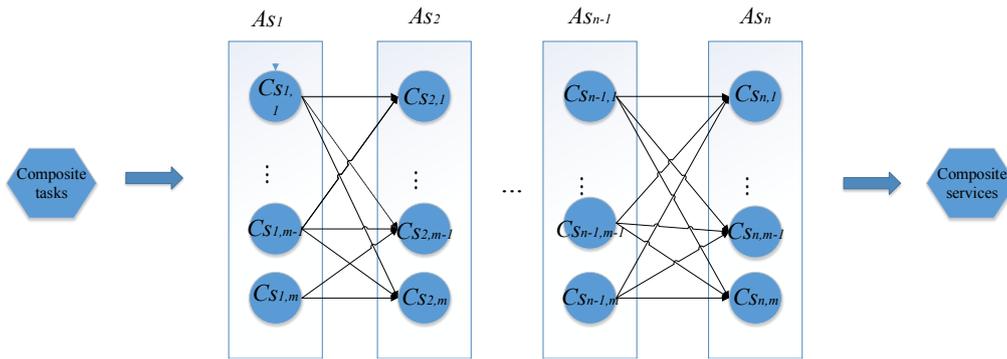


Figure.1 the model of QoS-driven service composition

There are mainly two types of services: the concrete services and the abstract service classes. The essence of QoS-driven service composition is: First, integrate the existing service classes into a new comprehensive class. Then, find the optimal concrete service form each abstract service to compose service for new service class.

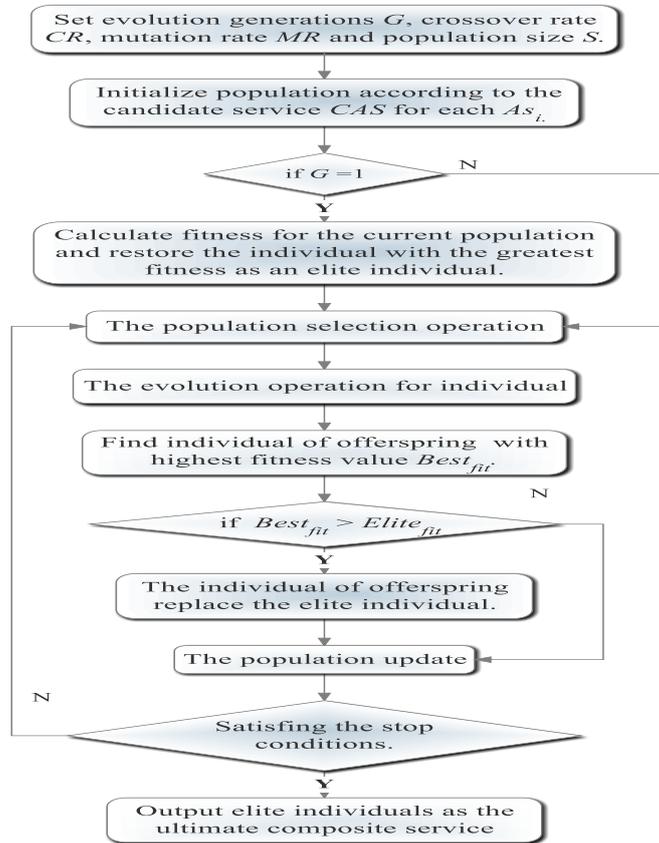


Figure. 2 The process of enhanced genetic algorithm

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

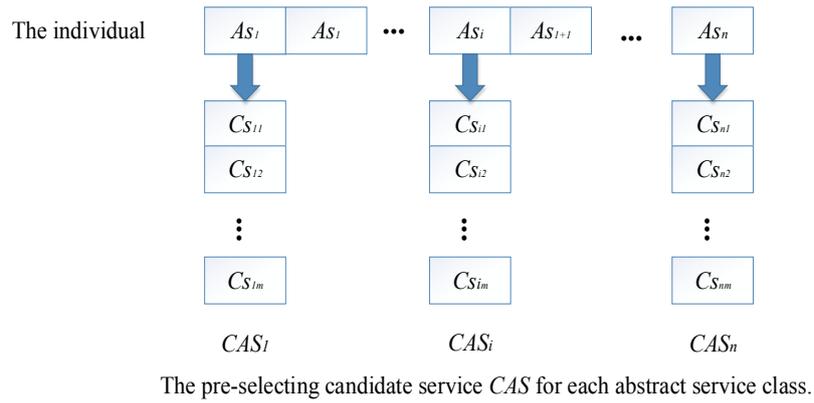


Figure.3 individual coding method

In this article, the coding method of population individuals adopts integer coding. As shown in Figure.3, the individual length is the total number n of abstract service classes participating in service composition. 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 .

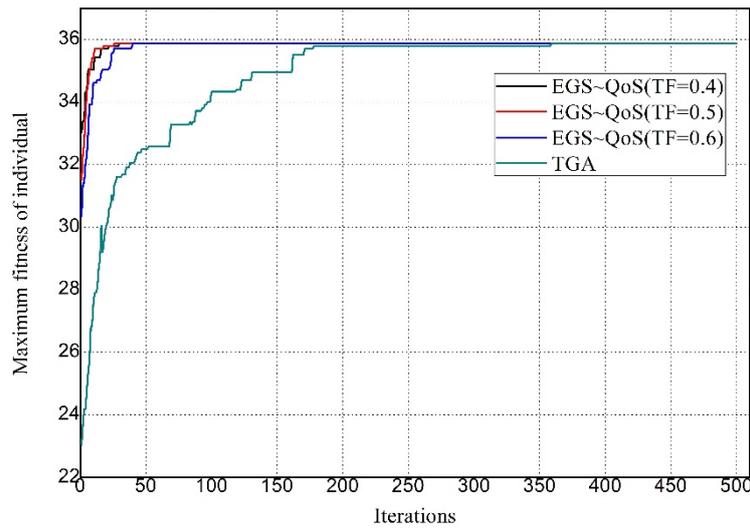


Figure.4 maximum fitness of individual for $n=10$ and $m=500$

From Figure.4, it is easy to see that, compared with TGA, the EGS-QoS algorithm converges 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.

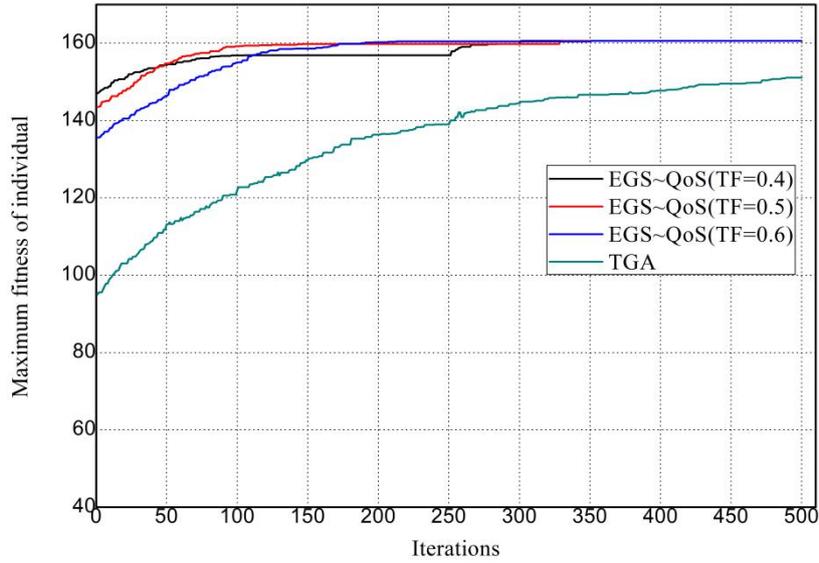


Figure. 5 maximum fitness of individual for n=50 and m=100

From Figure. 5, it is can be seen that, 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.

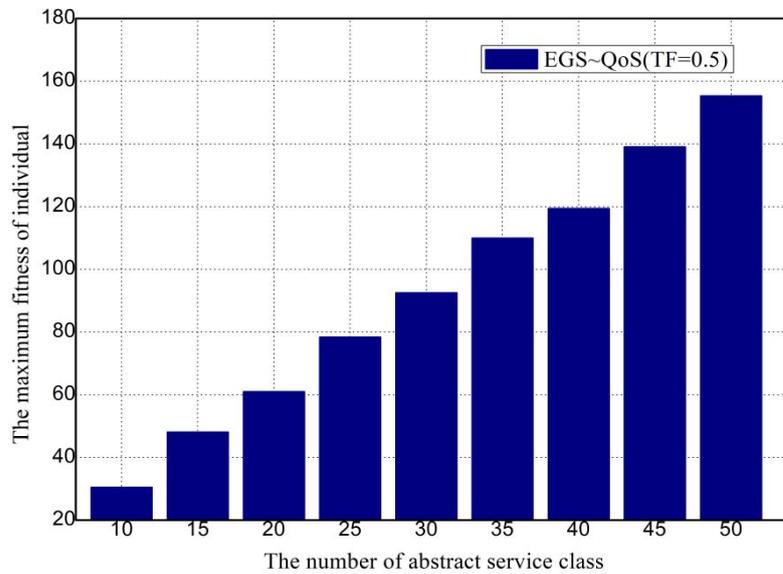


Figure. 6 maximum fitness of individual in different number of abstract service class

From Figure. 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.

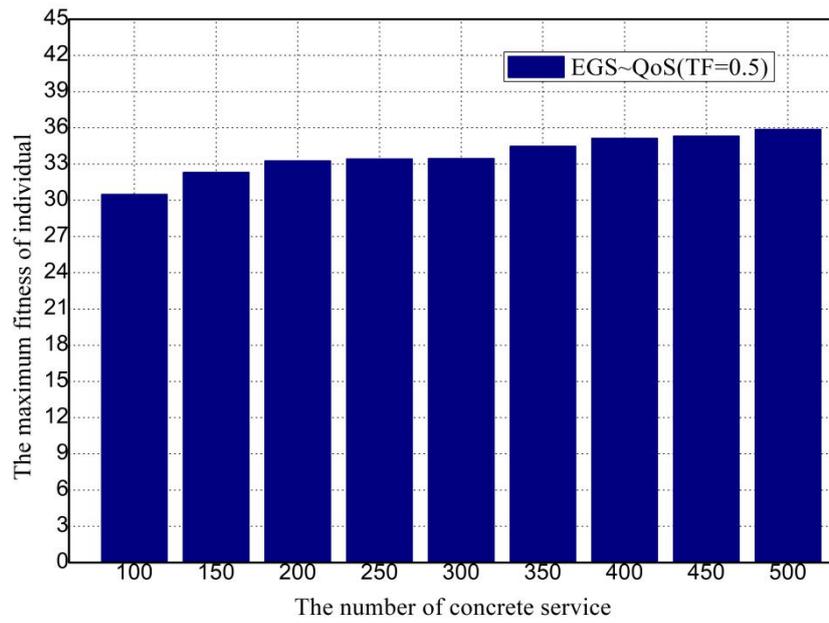


Figure.7 maximum fitness of individual in different number of concrete service class

From Figure.7, it is not difficult to find 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.

Figures

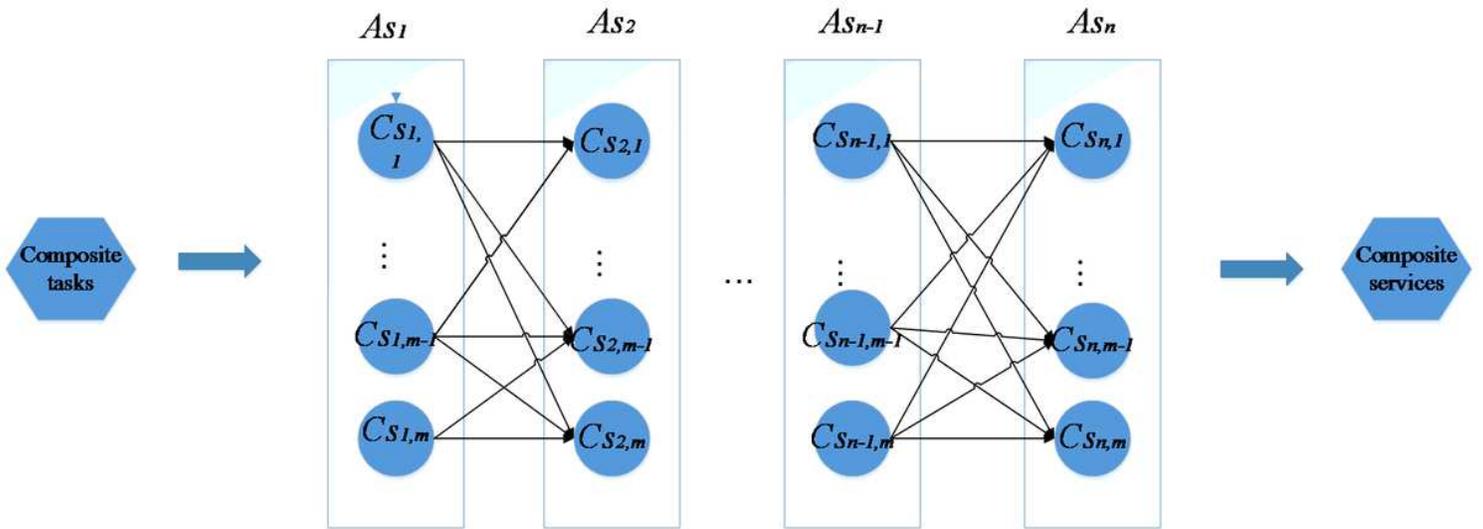


Figure 1

the model of QoS-driven service composition

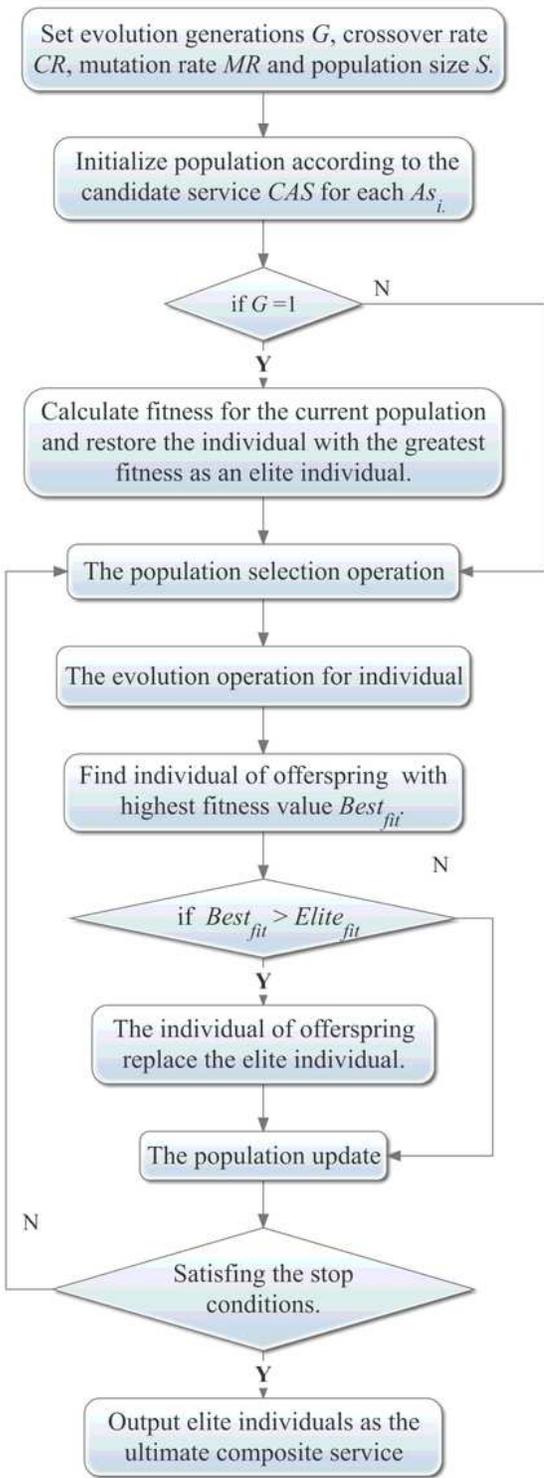
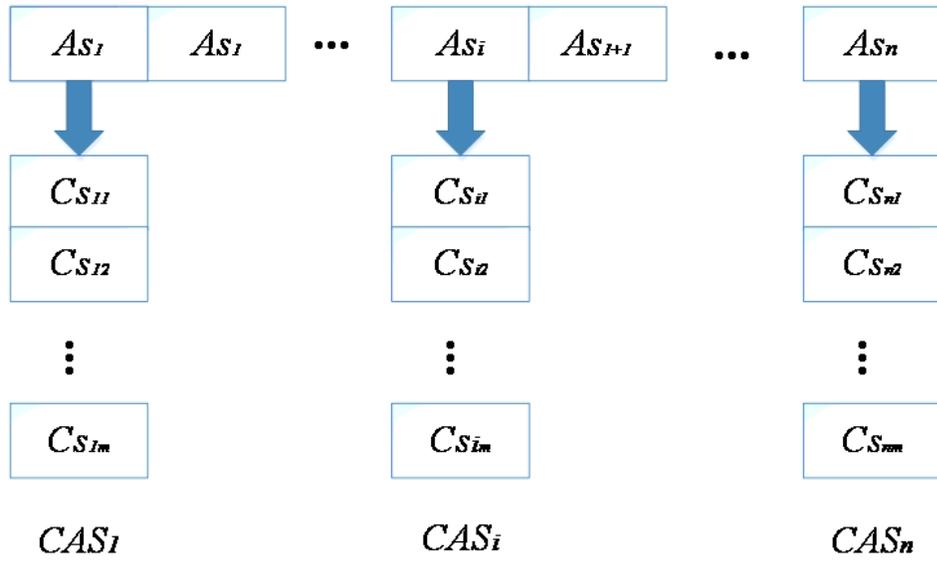


Figure 2

The process of enhanced genetic algorithm

The individual



The pre-selecting candidate service CAS for each abstract service class.

Figure 3

individual coding method

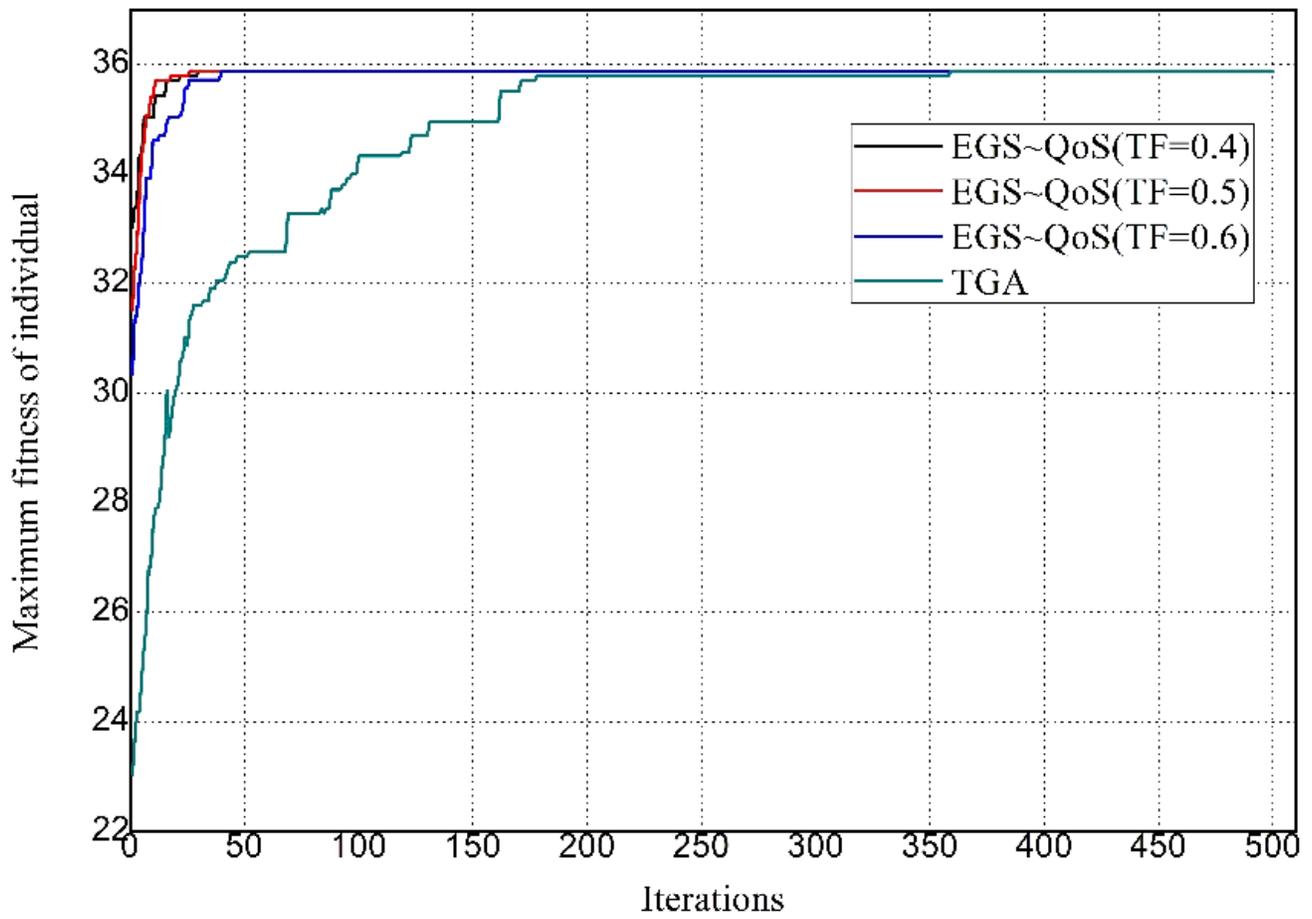


Figure 4

maximum fitness of individual for $n=10$ and $m=500$

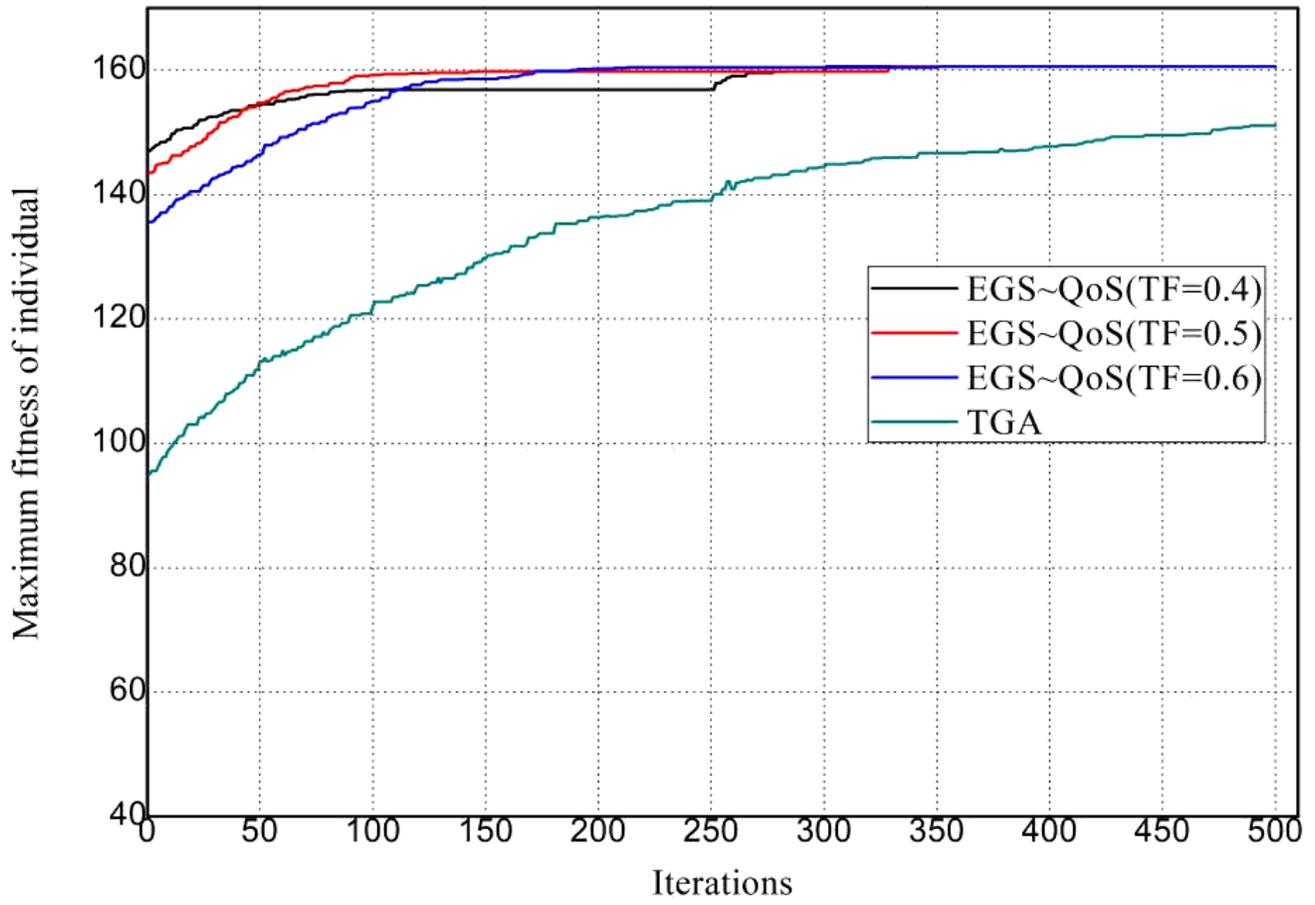


Figure 5

maximum fitness of individual for $n=50$ and $m=100$

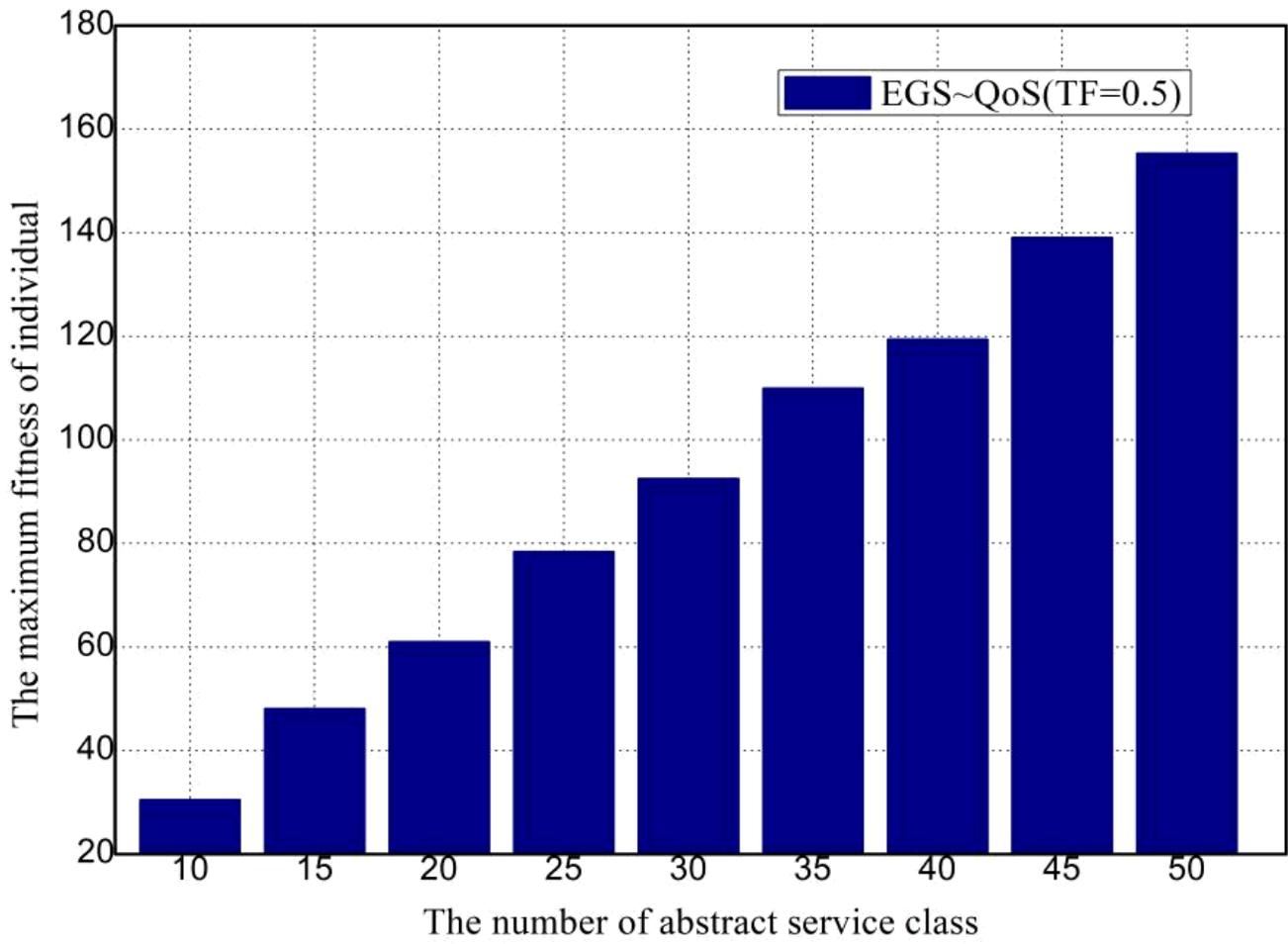


Figure 6

maximum fitness of individual in different number of abstract service class

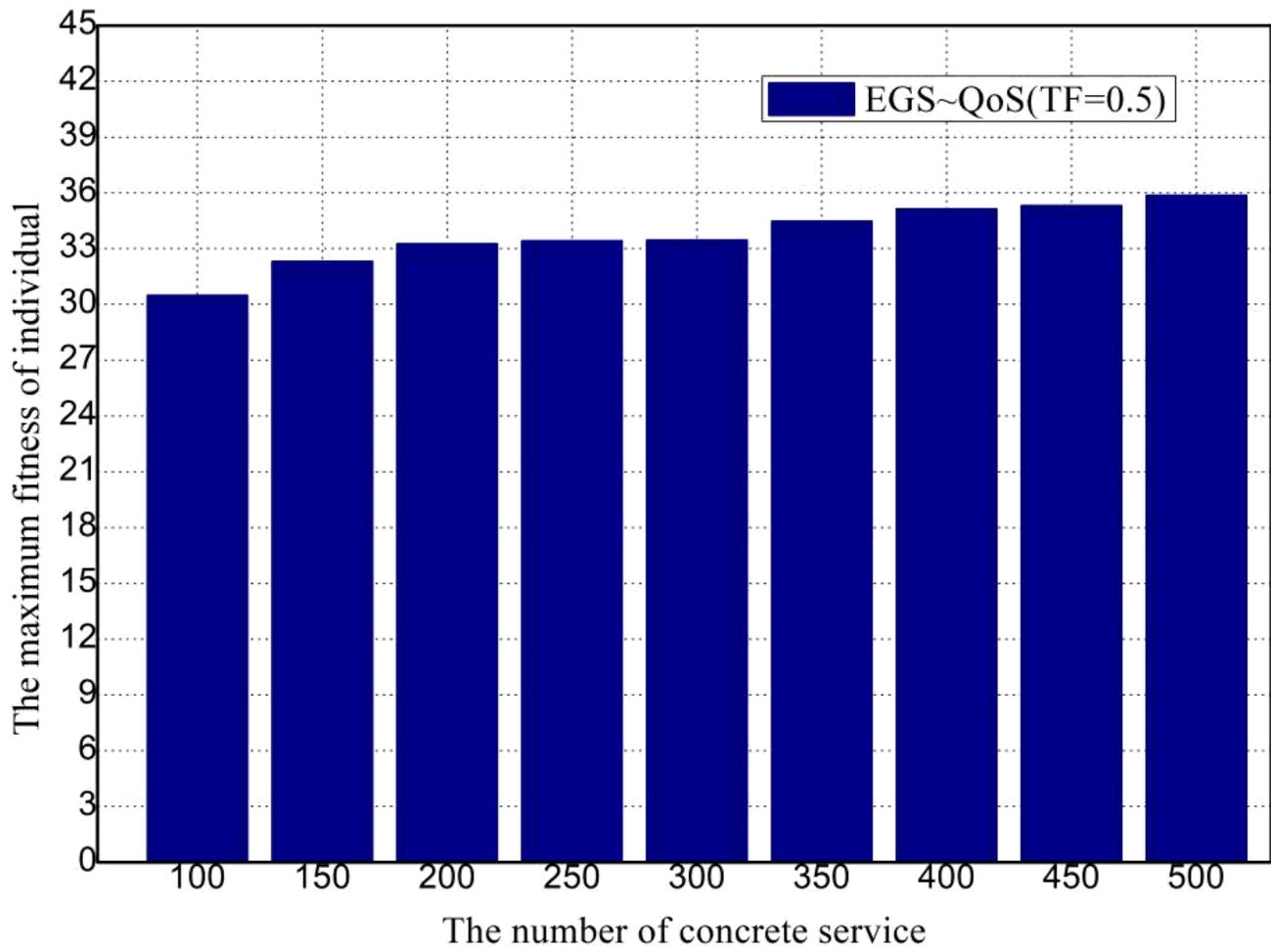


Figure 7

maximum fitness of individual in different number of concrete service class