A Multi-Objective Service Selection Method Based on Ant Colony Optimization for QoE Restrictions in the Internet of Things

Chuxuan Zhang[1,2], Bing Jia[1,2](✉), and Lifei Hao[1,2]

[1] College of Computer Science, Inner Mongolia University, Hohhot 010021, China
jiabing@imu.edu.cn
[2] Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China

Abstract. With the development of Wireless Sensor Network (WSN), the number of Internet of Things (IoT) services has increased dramatically. In order to use IoT services conveniently, it has become a key issue to reasonably aggregate information, content and applications, and filter services according to users' needs. Most of the existing service selection algorithms adopt heuristic search algorithm or Genetic Algorithm (GA). The heuristic algorithm is not stable, and GA cannot meet the needs of service selection because of the one-dimensional chromosome coding. For overcoming the disadvantages of these methods, this paper proposes a multi-objective service selection algorithm based on Ant Colony Optimization (ACO) for Quality of Experience(QoE) restrictions. The proposed method can get a feasible solution quickly and efficiently by utilizing the fast convergence speed of ACO. Specifically, QoE model was established firstly, and relevant constraints and quantitative methods are given. Secondly, a service selection model based on ACO was constructed to select specific services based on the above model. Finally, the proposed method is verified through simulations. Results show that, compared with GA-based method, the proposed algorithm can improve the recall rate and precision rate, and has a higher algorithm efficiency in solving the service selection problems.

Keywords: Internet of Things · Ant Colony Optimization · Service selection · QoE

1 Introduction

In recent years, the Internet of things (IoT) [1] technology has been widely concerned by people. IoT is characterized by loose coupling, platform independence, language neutrality and openness, etc. It has become an important part of the new generation of information technology. There are many IoT services, but the

J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 342–353, 2019.
https://doi.org/10.1007/978-3-030-21373-2_26

function of a single IoT service is pretty simple. The implementation of complex services requires the aggregation of multiple services. However, the selecting of services becomes difficult due to the existence of a large number of IoT services with the same or similar functions. Therefore, the realization of IoT service selection becomes a key problem to be solved.

In order to better meet the needs of users, [2] proposed the concept Quality of Experience (QoE) to solve problems from the perspective of user experience. QoE, as a means of service quality quantification, effectively helps service providers improve service quality and user's satisfaction. In [3], key indicators affecting QoE were studied and defined, and an evaluation and quantification algorithm for QoE was proposed. However, there are many factors that affect user experience, leading to the difficulty of modeling. [4] introduced user preference, adopt three-layer hierarchical model, and proposed a satisfaction calculation method based on weighted sum. However, the Analytic Hierarchy Process (AHP) algorithm used to calculate user preferences in this model is inefficient. [5] uses the expert opinions to preset user preferences, which improves the algorithm efficiency, but it is difficult to meet the needs of mobile application scenarios due to the inability to dynamically learn user preferences.

There are many algorithms for service selection, which can be summarized into four categories. The first one is Direct Search Method (DSM) [6]. This method traversed all possible paths, but was so inefficient that it was only suitable for a small number of services. Secondly, Heuristic Search Algorithm (HSA) [7], heuristics are added to speed up the search process. Although using the appropriate search strategy, the search speed is quite fast, but the stability is poor. The third category is Integer Programming Algorithm (IPA) [8]. IPA establishes the global optimization model for service aggregation and transforms this issue into a 0–1 linear programming problem. It improves search speed, but is still not ideal for large-scale service selection. Finally, Genetic Algorithm (GA) [9] is a computational model simulating natural selection and genetic mechanism in Darwinian evolution. Coding mode is the basis of GA, and will directly affect the design of selection, crossover and mutation operations, thus affecting convergence, complexity and efficiency. So different problems adopt different coding patterns, which is difficult to reuse.

To sum up, the above models and service selection methods both have advantages and disadvantages, but they are not quite suitable for IoT services. Aiming at IoT service selection for QoE restrictions, this paper proposed a multi-parameter linear weighted QoE quantitative model, and designs a corresponding service selection algorithm based on Ant Colony Optimization (ACO) [10]. The model is widely applicable and has good scalability by extensive analysis of IoT services and QoE evaluation methods. Compared with other service selection algorithms, the proposed method can effectively improve the precision rate and recall rate of service selection, and greatly reduce the computation time and complexity in the same scenario.

The rest of this paper is organized as follows. The next section formally gives a multi-parameter linear weighted QoE quantitative model for IoT services and the standardization method. The ACO is introduced in Sect. 3, and combined with the proposed model, a service selection algorithm based on ACO has been described in detail. In Sect. 4, experiments are designed and the proposed method is evaluated by extensive simulations. Finally, the last section summarizes the whole paper and outlooks the future work.

2 The Multi-parameter Linear Weighted QoE Quantitative Model

The study of the factors which affected QoE is crucial for the evaluation of QoE, because the basic objective of QoE evaluation is to predict the QoE which is difficult to measure directly from known or easily measured factors [11]. In order to quantify the QoE of IoT services reasonably, this paper proposes a multi-parameter linear weighted quantitative model by considering the characteristics of IoT services, e.g., variety and different grading factors. This model mainly examines four major aspects, including service performance experience, service provider's brand effect, users' sensitivity to price, and users' personal preferences, as shown in Fig. 1.

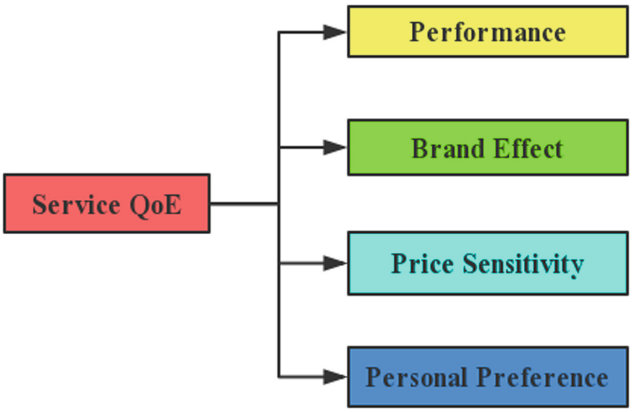


Fig. 1. The structure diagram of multi-parameter linear weighted QoE quantitative model.

Service performance experience can be directly measured by Quality of Service (QoS) parameters, including some common Internet service indicators such as response time, availability, reliability. The definition for QoE performance experience indicator of service can be formalized as follow.

$$q_p = \theta_{RT} \cdot RT' + \theta_A \cdot A + \theta_R \cdot R \tag{1}$$

where

$$RT' = \begin{cases} RT_{max} - RT, if\ RT \leq RT_{max} \\ 0, \qquad\qquad\qquad else \end{cases} \tag{2}$$

RT is the actual value of response time, A and R are the standard scores of availability and reliability, and θ_{RT}, θ_A and θ_R are the corresponding weights.

Brand Effect plays a great role in commodity economy as well as in IoT service selection. People preferred to choose the services of provider with high reputation or ranking. It is a common intuition that users have lower differentiating degree for services with a large brand effect and vice versa. So it is advisable to use logarithmic function for quantification, which is defined as Formula (3).

$$q_{be} = \ln(BE + 1) \tag{3}$$

where BE is the comprehensive score of brand effect which can be graded by incorporating reputation and ranking.

Different service prices will have different psychological experience for users such as the common 9-end commodity prices used to bring better sales. Therefore, it is appropriate to define the user's price sensitivity as a piecewise constant function as below.

$$q_{ps} = \begin{cases} S_1, & if\ 0 \leq price < p_1 \\ S_2, & if\ p_1 \leq price < p_2 \\ S_3, & if\ p_2 \leq price < p_3 \\ S_4, & if\ p_3 \leq price < p_4 \\ S_5, & if\ price < p_4 \end{cases} \tag{4}$$

where S_i is the score of price sensitivity, $S_1 < S_2 < S_3 < S_4 < S_5$ and $[p_i, p_{i+1}]$ is a price interval.

However, the previous service experience of users and the attractiveness of the content provided by the service often have a decisive impact on the user's choice, so the personal preference is represented by exponent function.

$$q_{pp} = e^{\frac{PP}{\alpha}} - 1 \tag{5}$$

where PP is a comprehensive score of personal preferences, which is determined by previous experience and service content, and α is an adjustment parameter.

Basing on the above definition of influencing factors, we can get a comprehensive QoE index of the IoT service as follow, which is the linear weighted of the above four factors.

$$qoe_i = \sum\nolimits_j \theta_j \cdot q_j \tag{6}$$

where $q_j \in \{q_p, q_{be}, q_{ps}, q_{pp}\}$ and θ_j are weight coefficients of the four influencing factors, respectively.

Owing to the above seven indicators $(RT', A, R, q_p, q_{be}, q_{ps}, q_{pp})$ are calculated by actual values, the ranges of results are different, which means it is impossible to evaluate the importance of each factor. Therefore, the Formula (7) is given according to [12], which can standardize each sub-index linearly so that their ranges of value are between $[0, 1]$.

$$P_i = \begin{cases} \frac{q_i - q_i^{min}}{q_i^{max} - q_i^{min}}, & if\ q_i^{max} - q_i^{min} > \varepsilon \\ 1, & if\ q_i^{max} - q_i^{min} \le \varepsilon \end{cases} \quad (7)$$

where $q_i \in [q_p, q_{be}, q_{ps}, q_{pp}]$, and the superscripts max and min represent maximum and minimum respectively.

3 The Service Selection Method Based on Ant Colony Optimization

Ant Colony Optimization (ACO) [10,13] is a bionic probabilistic algorithm which take full advantage of the ant colony's intelligence to find the optimal path so that it can solve complex problems. This method has the characteristics of distributed computation, positive information feedback and heuristic search, and is essentially a heuristic global optimization algorithm in evolutionary algorithm. At present, ACO has been widely used in many fields, the most common of which is to solve the Traveling Salesman Problem (TSP) [13,14]. However, this method can be applied to the service selection problem and can rapidly converge to the global optimal solution by the deformation of it and with the optimization of its parameters.

The behavior of a single ant is extremely simple, but the colony of thousands ants possess great intelligence due to they use pheromones to transmit information. As shown in Fig. 2, there are three paths from the ant nest (gray circle) to the food (yellow pentagram). In the process of searching for food, the direction of moving is selected according to the concentration of pheromones, and the food can be found finally. At the beginning, the ants' moving paths were random (see Fig. 2(a)) since there were no pheromone on the ground. Ants constantly release pheromone that marks their path as they moving. As time goes on, several ants find food, and there are several routes from the nest to the food (see Fig. 2(b)). But the pheromones gradually evaporate over time. Besides, since the behavior of ants is randomly distributed, there are more ants in the short path than in the long path in unit time, so the concentration of pheromones left by ants in the short path is higher. This provides strong guidance for the ants behind, and more and more ants gather on the shortest path (see Fig. 2(c)). Therefore, this process achieves the selecting of shortest path or the so-called optimal path.

It can be mapped the IoT services selection to the ACO scenario, as shown in Fig. 3. All m IoT services form a set of services and each service represented by a yellow pentagram. When the user's request arrives, there is a path from the user's request to each service. We use the reciprocal of the corresponding QoE value of a service to indicate its path length, i.e. $d_i = \frac{1}{qoe_i} (i = 1, 2, ..., m)$, where

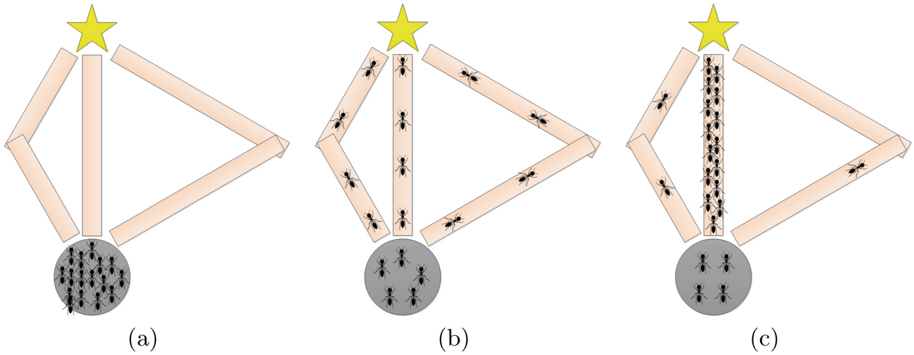


Fig. 2. The process by which ants search for food.

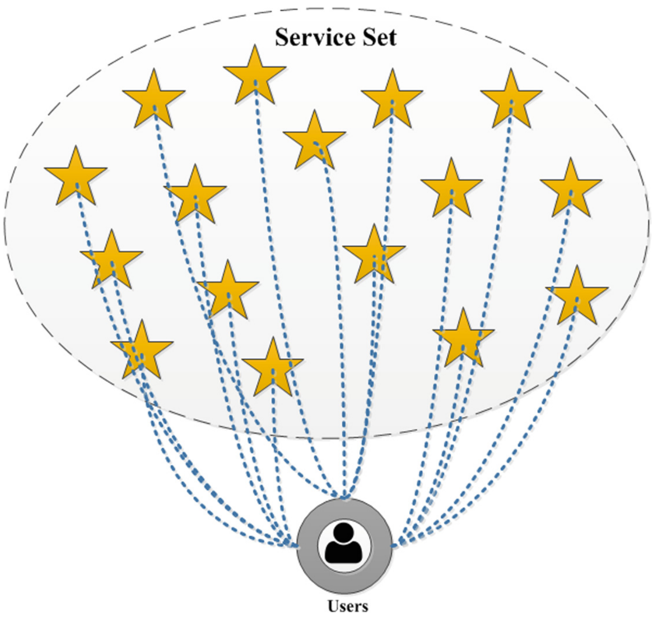


Fig. 3. The mapping from service selection problem to the ACO scenarios.

d_i is the distance from the nest to the i^{th} service. This is because the larger the QoE of a service is, the more the service meets the user's demands, which is similar to the shorter path from service to user.

Selecting a path from all paths by each ant, the pheromone concentration τ_i and some heuristic information η_i of this path should be taken into account at the same time in order to accelerate the convergence speed of algorithm.

The ants can choose each path by Roulette Wheel Selection (RWS) [15], where the probability of the j^{th} ant choosing its path is defined as follow.

$$p_i^j = \frac{\tau_i^\alpha \cdot \eta_i^\beta}{\sum_{k=0}^{m} \tau_k^\alpha \cdot \eta_k^\beta} \quad i \in \{1, 2, ..., m\} \tag{8}$$

where α and β control the relative importance of the pheromone versus the heuristic information, and

$$\eta_i = \frac{1}{d_i} = qoe_i \quad i \in \{1, 2, ..., m\} \tag{9}$$

Another important problem is the updating of pheromones. There are two main modes for pheromones updating. One is the global synchronous update, i.e. concentrate on updating all pheromones after all ants had selected the path. The other is local update, i.e. after each ant selects a path, it updates the pheromone of the current path immediately, and all pheromones of the selected path are updated until one iteration over. Of course, the latter one allows subsequent ants to perform different probabilistic paths, so that some do not make the same choices. But the former can be thought as all the ants are selected simultaneously, which can use the multi-threaded technology with a higher execution efficiency. After each iteration, the pheromones on each path should be reduced by a certain ratio, as shown in Formula (10). For the selected path, new pheromones will be left after the ant's pass, so the Formula (11) is used to represent the increasing in pheromone of the selected path.

$$\tau_i^{l+1} = (1 - \rho) \cdot \tau_i^l \quad i \in \{1, 2, ..., m\} \tag{10}$$

where ρ is the evaporation rate and l is the current iteration.

$$\tau_i^{l+1} = \tau_i^{l+1} + \frac{Q}{d_i} = \tau_i^{l+1} + Q \cdot qoe_i \quad i \in \{1, 2, ..., n\} \tag{11}$$

where Q is the pheromone increment constant.

In this way, the probability for selecting the path to high-quality service becomes larger, while the probability for selecting lower one becomes smaller, so that the first k IoT services that meet the requirements can be selected by using ordered pheromone concentration. The termination condition of the algorithm can be fixed iterations or the stagnant phenomenon appears (all ants choose the same path, and the solution will not change). The ACO-based IoT Service Selection algorithm (ACO-SS) is shown in Algorithm 1. The algorithm consists of four parts: the initialization in Line 1–2; the iteration in Line 3–18 includes the selection of service (path) by each ant and the update of pheromone; in the Line 19–21, the service set is sorted according to the pheromone matrix, and the first k services are taken as results.

Algorithm 1. ACO-based IoT Service Selection Algorithm

Input: Ant colony size n, the maximum iteration I, Service Set $\boldsymbol{S}$, Demand Service Number k, Evaporation rate ρ, Pheromone increment constant Q

Output: Demand Service Set $\boldsymbol{S_k}$

Initialize all elements in pheromone matrix $\boldsymbol{M_{ph}}$ as 1;
Compute the QoE matrix $\boldsymbol{S_{qoe}}$ for all the services;
while $i < I$ **do**
  $\boldsymbol{S_{select}} = \emptyset$;
  while $j < n$ **do**
    R_j =random$(0, 1)$;
    while $l < m$ **do**
      if $\frac{\sum_1^l M_{ph}^l \cdot S_{qoe}^l}{\boldsymbol{M_{ph}} \cdot \boldsymbol{S_{qoe}}} > R_j$ **then**
        $\boldsymbol{S_{select}} = \boldsymbol{S_{select}} \bigcup \{l\}$
        break;
      end
    end
  end
  $\boldsymbol{M_{ph}} = \boldsymbol{M_{ph}} \cdot (1 - \rho)$;
  foreach j *in* $\boldsymbol{S_{select}}$ **do**
    $M_{ph}^j = M_{ph}^j + Q \cdot S_{qoe}^j$;
  end
end
Sort $\boldsymbol{S}$ according to $\boldsymbol{M_{ph}}$;
Choose first k services in $\boldsymbol{S}$ as $\boldsymbol{S_k}$;
return $\boldsymbol{S_k}$

4 Simulations

4.1 Set up

The simulations in this section runs on a PC with win7 (64bit) operating system. The CPU is an i5 processor and the memory is 8GB. It is programmed to generate service set and to realize both algorithms with MATLAB2016Ra.

The simulation generates a set of 1000 services, and the attributes of each service are randomly produced. The value of fixed parameters in QoE model and the range of each service attribute are shown in Tables 1 and 2, respectively.

4.2 Comparison

In order to verify the actual effect of the proposed algorithm in solving the problem of service selection, we implemented another Service Selection algorithm based on Genetic Algorithm (GA-SS). GA-SS has changed the traditional

Table 1. The value of fixed parameter in QoE model.

Category	Parameter	Value
Performance	RT_{max}	1000
	θ_{RT}	0.3
	θ_A	0.4
	θ_R	0.3
Price sensitivity	S_1, S_2, S_3, S_4, S_5	5, 4, 3, 2, 1
	p_1, p_2, p_3, p_4	300, 800, 2000, 5000
Personal preference	α	40
QoE	$\theta_p, \theta_{be}, \theta_{ps}, \theta_{pp}$	0.25

Table 2. The value range of each service attribute.

Category	Attribute	Value range
Performance	RT(ms)	[1,1500]
	A	[1,100]
	R	[1,100]
Brand effect	BE	[0,99]
Price sensitivity	Price(USD)	[1,10000]
Personal preference	PP	[1,100]

GA to applicability of service selection. Its encoding adopts the form of service composition with the length $4 \times k_{max}$, where k_{max} is the maximum number of services required, and the population size is 100. The fitness function is calculated by Formula (6), the selection function adopts RWS, the crossover function uses one-point crossover and one-point mutation method was used for mutation function with the rate 0.2. At last, the number of iterations is 100 as well as in ACO-SS. Meanwhile, ACO-SS takes the following parameters: the ant colony size is 100; evaporation rate is 0.25 and Q is 1.5; both α and β are 1; the number of iterations is 100.

The precision rate and recall rate as the evaluation indexes defined in Formula (12) and (14). In the simulation, precision and recall are obtained by the mean of serval results.

$$precision = \frac{\sum s_i^s}{k} \times 100\% \quad i = (1, 2, ..., k) \tag{12}$$

where

$$s_i^s = \begin{cases} 1, & if\ qoe(s_i^s) \geq qoe_{th} \\ 0, & else \end{cases} \tag{13}$$

qoe_{th} is the QoE threshold of the service that meets the requirement.

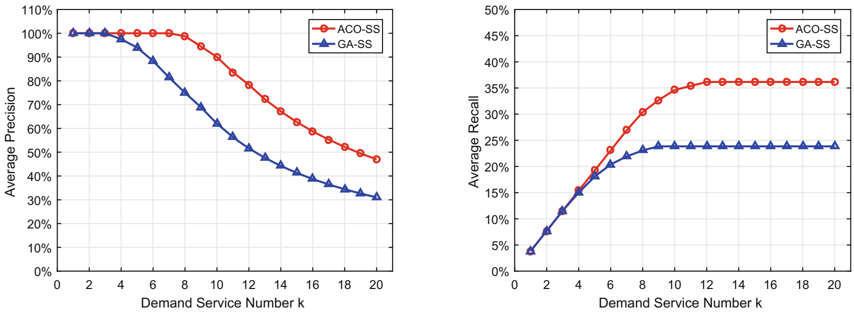
$$recall = \frac{\sum s_i^s}{\sum s_j} \times 100\% \quad i = (1, 2, ..., k), j = (1, 2, ..., n) \tag{14}$$

where

$$s_j = \begin{cases} 1, & if\ qoe(s_j) \geq qoe_{th} \\ 0, & else \end{cases} \tag{15}$$

n is the total number of services.

The experiment results are shown in Fig. 4. It can be clearly seen that both methods have close precision and recall when k is small, but the gap between them becomes larger with the increasing of k. In general, the precision and recall of ACO-SS are almost 10% higher than that of GA-SS on average.



(a) Mean Precision (b) Mean Recall

Fig. 4. Comparison of the mean precision and recall between two methods as the demand service number k increasing.

However, although adjusting the parameters can make both methods perform better in terms of precision and recall, it will significantly reduce the efficiency of them, which means users cannot get results within an acceptable time. By the analysis for both algorithms, it can be known that the algorithm efficiency of ACO-SS is mainly affected by the ants number and iterations. While for the GA-SS, the encoding length will also greatly affect its algorithm efficiency besides the population size and iterations due to the encoding length will directly affect the three operations: selection, crossover and mutation. As shown in Fig. 5, we compared the execution time of two methods with $k = 5$ as the number of iterations increasing. ACO-SS adopts the same parameters as above, while GA reduces the population size to 40 and the encoding length to $2 \times k$. It can be seen that the time of 100 iterations for both algorithms is less than 1s, which is within the acceptable range of users. In addition, the algorithm efficiency of ACO-SS is still several times higher than that of GA-SS, even if the GA's parameters are greatly reduced (its recall and precision are reduced accordingly).

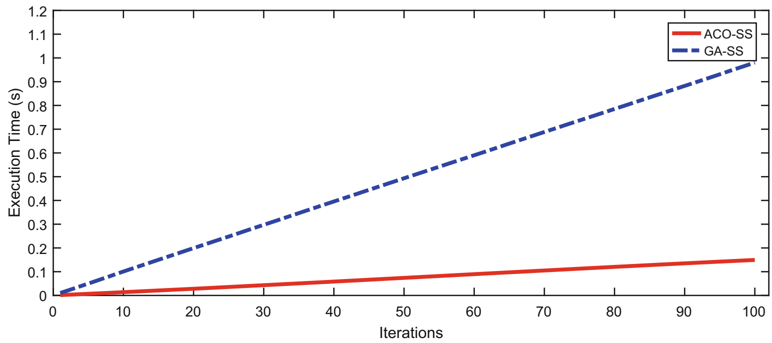


Fig. 5. Comparison of the average execution time between two methods with $k = 5$ as the iterations increasing.

5 Conclusions

In order to overcome the shortcomings of the existing service selection methods in IoT, a QoE-oriented multi-objective service selection algorithm based on ACO is proposed. By using the advantage of global optimization and fast convergence speed of the ACO, an optimal solution can be obtained quickly and efficiently. A multi-parameter linear weighted QoE quantitative model is constructed firstly, and the standardization methods are given. The model has a wide range of applicability and good scalability in IoT due to the lots of analysis for IoT services and QoE estimation methods. Based on this model, we described the ACO in detail and proposed the ACO-SS algorithm. Finally, the proposed method is validated by simulations and the results show that, compared with other algorithms such as GA, the proposed method can effectively improve the precision and recall for service selection in the same scenario with a far more higher computational efficiency and thus is a feasible and effective way to solve the issue of service selection in IoT. In the future, we will consider combing ACO-SS with other methods and conducting more extensive simulations to evaluate the efficiency and robustness of the proposed method on a larger scale data.

Acknowledgements. Thanks to the National Natural Science Foundation of China (Grants No. 41761086) for funding.

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Jain, R.: Quality of experience. Multimedia IEEE **11**(1), 95–96 (2014)
3. Jing, H., Wengdong, W.: A service implementation scenario measuring users' QoE. J. Beijing Univ. Posts Telecommun. **30**(2), 106–109 (2007)
4. Runqun, X., Junzhou, L., Aibo, S.: QoS preference-aware replica selection strategy in cloud computing. J. Commun. **32**(7), 93–102 (2011)

5. Herrera-Viedma, E., Chiclana, F., Herrera, F., et al.: Group decision-making model with incomplete fuzzy preference relations based on additive consistency. IEEE Trans. Cybern. **37**(1), 176–189 (2007)
6. Guohua, C.: Direct search algorithm to integer linear programming portfolio selection model. Commun. Appl. Math. Comput. **25**(1), 119–126 (2011)
7. Tantan, L., Ming, L.: Heuristic algorithm-based initiation method of probability hypothesis density filter for target tracking. J. Shanghai Jiaotong Univ. **52**(1), 63–69 (2018)
8. Jansen K., Rohwedder, L.: On Integer Programming and Convolution (2018)
9. Rojas, I., González, J., Pomares, H., et al.: Statistical analysis of the main parameters involved in the design of a genetic algorithm. Syst. Control Lett. **58**(9), 652–663 (2002)
10. Marco, D., Montes, O.M.A., Sabrina, O., Thomas, S.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2007)
11. Chunlin, L., Dan, L., Ling, X.: A service selection algorithm based on quantified QoE evaluation. Chin. J. Electron. **43**(11), 2145–2150 (2015)
12. Bing, J., Lifei, H., Chuxuan, Z., et al.: An IoT service aggregation method based on dynamic planning for QoE restraints. Mobile Netw. Appl. **24**, 25–33 (2018)
13. How does "ant colony algorithm" get the optimal solution from "more hands produce a stronger flame" (2018). https://blog.csdn.net/dog_bite_dog/article/details/79221048
14. Solnon, C.: Ant Colony Optimization and Constraint Programming. Wiley-IEEE Press, New York (2018)
15. Lipowski, A., Lipowska, D.: Roulette-wheel selection via stochastic acceptance. Physica A Stat. Mech. Appl. **391**(6), 2193–2196 (2011)