



# Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing

Yi Que<sup>1</sup> · Wei Zhong<sup>2</sup> · Hailin Chen<sup>1</sup> · Xinan Chen<sup>1</sup> · Xu Ji<sup>1</sup>

Received: 15 November 2017 / Accepted: 15 March 2018 / Published online: 26 March 2018  
© Springer-Verlag London Ltd., part of Springer Nature 2018

## Abstract

Developments in new information technology have indicated that single manufacturing services are now unable to satisfy users' multi-objective demands, especially in the process industry. As a new user-centric, service-oriented, demand-driven manufacturing model, cloud manufacturing can provide high-reliability, low-cost, fast-time, high-ability services. This study presents a new Manufacturers to Users (M2U) mode for cloud manufacturing, aiming at solving the core manufacturing service composition optimal selection (MSCOS) problem. The M2U mode expands the service areas and improves its dynamic optimal allocation capabilities of resources by efficient and flexible management and operation of services. Firstly, a comprehensive mathematical evaluation model with four critical quality of service (QoS)-aware indexes (time, reliability, cost, and ability) is constructed. Secondly, a new information entropy immune genetic algorithm (IEIGA) is proposed for the model solution. Finally, nine MSCOS problems of different scales are illustrated so as to compare the performance of the three algorithms. The results prove the effectiveness and superiority of the proposed algorithm and its suitability for solving large-scale service composition problems.

**Keywords** Cloud manufacturing · Quality of service · Service composition · Manufacturers to users · Immune genetic algorithm

## 1 Introduction

Cloud manufacturing was proposed and successfully developed with the rapid developments in information technology, such as the Internet of things, manufacturing technology, and cloud computing [1]. Cloud manufacturing, a type of intelligent manufacturing model, is able to handle more complex manufacturing requests to satisfy users' multi-objective demands than the traditional manufacturing modes. The concepts, developments, system structures, and main features for cloud manufacturing have been systematically explicated

in the literature [2–5]. However, to allow for expansion into process industries such as the petrochemical, steel, cement materials, and e-commerce industries, cloud manufacturing theory needs to be further extended to include large logistics, large-scale processes, and timely delivery.

In a market environment where information is widely collected, quickly disseminated, and analyzed, single manufacturing services have been unable to satisfy the increasingly complex multi-objective users' demands for high reliability, low cost, fast turnarounds, and advanced abilities. Traditional networked manufacturing, which usually provides immutable number of resources or services to customers, is inflexible and inefficient in data collection and model optimization, so the entire business process and business cooperation are relatively fixed and lack dynamic optimization capabilities. Under the circumstances, the escalation of the traditional process industries is needed. Manufacturers need to optimally allocate resources and strive to strengthen their core competencies, which requires the cooperation with external manufacturers and the satisfaction of the diverse and complicated demands from the market. Therefore, it is urgent to improve the manufacturing model by applying Manufacturers to Users (M2U) to cloud manufacturing. M2U cloud manufacturing

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s00170-018-1925-x>) contains supplementary material, which is available to authorized users.

---

✉ Xu Ji  
jxhpb@163.com

<sup>1</sup> College of Chemical Engineering, Sichuan University, No.24 South Section 1 Yihuan Road, Chengdu 610025, People's Republic of China

<sup>2</sup> China Construction West Construction Co Ltd, Chengdu 610065, People's Republic of China

expands the service areas and improves its dynamic optimal allocation capabilities of resources by efficient and flexible management and operation of services. Manufacturers to Consumers (M2C) business model means that manufacturers provide products and services directly to consumers, while M2U extends to all users, including the final consumers and also those manufacturing service providers. Figure 1 is the system framework of M2U-oriented cloud manufacturing, in which manufacturers are also consumers while providing services, and the cloud platform plays the key roles to improve the whole manufacturing capabilities, reliability, cost optimization, and delivery timeliness. In the above M2U mode, manufacturing resources consist of hardware resources, software resources, and human resources, and manufacturing capabilities contain production capability, safety management capability, etc.

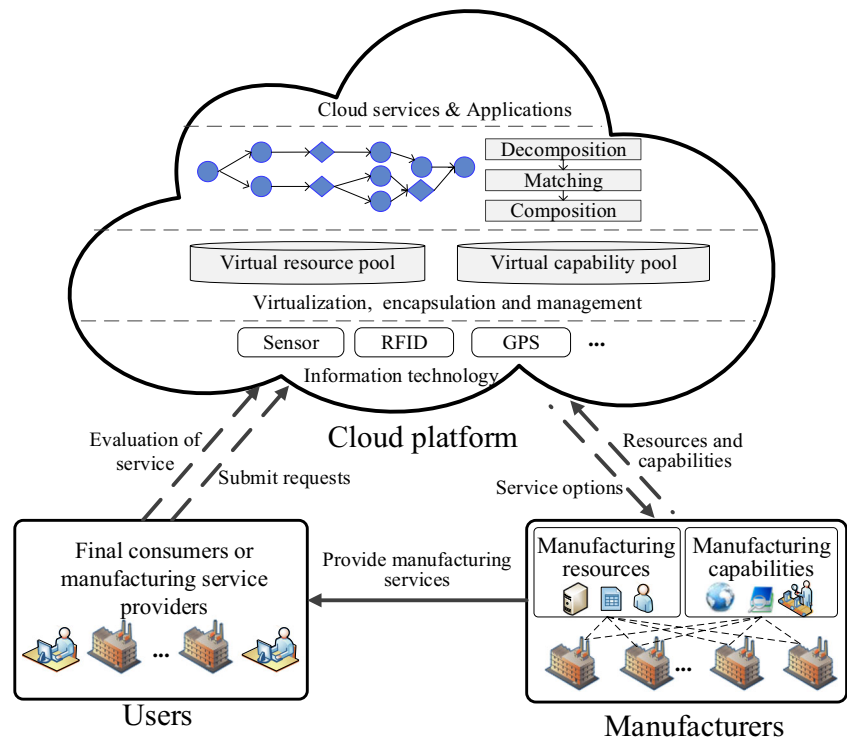
As shown in Fig. 1, cloud platform consists of four layers: sensing layer, virtual resource layer, functional layer, and service application layer. (1) On the sensing layer: large-scale distributed manufacturing resources and capabilities provided by manufacturers are sensed and interconnected by sensing devices and Internet, all of which form manufacturing services into a cloud platform. (2) On the virtual resource layer: the manufacturing resources and capabilities are virtualized and encapsulated in a virtual pool. (3) On the functional layer: complex manufacturing requests are analyzed, decomposed, and automatically matched with relevant cloud services from the cloud pool by specifically designed intelligent algorithms, and services are composed to complete the requests. (4) The

service application layer provides all kinds of services for users, such as design service, experiment service, and manufacture service. The general operation mode for M2U is that manufacturers provide a variety of online and offline manufacturing resources and capabilities to cloud platform, users submit various requests, the cloud platform evaluates users' requests and manufacturers' capabilities dynamically and organizes production and delivery based on multi-objective optimal service options considering the interests of both sides.

So, for M2U mode, it is the key point to select the optimal service solutions to satisfy users' requirements accurately and rapidly, which is called manufacturing service composition optimal selection (MSCOS) problem, and becomes a main focus in state-of-the-art research. Scholars have studied the service composition problem and proposed an evaluation model [6], and there still requires significant development in solving large-scale MSCOS problem because MSCOS for cloud manufacturing is large-scale, high-heterogeneity, strong-dynamics, and more complicated than previous service composition; therefore, it is essential to research the MSCOS problems in cloud manufacturing.

According to the special business patterns and management requirements of process industry enterprises, the focuses of this paper are the MSCOS for cloud manufacturing, especially the critical QoS indexes, considering users' preferences for cloud services, as well as the corresponding mathematical evaluation models for MSCOS. In this paper, an improved information entropy immune genetic algorithm (IEIGA) for

**Fig. 1** M2U-oriented cloud manufacturing system framework



MSCOS will be researched, which has better stability and solution quality than the unimproved algorithms.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 establishes a comprehensive mathematical evaluation model for the QoS composition, which covers the four attributes including time, reliability, cost, and ability. The adaptive immune genetic algorithm is presented in Sect. 4. In Sect. 5, a performance evaluation of the IEIGA in solving the MSCOS problem is conducted and the results compared with immune genetic algorithm (IGA) and standard genetic algorithm (SGA), the simulation results from which verify the feasibility and validity of the algorithm. Section 6 summarizes this paper and discusses possible future research directions.

## 2 Related works

In recent years, intelligent algorithms are introduced to service composition optimization problems because of their simplicity, versatility, and robustness in solving large-scale, complex service composition optimization problems for cloud manufacturing.

In reference [7], an ant colony optimization (ACO) algorithm was proposed to solve a multi-objective mathematical model of conflict resolution. To cope with the problem of fuzzy classification of manufacturing resources, a cooperative algorithm which combines simulated annealing and genetic algorithm was developed [8]. Lartigau et al. [9] proposed an adapted artificial bee colony (ABC) algorithm to ensure the veracity of a manufacturing time evaluation. A service composition optimization model based on a particle swarm optimization algorithm was proposed in references [10, 11], which provides the theoretical basis for resource service selections. In order to solve the composition problem and minimize the service level agreement (SLA) violations, a skyline-based genetic algorithm was adopted in reference [12]. In addition, genetic algorithm has been widely used because of their robustness, efficiency, practicality, and other significant features [13–15]. Some theoretical studies have proven that traditional simple genetic algorithm would find it difficult to converge to the global optimal solution, especially when solving complex optimization problems [16]. Under the circumstances, interaction and integration between different algorithms are needed to achieve better performances. Genetic algorithm has been improved to effective in several directions, which can be classified into three categories: adaptive improvements, global search improvement, and local search improvements [17, 18]. The most widely used adaptive improvement is the dynamic parameter improvements developed by M Srinivas [19], while the global search improvements are niche strategy [20], mean strategy [21], and chaos strategy [22]. The most representative local search improvement is immune genetic

strategy [23]. IGA is a promising improved algorithm because of its better global exploration performance and strong stability to solve large-scale manufacturing service composition problems [24]. However, IGA with deterministic crossover probability and mutation probability may get trapped into local optimum. To improve IGA's performances, information entropy is used to represent the population similarity [25], which allows for adaptive adjustments in the crossover and mutation probabilities based on the actual population. The improved IGA is developed as a new hybrid algorithm; it can significantly improve the performances and overcome the drawbacks of SGA.

In a word, for the sake of solving the MSCOS problem of cloud manufacturing, an improved IEIGA is presented by integrating the adaptive genetic algorithm with immune optimization algorithm.

## 3 MSCOS evaluation model

### 3.1 QoS attribute parameters

Assume that the cloud platform receives a set of service requests (denoted by  $R$ ) from users, then  $R = \{R_1, R_2, R_3, \dots, R_i\}$ , where any one of the user's request  $R_i (i = 1, 2, 3, \dots, I)$  has corresponding service quality requirements, indicated by  $Q(R_i)$ . There have been many significant studies conducted on the selection of appropriate QoS parameters [26–30]; therefore, to better implement the algorithm and ensure its practical measurability, the key QoS attributes are discussed firstly. The key elements to increase users' loyalty are time, cost, reliability, and manufacturing ability. The four QoS indexes are therefore labeled time, reliability, cost, and ability, with the user's QoS request being  $q^4(R_i)$   $Q(R_i) = \{q^1(R_i), q^2(R_i), q^3(R_i), q^4(R_i)\}$ . Descriptions for these four attributes are shown in Table 1.

### 3.2 Comprehensive mathematical model

In general, there are multiple service composition options in the cloud resource pool. Any request submitted by the user on

**Table 1** Typical QoS attributes of services

QoS attribute	Description
Time	Service execution duration.
Reliability	The execution reliability of the requested product or service.
Cost	The cost of product or service.
Ability	Comprehensive ability of the manufacturers, including production capability, safety management capability, technological research capability.

the cloud platform can be decomposed into several subtasks (denoted by  $ST$ ). Let  $R_i$ , which can be decomposed into  $j$  subtasks, denotes a user’s request for a service.  $R_i$  can be expressed as follows:  $R_i = \{ST_i^1, ST_i^2, ST_i^3, \dots, ST_i^j\}$ , where  $j$  is the total number of subtasks.

Each decomposed  $ST$  is matched with a cloud service subtask (denoted by  $CSS$ ) in a virtual cloud resource pool, with the cloud service (denoted by  $CS$ ) expressed as follows:  $CS_i = \{CSS_i^1, CSS_i^2, CSS_i^3, \dots, CSS_i^j\}$ . Let  $CCS$  denote the cloud candidate services for each  $CSS$ , the corresponding  $CCS_i^j = \{CCS_{i1}^j, CCS_{i2}^j, CCS_{i3}^j, \dots, CCS_{ik}^j\}$ , where  $j$  is the total number of subtasks, and  $k$  is the total number of candidates for each  $CSS$ .

The service composition process selects one candidate  $CCS_k^j$  for each  $CSS$  according to specific requirements; the selected services are constructed to complete the submitted request. The best solution for the user’s QoS requirements is selected by intelligent algorithms. The MSCOS process discussed above is presented in Fig. 2.

Based on the index system, the MSCOS evaluation system is established. Each index has a corresponding weight:  $w_1$  is the weight for time,  $w_2$  is the weight for reliability,  $w_3$  is the weight for cost, and  $w_4$  is the weight for ability, with the four weights being equal to 1.

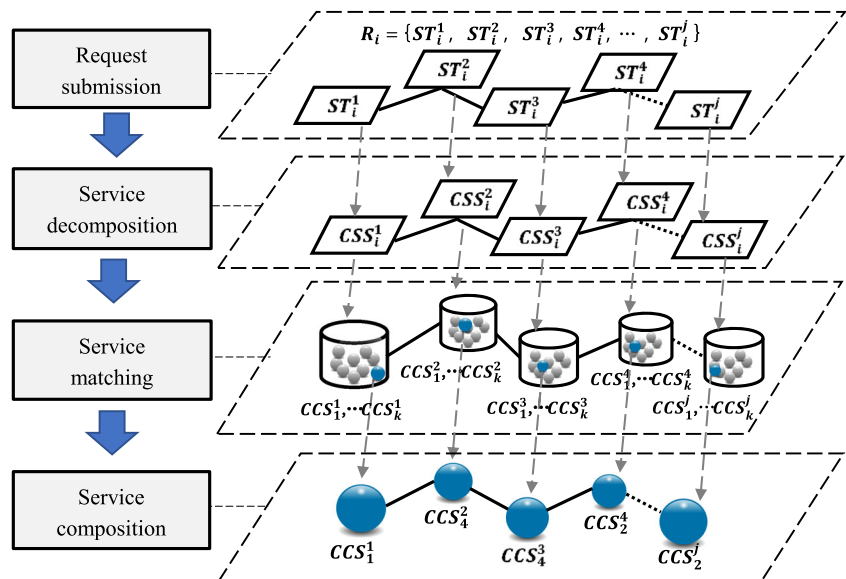
There are four main types of service compositions: sequential, parallel, selective, and circular, of which three (parallel, selective, and circular) can be converted into sequential structures using mature modern technology [27]. According to Tao et al. [31] and Liu et al. [32], the QoS aggregation functions are illustrated in Table 2. In the table below, time and cost represent cumulative addition indices from each subtask, ability is a minimum index value, meaning that the manufacturing processes need to have a minimum value, and reliability is a multiplication index.

To eliminate the influences of the dimension and the value size, data normalization is employed in the MSCOS model. A unified quantization method is used to solve this problem [26, 33, 34], with the QoS attributes being normalized as follows:

$$\tilde{q}^n(CSS_i) = \begin{cases} \frac{\max(q^n(CSS_i)) - q^n(CSS_i)}{\max(q^n(CSS_i)) - \min(q^n(CSS_i))}, & \max(q^n(CSS_i)) \neq \min(q^n(CSS_i)) \\ 1 & , \max(q^n(CSS_i)) = \min(q^n(CSS_i)) \end{cases} \tag{1}$$

$$\tilde{q}^n(CSS_i) = \begin{cases} \frac{q^n(CSS_i) - \min(q^n(CSS_i))}{\max(q^n(CSS_i)) - \min(q^n(CSS_i))}, & \max(q^n(CSS_i)) \neq \min(q^n(CSS_i)) \\ 1 & , \max(q^n(CSS_i)) = \min(q^n(CSS_i)) \end{cases} \tag{2}$$

Fig. 2 Manufacturing service composition optimal selection (MSCOS) process



**Table 2** QoS aggregation functions

QoS attribute	Function
Time	$q^1(CSS_i) = \sum_{j=1}^J q^1(CCS_k^j)$
Reliability	$q^2(CSS_i) = \prod_{j=1}^J q^2(CCS_k^j)$
Cost	$q^3(CSS_i) = \sum_{j=1}^J q^3(CCS_k^j)$
Ability	$q^4(CSS_i) = \min_{j=1}^J (q^4(CCS_k^j))$

The QoS attributes are divided into positive and negative categories. Time and cost are considered negative properties, that is, the higher the time and cost values, the lower the QoS evaluation, so  $q^1(CSS_i)$  and  $q^3(CSS_i)$  are calculated using Eq. (1). Reliability and ability are considered positive properties, that is, the higher the reliability and ability values, the better the QoS evaluation, so  $q^2(CSS_i)$  and  $q^4(CSS_i)$  are calculated using Eq. (2).

Based on previous studies [34–37], the simple additive weighting (SAW) is used to aggregate different objectives into a comprehensive evaluation function as follows:

$$\tilde{Q}(CSS_i) = \tilde{q}^1(CSS_i) \times w_1 + \tilde{q}^2(CSS_i) \times w_2 + \tilde{q}^3(CSS_i) \times w_3 + \tilde{q}^4(CSS_i) \times w_4 \tag{3}$$

As the maximum value for the evaluation function can be obtained using algorithms, manufacturers can organize production planning based on the optimization scheme, which is determined using Eq. (4):

$$\max \tilde{Q}(CSS_i) \tag{4}$$

From the user’s QoS requirements for the four service evaluation attributes, the selection service should satisfy all constraints. If any one of the four conditions is not met, there is a service composition failure. The constraints are shown in Eq. (5).

$$\begin{cases} q^1(R_i) \geq q^1(CSS_i) \\ q^2(R_i) \leq q^2(CSS_i) \\ q^3(R_i) \geq q^3(CSS_i) \\ q^4(R_i) \leq q^4(CSS_i) \end{cases} \tag{5}$$

### 4 Intelligent algorithms for MSCOS

IGA is an improved genetic algorithm based on a biological immune mechanism that is a combination of life science immune principles and SGA [38]. Therefore, it also includes

some typical genetic algorithm operators such as crossover and mutation. Due to the deterministic crossover and mutation probabilities of the genetic algorithm, the evolutionary search process is slow, which means that stagnation and some other shortcomings may appear in IGA.

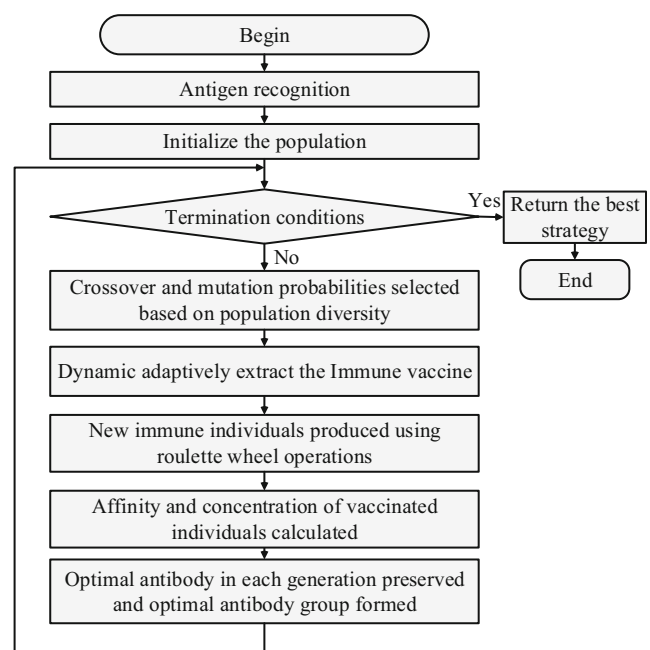
To improve the performances of IGA, information entropy is used to represent the population similarity [25], which allows for adaptive adjustments in the crossover and mutation probabilities based on the actual population. In this paper, a proposed IEIGA is used to solve the MSCOS problem. The flow chart of IEIGA is shown in Fig. 3.

#### 4.1 Initialization

An appropriate coding structure is essential to determine the problem solution. Real number coding is used to solve the MSCOS problem, as it can reduce algorithmic complexity and improve accuracy [36]. The sequence numbers for the gene bits indicate the CSS serial numbers, and the gene value at each locus represents the CCS index. The encoding is shown in Fig. 4.

#### 4.2 Fitness function

In each generation, the genome is evaluated based on the initial objective function. Let  $X_i$  represent the solution to the MSCOS problem, based on Eq. (3) in the  $\tilde{Q}(CSS_i)$  formula, the fitness function is defined as follows:



**Fig. 3** IEIGA process



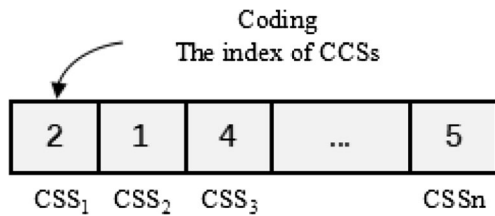


Fig. 4 MSCOS real number coding

$$Fitness(X_i) = \begin{cases} \tilde{Q}(CSS_i)_{X_i} & \\ \tilde{Q}(CSS_i)_{X_i} - 2, \text{ if } q^1(R_i) < q^1(CSS_i) & \\ \tilde{Q}(CSS_i)_{X_i} - 2, \text{ if } q^2(R_i) > q^2(CSS_i) & \\ \tilde{Q}(CSS_i)_{X_i} - 2, \text{ if } q^3(R_i) < q^3(CSS_i) & \\ \tilde{Q}(CSS_i)_{X_i} - 2, \text{ if } q^4(R_i) > q^4(CSS_i) & \end{cases} \quad (6)$$

When the constraints are satisfied in the above expression,  $\tilde{Q}(CSS_i)$  is used as the evaluation formula, that is, when any one of the conditions submitted by the user is not satisfied, there is a service composition failure.

### 4.3 Adaptive crossover and mutation

Let  $H_i(N)$  represent the information entropy for gene  $j$ , and  $P_{ij}$  be the probability that the number  $i$  appears on the locus. Then, the average information population entropy can be represented as follows:

$$H(N) = \frac{1}{L} \sum_{j=0}^{L-1} H_j(N) \quad (7)$$

$$H_{ij}(N) = \sum_{n=0}^{(S-1)} -P_{ij} \log_2 P_{ij} \quad (8)$$

The population similarity can be calculated as follows:

$$A(N) = \frac{1}{1 + H(N)} \quad (9)$$

$A(N)$  shows the overall population similarities and varies between 0 and 1. If the  $A(N)$  value is larger, the population similarity is higher, and when  $A(N) = 1$ , the individual antibodies in the population are identical.

The crossover probability  $P_c$  and the mutation probability  $P_m$  are two important parameters for IGA. If they are too large or too small, the algorithmic convergence rate is directly affected. The adaptive strategy can automatically change the population diversity of  $P_c$  and  $P_m$ , which are adjusted using the following formulas.

$$P_c = e^{2(A(N)-1)} \quad (10)$$

$$P_m = 0.1e^{2(A(N)-1)} \quad (11)$$

### 4.4 Vaccine extraction

The immune vaccine is dynamically and adaptively extracted to ensure overall vaccine effectiveness and the optimal  $K_b$  antibody in the population conserved. As the optimal antibody group has  $K^*K_b$  antibodies in the  $K$ th generation, each gene for each antibody has  $k_1, k_2, \dots, k_s$ , for which a total of  $s$  symbols can be selected. The probability of the  $i$ th allele  $k_i$  can be expressed as follows:

$$P_{ij} = \frac{1}{K^*K_b} \sum_{j=1}^{K^*K_b} a_j \quad (12)$$

When  $g(i) = k_i, a_j = 1$ , otherwise  $a_j = 0$ , where  $g(i)$  represents the  $i$ th allele in the population. The  $k_i$  with the greatest probability on the allele is used as the vaccine fragment, and finally, the vaccine is extracted as  $H = (h_1, h_2, h_3, \dots, h_L)$ .

### 4.5 Vaccination

A superior population is determined by selecting the antibody to be inoculated from the parent population and selecting one or more gene fragments using roulette wheel selection to generate new immune individuals by replacing the gene code values. From the above extracted vaccine  $H = (h_1, h_2, h_3, \dots, h_L)$ , the gene fragment is selected as follows:

$$q_i = \frac{P_i}{\sum_{j=1}^L P_j} (i = 1, 2, 3, \dots, L) \quad (13)$$

where  $q_i$  corresponds to  $h_i$ . The roulette wheel principle determines the probability that each vaccine fragment is selected to determine the final value for  $q_i$ .

### 4.6 Simulation procedures

The above definitions and improvements can effectively improve the performances of AIGA. The algorithm has been called in the computer integrated process system (CIPS), and the detailed simulation procedures are shown as follows:

- Step 1: The objective function is the basis of antigen recognition. The optimization goal  $\tilde{Q}(CSS_i)$  is used as immune genetic antigens.
- Step 2: All the initial parameters are set including population size, maximum iterations, diversity parameter, etc. Besides, the initial population is generated randomly.
- Step 3: The fitness values of antibodies is evaluated and optimal solution is obtained. The fitness is calculated using Eq. (6).

- Step 4: The current generation greater or equal to maximum iterations is used as the termination condition. If the terminal condition is satisfied, then the optimal solution is output, otherwise, turn to step 5 until terminal condition is satisfied.
- Step 5: The crossover probability  $P_c$  and the mutation probability  $P_m$  can be adjusted automatically according to Eqs. (7) to (11), and the genetic operations are performed.
- Step 6: Vaccines are extracted dynamically by using Eq. (12) before vaccination.
- Step 7: The antibody to be vaccinated is selected from the parental population, one or more gene fragments is selected by roulette wheel operations according to Eq. (13), new immune individuals is generated by replacing the gene code, and a better population is formed.
- Step 8: The affinity and concentration of each vaccinated antibody are calculated and the population is then updated by the immune selection operation.
- Step 9: The optimal affinity of  $K_b$  antibodies is preserved into the memory unit, while the low affinity of antibodies is replaced, and then turn to step 4 to judge whether the terminal condition is satisfied.

### 5 Case study

To verify the validity of the model and algorithm, experiments are designed based on a set of simulated data. There will exist a variety of candidate services with different QoS properties for each subtask. Each subtask can be a designing process, an experimental process, a manufacturing resource, a logistic resource, etc. Time, reliability, cost, and ability are used as the QoS attributes; the details of the four QoS attributes of a CSC are shown in Table 1. The QoS dataset is generated based on practical manufacturer situation of process industries. The time and cost of each subtask are assigned in different ranges, and the reliability is randomly generated with a uniform distribution between 0.9 and 1. In the practical performance of enterprises, the production capability of manufacturers is a very important indicator; thus, the production capability is used to represent ability, and the ability of each candidate is between 100 and 150]. Each aggregative QoS value of the composition candidates is normalized to the same interval [0, 1]. The QoS information for the service candidates is shown in Table 3 .

This paper applies the M2U cloud manufacturing model to simulate the experiments, for which three algorithms are used to solve the MSCOS problem. The comparison of three algorithms can testify whether the improvement strategies can achieve better solutions. These experiments are performed

**Table 3** Service candidate list

Subtask	Candidates	Time	Reliability	Cost	Ability
$ST^1$	$CCS_1^1$	3	0.96	173	120
	$CCS_2^1$	2	0.95	165	140
	$CCS_3^1$	4	0.97	195	125
...	...	...	...	...	...
$ST^2$	$CCS_1^2$	3	0.96	58	110
	$CCS_2^2$	3	0.94	54	120
	$CCS_3^2$	6	0.97	58	125
...	...	...	...	...	...
$ST^3$	$CCS_1^3$	12	0.95	51	120
	$CCS_2^3$	15	0.94	56	110
	$CCS_3^3$	10	0.92	55	115
...	...	...	...	...	...
$ST^4$	$CCS_1^4$	95	0.98	28	105
	$CCS_2^4$	85	0.95	30	110
	$CCS_3^4$	55	0.94	29	100
...	...	...	...	...	...

using MATLAB R2012b on a 32-bit Windows 7 operating system, on a 2.40-GHz PC with 4-GB RAM.

### 5.1 Parameters

$\tilde{Q}(CSS_i)$  is the randomly generated QoS data for each request,  $Q(R_i)$  is the actual user’s requirements for service request  $R_i$ , and the user’s requirements are the constraints in the evaluation function. The maximum value for  $\tilde{Q}(CSS_i)$  is the output from the MSCOS model.

In the algorithms, the initial population size is 20 and the maximum number of iterations is 400. The experimental results for each test case are the average value of 100 running times to ensure validity. The special parameters for the three algorithms are set as Table 4 .

**Table 4** The settings of algorithm parameters

Algorithm	Parameters
SGA	With reference to [36], The crossover probability is 0.8, and the mutation probability is 0.15. A random selection mechanism is applied.
IGA	As the same as SGA, the crossover probability is 0.8, and mutation probability is 0.15. Then the diversity parameter is set to be 0.95.
IEIGA	Since the crossover probability $P_c$ and the mutation probability $P_m$ can be adjusted adaptively to suit different phases of the search process, the diversity parameter left to be set as $P_s = 0.95$ .

To satisfy all users’ requirements, the comprehensive evaluation weight should have both subjectivity and objectivity. A combination assigning method is used to consider the preferences and requirements of different users and emulate the actual situation. The subjective weight is determined based on the analytic hierarchy process (AHP) and the objective weight is obtained using the variation coefficient method.

AHP is a kind of subjective weighting method, in which the various factors in complex problems are divided into an orderly hierarchy of interconnections to ensure rationality. It has a wide range of applications, including the field of evaluation, resource allocation, etc [39, 40]. As this method relies on the subjective judgments of experts, it may influence the accuracy of the evaluation results.

The variation coefficient method is an objective weight method that uses the information in each index to calculate its weight. The weight of the evaluation system is then determined by combining the variations in the coefficient method with the analytic hierarchy process, all of which ensures that the evaluation results are more accurate and practical [41].

The combined weight coefficient is calculated using the following formula:

$$w_j = \lambda A_j + (1-\lambda)V_j \tag{14}$$

where  $w_j$  is the combined weight coefficient of the  $j$ th index,  $A_j$  and  $V_j$  are the weight coefficients of the  $j$ th index, respectively, obtained using the AHP and the variation coefficient method, and  $\lambda$  is the preference coefficient,  $\lambda \in (0, 1)$ .

Depending on the service request preference degree, the weight vector obtained from the AHP can be expressed as  $A = [0.2210, 0.3280, 0.1860, 0.2650]$ . The data for the time, reliability, cost, and ability indexes are analyzed using the variation coefficient method, with the weight vector being  $V = [0.5350, 0.0129, 0.3989, 0.0532]$ . In this paper,  $\lambda = 0.5$ , and therefore, by using Eq. (14), the combined weight coefficient is  $W = [0.3780, 0.1704, 0.2925, 0.1591]$ .

### 5.2 Performance of the three algorithms

Three algorithms (SGA, IGA, IEIGA) are tested on nine different scales as shown in Table 5, where  $M$  represents the number of decomposed subtasks, and  $N$  represents the number of cloud service candidates.

**Table 5** Nine different MSCOS scales

Test case	NO.1	NO.2	NO.3	NO.4	NO.5	NO.6	NO.7	NO.8	NO.9
M	5	5	5	10	10	10	15	15	15
N	50	100	150	100	200	300	150	300	450

The output curve for the case is shown in Fig. 5, where  $M = 5$  and  $N = 50$ . From the analysis of the performance of the three algorithms, the following is concluded.

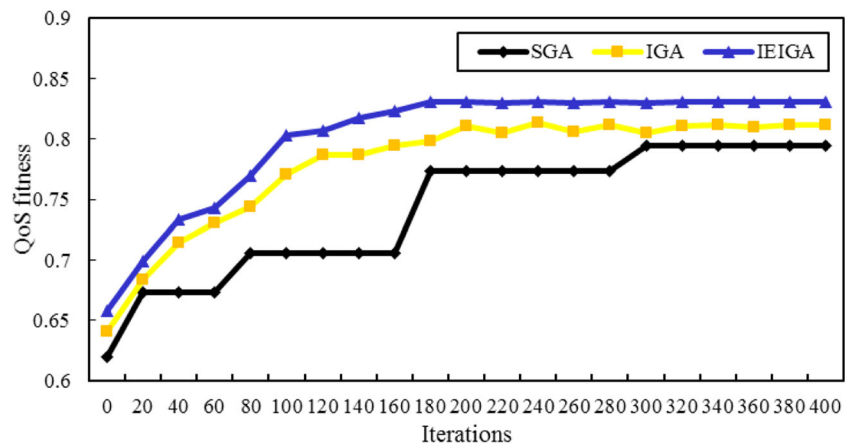
- (1) From the curve results, the IEIGA QoS fitness is close to 0.8311, the IGA fitness is 0.8116, and the SGA fitness is 0.7948; therefore, the QoS value obtained using the IGA is obviously higher than that of the SGA. Compared with the IGA and SGA, the IEIGA performs best, with the ranking of the three algorithms from low to high being  $SGA < IGA < IEIGA$ .
- (2) In terms of stability, SGA is found to easily fall into the local optimal solution and the IGA solution process is lengthy and unstable; however, as the IEIGA has a stable solution process, it is concluded that the  $SGA < IGA < IEIGA$ .
- (3) The convergence rate for the three algorithms is significantly higher for the IEIGA than for the IGA or SGA. The IEIGA is able to adaptively select the crossover and mutation probabilities in the evolutionary process, and the dynamic vaccine extraction improvement strategy effectively improves the convergence speed.

Figure 6 shows the QoS values for the optimal solutions obtained by the three algorithms over nine scales. The statistical results indicate that the IEIGA solution is superior to the other two algorithms in all scales. With an increase in candidate services, the results for all three algorithms improve because new and better services increase QoS fitness. However, with an increase in the number of subtasks, the QoS fitness is found to decrease because the quality assurance index in the QoS aggregation function (as shown in Table 2) has a multiplication effect. Therefore, the QoS IEIGA and IGA fitness values are similar for small-scale problems as the values of the two algorithms are basically the same; however, as the subtask scale expanded, the differences between the three algorithms are clear, with the performance improvement of IEIGA being more significant. The results, therefore, demonstrate the effectiveness of the improved strategy and the superiority of IEIGA in solving a large-scale MSCOS model.

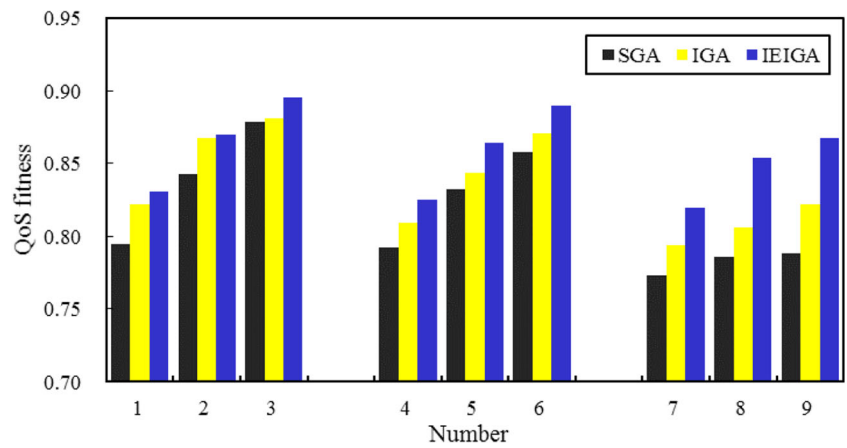
In the stability evaluation for the three algorithms on the nine scales (Fig. 7), the IEIGA standard deviation is the lowest, indicating that the IEIGA has the strongest stability. As the



**Fig. 5** Fitness curves for the three algorithms under 5 subtasks and 50 candidates



**Fig. 6** QoS fitness comparisons for the nine problem scales

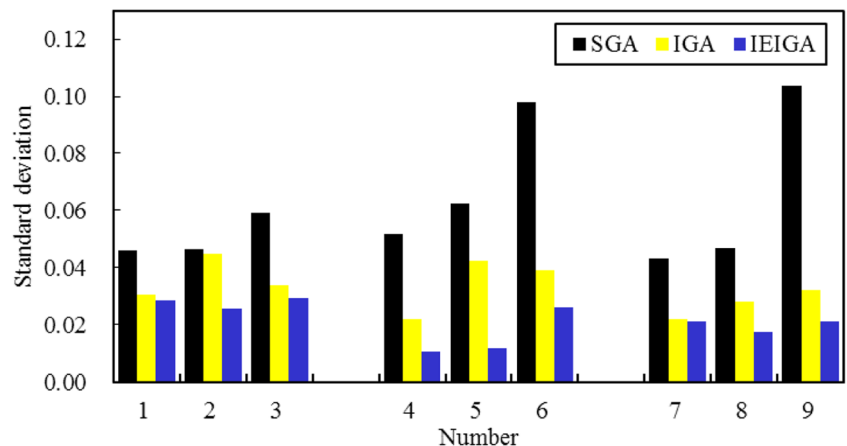


number of candidate services and subtasks increased, the standard deviation of SGA rises significantly, while that of the IEIGA remains at around 0.02, indicating that the SGA algorithm is not suitable for solving large-scale service composition optimization problems.

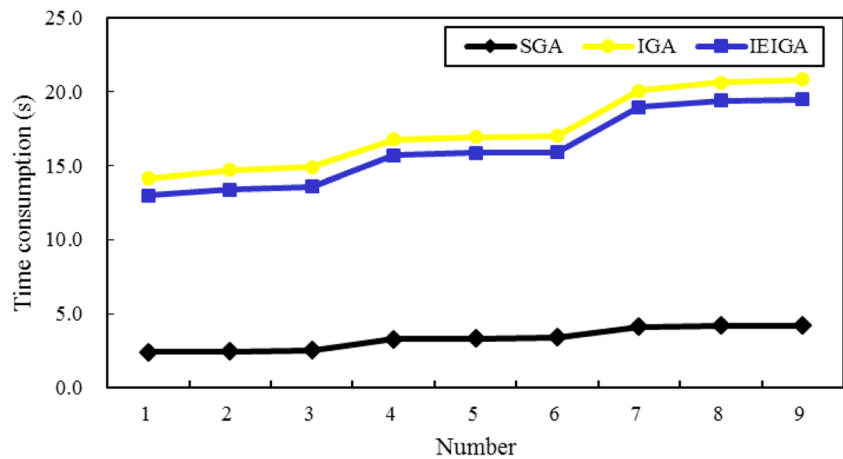
The average times taken for the three algorithms are shown in Fig. 8. It can be clearly seen that the variation trends for the three algorithms are similar, and time taken by all three algorithms changes minimally with a growth in the candidate

numbers  $N$ . However, as the subtask number  $M$  increases, the time taken also increases, because the QoS aggregation function becomes more complicated. In contrast, an increase in candidate number  $N$  does not affect the QoS aggregation function calculation, as the increase only indicates a rise in the service selection range. Overall, the time taken by the SGA is shorter because it is simpler than either the IEIGA or IGA; however, the time taken by the improved IEIGA with an adaptive strategy is shorter than for the IGA.

**Fig. 7** Standard deviation comparison for the nine problem scales



**Fig. 8** Time comparison for the nine problem scales



In general, the proposed IEIGA integrates a genetic algorithm with an immune algorithm shows the best stability and solution quality and significantly improves the solution process for large-scale service composition optimization problems.

## 6 Conclusions and future work

Single manufacturing services have been unable to satisfy the increasingly complex multi-objective users' demands; therefore, the escalation of the traditional process industries is needed. As a new user-centric, service-oriented, demand-driven manufacturing model, cloud manufacturing can provide high-reliability low-cost, fast-time, high-ability services. This study presents a new M2U model for cloud manufacturing in process industries such as the petrochemical, steel, cement materials, and e-commerce industries. The comprehensive mathematical model has four critical users' preference QoS-aware indexes (time, reliability, cost, and ability), with the weight of each attribute determined using a combination assignment method. To solve the core MSCOS problem in M2U cloud manufacturing models, this paper proposes a new adaptive IEIGA. Nine different MSCOS problem scales are applied to compare the performance of the three algorithms (SGA, IGA, IEIGA); the experimental results show that the proposed method is able to achieve good stability and high-quality solutions for large-scale MSCOS problems, thereby verifying its effectiveness.

There are two main future research directions: designing more effective algorithms for multi-objective service selection optimization problems and simultaneously considering the interests and satisfaction of both the manufacturers and users.

**Acknowledgements** The authors would like to thank Li Zhou and Yu Du for their constructive suggestions on the improvements of the service model, and gratefully acknowledge the support of the National Natural Science Foundation, China (No. 51473102, No. 21776183).

## References

1. Tao F, Zhang L, Venkatesh VC, Luo Y, Cheng Y (2011) Cloud manufacturing: a computing and service-oriented manufacturing model. *Proc Inst Mech Eng B J Eng* 225(10):1969–1976
2. Tao F, Cheng Y, Zhang L, Nee AYC (2017) Advanced manufacturing systems: socialization characteristics and trends. *J Intell Manuf* 28(5):1079–1094
3. Tao F, Zhang L, Liu YK, Cheng Y, Wang LH, Xu X (2015) Manufacturing service management in cloud manufacturing: overview and future research directions. *J Manuf Sci Eng Trans ASME* 137(4):040912
4. Zhang L, Luo YL, Tao F, Li BH, Ren L, Zhang XS, Guo H, Cheng Y, Hu AR, Liu YK (2014) Cloud manufacturing: a new manufacturing paradigm. *Enterp Inf Syst* 8(2):167–187
5. Ren L, Zhang L, Wang LH, Tao F, Chai XD (2017) Cloud manufacturing: key characteristics and applications. *Int J Comput Integr Manuf* 30(6):501–515
6. Tao F, Zhao DM, Hu Y, Zhou Z (2008) Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. *IEEE Trans Industr Inform* 4(4):315–327
7. Huang XR, Du BG, Sun LB, Chen F, Dai W (2016) Service requirement conflict resolution based on ant colony optimization in group-enterprises-oriented cloud manufacturing. *Int J Adv Manuf Technol* 84(1–4):183–196
8. Hu YJ, Chang XF, Wang Y, Wang ZL, Shi C, Wu LZ (2016) Cloud manufacturing resources fuzzy classification based on genetic simulated annealing algorithm. *Mater Manuf Process* 32(10):1109–1115
9. Lartigau J, Xu XF, Nie LS, Zhan DC (2015) Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved artificial bee Colony optimisation algorithm. *Int J Prod Res* 53(14):4380–4404
10. Fan XQ (2013) A decision-making method for personalized composite service. *Expert Syst Appl* 40(15):5804–5810
11. Zheng H, Feng YX, Tan JR (2016) A fuzzy qos-aware resource service selection considering design preference in cloud manufacturing system. *Int J Adv Manuf Technol* 84(1–4):371–379
12. Wang DD, Yang Y, Ming ZQ (2015) A genetic-based approach to web service composition in geo-distributed cloud environment. *Comput Electr Eng* 43:129–141
13. Wang D, Shao XD, Liu SM (2017) Assembly sequence planning for reflector panels based on genetic algorithm and ant Colony optimization. *Int J Adv Manuf Technol* 91(1–4):987–997
14. Lin YK, Chong CS (2017) Fast GA-based project scheduling for computing resources allocation in a cloud manufacturing system. *J Intell Manuf* 28(5):1189–1201

15. Keane AJ (1995) Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness. *Artif Intell Eng* 9(2):75–83
16. Jiang H, Yi JJ, Chen SL, Zhu XM (2016) A multi-objective algorithm for task scheduling and resource allocation in cloud-based disassembly. *J Manuf Syst* 41:239–255
17. Horton P, Jaboyedoff M, Oblad C (2017) Global optimization of an analog method by means of genetic algorithms. *Mon Weather Rev* 145(4):1275–1294
18. Okada I, Takahashi K, Zhang WQ, Zhang XF, Yang HY, Fujimura SR (2014) A genetic algorithm with local search using activity list characteristics for solving resource-constrained project scheduling problem with multiple modes. *IEEJ Trans Electr Electron Eng* 9(2): 190–199
19. Srinivas M, Patnaik LM (2002) Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans Syst Man Cybern* 24(4):656–667
20. Luo QK, Wu JF, Sun XM, Yang Y, Wu JC (2012) Optimal design of groundwater remediation systems using a multi-objective fast harmony search algorithm. *Hydrogeol J* 20(8):1497–1510
21. Ye ZS, Li ZZ, Xie M (2010) Some improvements on adaptive genetic algorithms for reliability-related applications. *Reliab Eng Syst Saf* 95(2):120–126
22. Huang BQ, Li CH, Tao F (2014) A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system. *Enterp Inf Syst* 8(4):445–463
23. Zinflou A, Gagné C, Gravel M (2012) GISMOO: a new hybrid genetic/immune strategy for multiple-objective optimization. *Comput Oper Res* 39(9):1951–1968
24. Xu XD, Li CX (2007) Research on immune genetic algorithm for solving the job-shop scheduling problem. *Int J Adv Manuf Technol* 34(7–8):783–789
25. Li ZF, Gu JF, Zhuang HY, Kang L, Zhao XY, Guo Q (2015) Adaptive molecular docking method based on information entropy genetic algorithm. *Appl Soft Comput* 26:299–302
26. Xiang F, Hu YF, Yu YR, Wu HC (2014) Qos and energy consumption aware service composition and optimal-selection based on pareto group leader algorithm in cloud manufacturing system. *Cent Europ J Oper Res* 22(4):663–685
27. Zhou JJ, Yao XF (2017) De-caabc: differential evolution enhanced context-aware artificial bee colony algorithm for service composition and optimal selection in cloud manufacturing. *Int J Adv Manuf Technol* 90(1–4):1085–1103
28. Liu ZZ, Song C, Chu DH, Hou ZW, Peng WP (2017) An approach for multipath cloud manufacturing services dynamic composition. *Int J Intell Syst* 32:371–393
29. Chen FZ, Dou RL, Li MQ, Wu H (2016) A flexible qos-aware web service composition method by multi-objective optimization in cloud manufacturing. *Comput Ind Eng* 99:423–431
30. Zhou JJ, Yao XF (2017) Hybrid teaching–learning-based optimization of correlation-aware service composition in cloud manufacturing. *Int J Adv Manuf Technol* 91(9–12):3515–3533
31. Tao F, Laili YJ, Xu LD, Zhang L (2013) FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. *IEEE Trans Ind Inform* 9(4):2023–2033
32. Liu ZZ, Xue X, Shen JQ, Li WR (2013) Web service dynamic composition based on decomposition of global QoS constraints. *Int J Adv Manuf Technol* 69(9–12):2247–2260
33. Xu WJ, Tian SS, Liu Q, Xie YQ, Zhou ZD, Pham DT (2016) An improved discrete bees algorithm for correlation-aware service aggregation optimization in cloud manufacturing. *Int J Adv Manuf Technol* 84(1–4):17–28
34. Zhou JJ, Yao XF (2016) A hybrid artificial bee colony algorithm for optimal selection of qos-based cloud manufacturing service composition. *Int J Adv Manuf Technol* 88(9–12):3371–3387
35. Li CS, Wang SL, Kang L, Guo L, Cao Y (2014) Trust evaluation model of cloud manufacturing service platform. *Int J Adv Manuf Technol* 75(1–4):489–501
36. Laili YJ, Tao F, Zhang L, Sarker BR (2012) A study of optimal allocation of computing resources in cloud manufacturing systems. *Int J Adv Manuf Technol* 63(5–8):671–690
37. Liu B, Zhang ZL (2017) Qos-aware service composition for cloud manufacturing based on the optimal construction of synergistic elementary service groups. *Int J Adv Manuf Technol* 88(9–12):2757–2771
38. Yang SL, Yang M, Wang S, Huang KD (2016) Adaptive immune genetic algorithm for weapon system portfolio optimization in military big data environment. *Cluster Comput* 19(3):1359–1372
39. Zhu B, Xu ZS, Zhang R, Hong M (2016) Hesitant analytic hierarchy process. *Eur J Oper Res* 250(2):602–614
40. Wu XL, Li RW, Cao YL, Ni YH, Xu X, Qian XY (2016) The value network optimization research based on the analytic hierarchy process method and the dynamic programming of cloud manufacturing. *Int J Adv Manuf Technol* 84(1–4):1–9
41. Zhao J, Jin JL, Zhu JZ, Xu JC, Hang QF, Chen YQ, Han DH (2016) Water resources risk assessment model based on the subjective and objective combination weighting methods. *Water Resour Manag* 30(9):3027–3042

## Publisher's Note

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