IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON COLLABORATION FOR INTERNET OF THINGS

Received August 3, 2018, accepted September 4, 2018, date of publication September 10, 2018, date of current version October 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869275

An Approach to Iot Service Optimal Composition for Mass Customization on Cloud Manufacturing

TIANYANG LI[1], TING HE[1,2], ZHONGJIE WANG[1], (Member, IEEE), AND YUFENG ZHANG[3]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2]College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
[3]Birmingham Business School, University of Birmingham, Birmingham B15 2TT, U.K.

Corresponding author: Ting He (xuantinghe@gmail.com)

This work was supported in part by the National Science Foundation of China under Grant 71571056, in part by the National Key Research and Development Program of China under Grant 2017YFB1400604, in part by the Scientific Research Funds of Huaqiao University under Grant 16BS304, and in part by the National Science Foundation of China under Grant 61772155, Grant 61832014, Grant 61832004, and Grant 61472106.

ABSTRACT For implementing mass customization on cloud manufacturing (CMfg), Internet of Things (IoT)-enabled service optimal composition (ISOC) is a key technology used to effectively composite a manufacturing cloud service with selected IoT services to satisfy the users' customized production requirements. The solving of the ISOC problem is done inefficiently in the context of the growing scale of IoT services and increasing sophistication of the ISOC execution path, being partly due to insufficient investigation of accumulated empirical knowledge (EK) of IoT services. In this paper, we propose an EK-oriented genetic algorithm (EK-GA) for the large-scale IoT service composition. First, by fully considering the distinctive features of IoT service and service domain features, the EK of IoT services are richly explored to divide the service space. Second, EK-oriented optimization strategies in the initial population, operators, and fitness function are presented to improve the local and global search abilities of the genetic algorithm for solving ISOC problems. Finally, the effectiveness of EK-GA for solving real-world ISOC problems in a private CMfg is verified through three types of experiments. By exploiting EK of IoT services for ISOC problems, this work makes novel contributions for mass customization on CMfg and enriches the practice of EK-oriented intelligence optimization.

INDEX TERMS IoT services, service composition, mass customization, cloud manufacturing.

I. INTRODUCTION

In the Cloud Manufacturing (CMfg) paradigm, Internet of Things (IoT)-enabled services take the form of virtual network slices of the shared physical entities and devices [1]–[3], such as 3D printer, intelligent robots/robot arms, simulation and detection devices, machines and resources based on battery powered sensors and actuators, etc. For implementing mass customization on CMfg, it is a promising way to assemble a set of IoT services as a manufacturing cloud service for effectively satisfying user customized production requirements [4]–[6]. According to the given user requirements and constraint conditions, the IoT service assembly process in CMfg starts with the product design and planning of a composed IoT service execution path, collects a corresponding composed IoT services for each path node from the candidate service list and finally optimally selects the IoT service composition/solution for all possible paths to satisfy user requests [7]. A key technology driving both performance and efficiency of IoT service composition is the optimal selection of IoT services. Undoubtedly, building such a Quality of Service (QoS)-aware IoT Service Optimal Composition (ISOC) is a typical multi-criterion nondeterministic polynomial-hard problem [8].

The execution path of ISOC becomes long and complex because of the personalized and complicated production request. Combined with the predicted explosion in the number of connected IoT services and devices in CMfg, the solving of ISOC problems is a time consuming challenge [9]. In an extreme situation, without an effective algorithm, CMfg can collapse when concurrent, massive numbers of users request plentiful IoT service compositions from a mass of alternative candidate IoT services with similar functionalities and the same level of QoS. Moreover, the empirical knowledge (EK) of IoT services has gradually accumulated with

applications and evolutions of IoT services in services and manufacturing industries over the long term [10]. It reflects the substantive characteristics of IoT services used to meet a certain type of customized requirement and strongly influences the solving of ISOC problems in CMfg [11]. Unfortunately, the distinctive features of IoT services, which are different with the stable web services and virtual cloud manufacturing services [12], have challenged the excavation and utilization of EK of IoT services. The EK of IoT services has been explored and used insufficiently to design algorithms for solving ISOC problems [11], [13].

To overcome these gaps, an EK-oriented method, which should take advantage of the fully mined value of EK to properly process the large-scale service space in CMfg and effectively and efficiently search the global optimal solution, must first be created [6], [7], [11]. However, the current IoT service composition methods either barely consider the growing scale of IoT services or roughly exploit the EK of IoT services with inefficient optimization strategies for obtaining the global optimal solution, which further cause inefficiencies in solving ISOC problems.

No comprehensive method exists in the literature that richly investigates EK of IoT services and applies it to the design of effective algorithms. The lack of literature precedent raises two questions: 1) How to discover the EK of IoT services with the essential characteristics of IoT services and use it to process the large-scale service space of ISOC problems in CMfg? 2) How can the EK of IoT services be reasonably used to improve optimization strategies of an algorithm for enhancing its local and global search capabilities?

This paper thus proposes an innovative EK-oriented genetic algorithm (EK-GA) that can be used to implement the mass customization with IoT services in CMfg by effectively solving ISOC problems. The goal is to provide an algorithm to solve ISOC problems with the discovered EK of IoT services based on the concept of Service Domain Features (SDFs), particularly for large-scale IoT service compositions. The effectiveness and efficiency of the proposed algorithm have been validated with three types of experiments.

The paper is structured as follows. SectionII provides an overview of related ISOC works and illustrates current problems. Section III introduces the formulation of ISOC problems and their general mathematical model. The discovery of EK of IoT service and the process for processing the service space are detailed in Section IV. Section V elaborates the EK-GA with optimization strategies. The effectiveness of the EK-GA is verified with three types of experiments in Section VI. Finally, Section VII concludes the paper and provides future research directions.

II. RELATED WORKS

The fundamental researches of IoT services in both cloud computing and CMfg, such as IoT service model [14], IoT-based access of manufacturing resources [3], [15] cloud-based IoT mashup service model [12], monitoring, security and principle of IoT-based cloud systems [16]–[19], IoT-based CMfg [4], collaborative framework of IoT services in manufacturing [20], QoS prediction of IoT service [21], and mathematical formulations of IoT service composition [22], [23], have been well-studied. Successfully tackling such fundamental research challenges plays a wider role in encouraging research into compositing IoT services to satisfy users' customized production requirement.

Methods to IoT service composition have been extensively studied from various perspectives. Some typical studies are summarized in table 1.

TABLE 1. Methods for IoT service composition.

Categories	Current methods
Social Network Based Methods	1. Chen et al. [24] proposed an adaptive and scalable trust management method to support service composition applications in SOA-based IoT systems. 2. By fully considering the social network theory and social properties of IoT services, work [25] designed flexible and scalable schemes for IoT service composition to satisfactorily meet users' requirements from several aspects.
Context and QoS aware methods	1. To support dynamic reasoning on user tasks and service behaviours to deal with the heterogeneity of IoT domains, the study [26] presented an adaptive service composition framework based on an abstract service model representing services and user tasks in terms of their signature, specification and conversation. 2. Work [27] presented a modified LinUCB-hybrid algorithm that regulates the exploitation of known services based on contextual cues and makes use of exploration at appropriate times to discover contextually relevant IoT services for composition by leveraging the detected usage patterns and prior knowledge of crowds of mobile users interacting with various IoT services in-situ. 3. A quantum-inspired genetic algorithm-based approach was proposed in [28], for IoT service selection and composition in cloud middleware based on the end user quality of service requirements.
Energy-aware Methods	1. Works [23] and [29] extended previous QoS-based scheme to consider power consumption when searching for and integrating the least possible number of IoT services, in order to fulfil user requirements with a created energy-aware composition plan.
Model and Semantic based Methods	1. In paper [30], they presented an algebraic model for static IoT service composition that allows the definition of composite services that encompass multiple workflows for run-time scenarios. 2. In paper [31], a Petri net-based model for IoT service composition was proposed to evaluate the cost-effectiveness, a Find to Optimal algorithm was used to find a cost-effective composition path, and a FBased Monitor algorithm was proposed to solve the composition. 3. In order to compose asynchronous RESTful web services and make use of various IoT services, paper [32] proposed an asynchronous RESTful web service recursive measure, which is based on the BPEL extension and an architecture of IOT RESTful web services. 4. Work [33] proposed a semi-automatic approach, tailored to suit the needs of developers, allowing them to easily discover, consume and interconnect IoT services so as to create more complex ones by exploiting their semantics.

From the table, we can see that most of the existing solutions focus relatively narrow on compositing IoT services. Indeed, the trust, relationships, semantics, context,

energy consuming and workflows of IoT services are key factors or management perspectives in composting IoT services. While, in CMfg, with the cloud infrastructures and the growing scale of connected IoT services, these perspectives become the basic management factors for managing and collecting candidate IoT services. Without considering the large scale of IoT services, above methods lead to simple capability aggregation mechanism of IoT services. In addition, only a few works [25], [27] considered to explore partial EK of IoT services for facilitating the IoT service optimal composition. EK of IoT service was not fully developed and used. Consequently, they are inefficient to solve ISOC problems in CMfg.

In service domain, SDFs, including prior, correlation and similarity, are effective means of analyzing EK of services [11]. A number of service optimal selection and composition methods have been proposed to deal with the large scale of candidate services by using a single feature or multiple SDFs. Some typical research efforts are summarized in table 2.

TABLE 2. Methods for service composition with SDFs.

Categories	Current methods
Prior	1. Work [34] employed historical execution sequential patterns or execution dependency of services to select and composite services. 2. Study [35] initialized current optimization algorithms with a historical solution of a request that was similar to the current user requests for compositing services.
Correlation	1. Paper [36] employed the business correlations of services as the key factor to negotiate the choice of concrete services for service composition. 2. Work [37] employed an auxiliary graph to express the QoS correlation-aware service composition, and designed a fast algorithm to search optimal solutions. 3. In paper [13], a formalized description of QoS correlation between two services was first developed, and then algorithms to discover correlations was designed to select and composite services in CMfg.
Similarity	1. Similarity measures to determine the closeness of a historical composition solution with respect to any new request for reusing existing solutions was defined in paper [38]. 2. Work [39] employed the internal features of services and end users, such as locations, configurations, functionality, and user profiles, to calculate similarity of services and then predict the end-to-end QoS values of services for compositing service. 3. A framework to use QoS time series inter-correlations was established in work [40], then a novel time-series group similarity approach to predict QoS values was applied for compositing service.
SDFs (including prior, correlation and similarity)	1. Work [10] defined three SDFs and analysed the influences of these SDFs in compositing services toward proposing a new Service domain-oriented Artificial Bee Colony algorithm paradigm (S-ABC) based on the optimization mechanism of ABC. 2. Work [11] developed a context-aware and Differential Evolution-enhanced Artificial Bee Colony (DE-caABC) algorithm based on SDFs and divisions of the service space for service compositions in CMfg.

As seen from the table, SDFs are key regulators for analysing and utilizing the EK of services, which can effectively facilitate the optimal composition of services. However, above methods in table 2 are not suitable for compositing IoT services in CMfg.

On one hand, current methods define coarse-grained definitions of SDFs and heavily rely on historical solutions and correlation service schemes, which leads to the limited improvement of searching capabilities. At the same time, EK of new services is just analysed with QoS, without assigning the effective prior value. Hence, the potential of SDFs for compositing services remains to be fully developed.

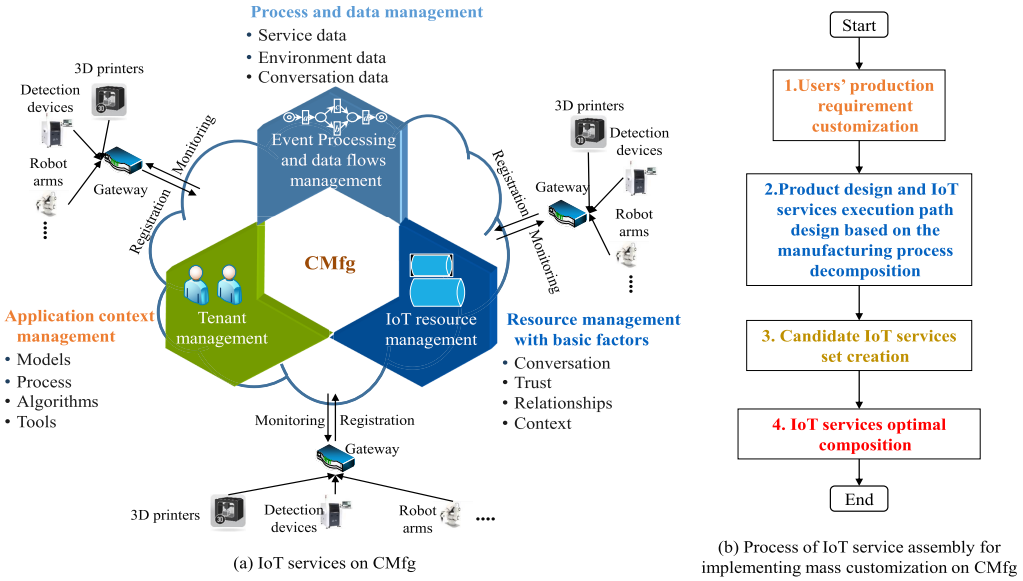
On the other hand, there are several differences between web service and IoT service, current web service composition methods cannot deal with IoT services directly. For instance, devices based IoT services are commonly resource-limited and QoS relevant with the regional agglomeration of devices, while web services are usually powerful computers with sufficient computing, network abilities and weak QoS correlations. In addition, IoT services tend to be unstable and have more uncertainty of network availability while web services are relatively static and reliable, and historical solutions of IoT services may not even be reused for ISOC problems.

In this paper, we investigate the EK of IoT services more deeply with taking a full consideration of IoT services' distinctive features to make the division of the service space more reasonable. Combined with appropriate optimization strategies, we attempt to design a more effective and efficient genetic algorithm to solve ISOC problems with a large-scale service space.

III. PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

The customized production request in CMfg is sequential and massive. When a request is submitted to a CMfg platform, it is first pre-processed with the production design and transferred to a manufacturing task. Then, the task is decomposed into several subtasks in a IoT service execution process based on the manufacturing process. Next, the platform gathers a candidate service set for each subtask. Finally, only one IoT service can be optimally selected from each candidate service set based on multiple objectives or a single objective to be invoked in sequence to construct a composite service execution path. As shown in Fig.1, with the effective management of IoT services in CMfg, such a IoT service assembly process contains four steps. In this paper, we focus on designing an effective algorithm for step 4 to compositing optimal IoT services with a larger scale of candidate service set.

The value of QoS attributes Q_n of an ISOC is aggregated by the corresponding attribute values of all subtasks. Let $q(ISOC_1) = \{q_{1,isoc}, q_{2,isoc}, \cdots, q_{n,isoc}\}$ and $q(IS_i) = \{q_{1,i}, q_{2,i}, \cdots, q_{n,i}\}$ express the aggregated QoS values of ISOC and the QoS values of a single IoT service belonging to the ISOC; then, $q_{n,isoc} = f(q_{n,1}, \cdots, q_{n,j})$. f is an aggregation function based on the structure of the composite service execution path, and n, j are the number of QoS



(a) IoT services on CMfg

(b) Process of IoT service assembly for implementing mass customization on CMfg

FIGURE 1. The architecture of CMfg with IoT services and process of IoT service assembly.

attributes and the number of IoT services in the composition. Currently, there are four main workflow patterns in the composite service execution path: sequential, parallel, selective, and circular. The QoS attributes and their aggregation functions for types of workflow patterns used in this paper are shown in tables 3 and 4, respectively.

TABLE 3. QoS attributes for ISOC.

QoS attributes	Description
Cost	Utility fee of service for each request
Execution time	Time to perform the whole production functionality
Respond time	The waiting time before the ISOC execution
Availability	1-failure rate
Reliability	Uptime/(uptime+downtime)
Energy	Energy consumption of ISOC

TABLE 4. QoS aggregation functions for workflow patterns.

QoS attributes	Sequence	Parallel	Switch	Loop
Cost (c)	$\sum_{i=1}^{k} c(s_i)$	$\sum_{i=1}^{k} C(s_i)$	$\sum_{i=1}^{k} p * c(s_i)$	$l * \sum_{i=1}^{k} c(s_i)$
Execution time (e_t)	$\sum_{i=1}^{k} e_t(s_i)$	$\max\{e_t(s_i)\}$	$\sum_{i=1}^{k} p * e_t(s_i)$	$l * \sum_{i=1}^{k} e_t(s_i)$
Respond time (r_t)	$\sum_{i=1}^{k} r_t(s_i)$	$\max\{r_t(s_i)\}$	$\sum_{i=1}^{k} p * r_t(s_i)$	$l * \sum_{i=1}^{k} r_t(s_i)$
Availability (a)	$\sum_{i=1}^{k} a(s_i)$	$\min\{a(s_i)\}$	$\sum_{i=1}^{k} p * a(s_i)$	$\sum_{i=1}^{k} a(s_i)$
Reliability (r)	$\sum_{i=1}^{k} r(s_i)$	$\min\{r(s_i)\}$	$\sum_{i=1}^{k} p * r(s_i)$	$\sum_{i=1}^{k} r(s_i)$
Energy (e)	$\sum_{i=1}^{k} e(s_i)$	$\sum_{i=1}^{k} e(s_i)$	$\sum_{i=1}^{k} p * e(s_i)$	$l * \sum_{i=1}^{k} e(s_i)$

Commonly, these QoS attributes can be used to establish a multi-objective or single-objective optimization function for ISOC according to the given request. However, the values of QoS attributes should be normalized firstly. There are two types of QoS attributes: positive and negative. Positive attributes mean that the higher the value of the QoS criteria, the better the quality, for example, availability and reliability. However, negative attributes mean that the lower the value of the QoS, the better the quality, for example, execution time, cost and energy. For a single maximization optimization problem of ISOC, the values of positive and negative QoS attributes should be normalized via the following formulas.

$$q_{i,nor-} = \begin{cases} \dfrac{q_{i,\max} - q_{i,isoc}}{q_{i,\max} - q_{i,\min}}, & \text{if } q_{i,\max} \neq q_{i,\min} \\ 1 & \text{if } q_{i,\max} = q_{i,\min} \end{cases} \quad (1)$$

$$q_{i,nor+} = \begin{cases} \dfrac{q_{i,isoc} - q_{i,\min}}{q_{i,\max} - q_{i,\min}}, & \text{if } q_{i,\max} \neq q_{i,\min} \\ 1 & \text{if } q_{i,\max} = q_{i,\min} \end{cases} \quad (2)$$

where $q_{i,\max}$ and $q_{i,\min}$ are the minimum and maximum aggregated values of the i^{th} QoS criteria of the ISOC;

Based on the normalized values of QoS attributes, a utility function can be used to formulate a single maximization optimization problem as follows:

$$\max f(X) = \sum_{i=1}^{n} w_i q_{i,X}$$
$$\text{s.t. } q_{i,X} \geq q_{i,X_n} \cup q_{i,X} \leq q_{i,X_p} \quad (3)$$

where $f(X)$ is the utility function, w_i is the weight for each aggregated value of QoS criteria, $q_{i,X}$ is the aggregated values of the i^{th} QoS criterion of the ISOC, and q_{i,X_n} and q_{i,X_p} are constrained values for negative and positive QoS attributes.

In this paper, we embody this formulation to verify our proposed algorithm.

IV. SERVICE SPACE PRE-PROCESSING WITH EK OF IOT SERVICES

The nature of SDFs is a manifestation of EK which is an important objective regulator in service domain. Based on the definition of SDFs and considered the distinctive feature of IoT services, Bayesian techniques are mainly used to discover and analyze the EK of IoT service for effectively and efficiently solving ISOC problems.

In the following, we firstly introduce how to discover the EK of IoT services, and then describe how to use it to process the large-scale service space of ISOC problems.

A. THE DISCOVERY OF EK OF IOT SERVICES

The EK of IoT services is analysed according to the definitions of SDFs, specifically, prior, correlation and similarity, which can be defined in the following.

Definition 1 (Prior): The prior is a posterior probability of a candidate IoT service $s_{i,j}$ being used to satisfy a request in one class of requirements c_d.

This probability is experiential knowledge abstracted from the IoT service-composition execution record of c_d. From an intuitional perspective, for a specific service request in class c_d, there always exist IoT services or solutions with higher use frequency and user satisfaction.

This objective law can be expressed as a potential use probability of $s_{i,j}$ for a request with higher user satisfaction, which can be calculated by the Bayesian formula as follows:

$$P(c_{d+}/s_{i,j}) = \frac{P(c_{d+})P(s_{i,j}/c_{d+})}{P(s_{i,j})} \tag{4}$$

where $P(c_{d+}) = H^+/H$ is the percentage of positive records (records with higher user satisfaction) from using the partition function $U(h) \geq T$, $U(h)$ is the partition function for calculating the degree of user satisfaction, T is the threshold of value of $U(h)$, H is the number of whole historical records, H^+ is the number of positive records that meet $U(h) \geq T$, $H^- = H - H^+$ is the number of negative records, $P(s_{i,j}$âĄ"$c_{d+}) = N^+/H^+$ is the used probability that $s_{i,j}$ is used in positive records, N^+ is the occurrence number of $s_{i,j}$ in positive records, $P(s_{i,j}) = N/H$ is probability of $s_{i,j}$ being used by requests in c_d, and N is the occurrence number of $s_{i,j}$ in the overall records of c_d.

The prior value of an existing solution S in historical records can be calculated by the Naive Bayes formula:

$$P(S/v_j) = \arg\max_{v_j \in V} P(v_j) \prod_{s_{i,j} \in S} P(s_{i,j}/v_j) \tag{5}$$

here, $V = \{c_{d+}, c_{d-}\}$, and $s_{i,j}$ is the basic service of a solution S. We use the normalized $P(S/c_{d+})$ to represent the prior value of solution S.

For example, table5 shows a segment of historical data. Based on the above formulas, the posterior probability of a certain candidate service and solution can be calculated, specifically,$P(c_1/s_{11}) = \frac{\frac{4}{6}\times\frac{3}{4}}{4/6} = 0.75$,$P(R_1/c_{1+}) = \frac{4}{6} \times \frac{3}{4} \times \frac{4}{6} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{12}$, and$P(R_1/c_{1+}) = \frac{1}{144}$.

TABLE 5. A segment of historical data.

No.	Requset classes	Services records	User satisfaction
R1	c_1	$s_{11}, s_{22}, s_{31}, s_{42}$	c_{1+}
R2		$s_{12}, s_{21}, s_{32}, s_{41}$	
R3		$s_{11}, s_{21}, s_{33}, s_{41}$	
R4		$s_{11}, s_{21}, s_{32}, s_{44}$	c_{1-}
R5		$s_{11}, s_{22}, s_{31}, s_{42}$	
R6		$s_{12}, s_{21}, s_{32}, s_{41}$	
R7	c_2	$s_{16}, s_{24}, s_{33}, s_{44}, s_{51}$	c_{2+}
R8		$s_{16}, s_{23}, s_{32}, s_{44}, s_{53}$	
R9		$s_{11}, s_{21}, s_{32}, s_{42}, s_{53}$	
R10		$s_{11}, s_{21}, s_{32}, s_{42}, s_{53}$	l_{2-}
R11	c_3	$s_{12}, s_{22}, s_{34}, s_{44}, s_{54}, s_{63}, s_{77}$	c_{3+}
R12		$s_{14}, s_{24}, s_{31}, s_{43}, s_{51}, s_{62}, s_{71}$	
R13		$s_{11}, s_{22}, s_{32}, s_{43}, s_{52}, s_{62}, s_{75}$	
R14		$s_{14}, s_{24}, s_{31}, s_{43}, s_{51}, s_{62}, s_{71}$	c_{3-}

Definition 2 (Correlation): The correlation is an evaluation value of the association relationship of IoT services $(s_{i,j}, s_{k,t}, \cdots)$ that are synergistically used to meet the request in c_d with a high users' satisfaction.

The correlation of IoT services includes business and QoS correlation. For the business correlation, the Apriori algorithm is first used to find synergistic IoT services. Then, the correlation value of these synergistic services can be calculated as follows:

$$C_b(c_{d+}/(s_{i,j}, s_{k,t}, \cdots)) = \begin{cases} \dfrac{N^+ - N^-}{N^+}, & \text{if } \dfrac{N^+}{N^-} > 1 \\ 0, & \text{else} \end{cases} \tag{6}$$

where $N+$ and $N-$ are the occurrence numbers of $s_{i,j}, s_{k,t}, \cdots$ in positive and negative records, respectively. Take the synergistic two IoT services s_{21} and s_{41} in table5 as an example, $C_b(l_1/(s_{21}, s_{41})) = \frac{2-1}{2} = \frac{1}{2}$.

For the QoS correlation, which is a distinctive feature of IoT services, we only consider two neighbour IoT services that have a higher QoS dependence. In the historical data of the request class c_d, except the start and final IoT services of each record, each IoT service has two sets of neighbour services. For certain $s_{i,j}$, the first set F includes services that executed before it in different records, and the second set S contains services that executed after it. For example, in table5, two sets of neighbour services of s_{21} in c_1 are $F = \{s_{11}, s_{12}\}$ and $S = \{s_{32}, s_{33}\}$, respectively. For each set, there is an average value for evaluating the difference degree of QoS correlation between $s_{i,j}$ and each s_k in it. The average value of service for F and S (here, we take services in F as an example) can be calculated by:

$$\phi(c_{d+}/(s_{i,j}, F)) = \frac{\sum C_q(c_{d+}/(s_{i,j}, s_k))}{|F| \times \max C_q(c_{d+}/(s_{i,j}, s_k))} \tag{7}$$

where $C_q\ (c_{d+}/(s_{i,j}, s_k))$ is the normalization function with the avarage QoS value of services $(s_{i,j}, s_k)$, $s_k \in F$ is a IoT service in service set F . Its formula is defined as the following:

$$C_q(c_{d+}/(s_{i,j}, s_k)) = \frac{f(s_{i,j}, s_k) - f_{\min}(s_{i,j}, s_k)}{f_{\max}(s_{i,j}, s_k) - f_{\min}(s_{i,j}, s_k)} \tag{8}$$

where $f(s_{i,j}, s_k) = \frac{\sum_{k=1}^{N^+}\sum_{t=1}^{6} w_t q_t - \sum_{k=1}^{N^-}\sum_{t=1}^{6} w_j q_t}{N^+ - N^-}$ is the utility function, the same as the function in Eq.3 for calculating an average aggregated QoS value of two neighbour services in the historical data.

When $\phi(c_{d+}/(s_{i,j}, F))$ is very small, it illustrates that the difference degree of QoS correlation between $s_{i,j}$ and each s_k in F or S is very large. Hence, top k IoT services s_k in F or S, which have largest values of $C_q(c_{d+}/(s_{i,j}, s_k))$, should be considered to have highest QoS correlation with $s_{i,j}$.

For example, we assume that we initially use $f(s_{i,j}, s_k)$ to calculate two values 2, 1 for services in $\{s_{11}, s_{12}\}$ and then use $C_q(c_{d+}/(s_{i,j}, s_k))$ to obtain two normalized values 1, 0. The difference degree of QoS correlation between s_{21} and s_{11} and between s_{21} and s_{12} can be calculated by $\phi(c_{d+}/(s_{i,j}, F)) = 0.5$. In other words, s_{21} and s_{11} have higher QoS correlation than do s_{21} and s_{12}, and the greater value with $C_q(c_{d+}/(s_{i,j}, s_k))$ represents the higher QoS Correlation.

Instability is another distinctive feature of IoT services. On one hand, some IoT services cannot be used at a certain time and have no records for analyzing their prior. On the other hand, some IoT services update their devices or entities with new QoS abilities, which need to reevaluate their prior values. In this case, similarity is used to estimate the prior of unstable IoT services.

Definition 3 (Similarity): The similarity is the similar degree of QoS values between two candidate IoT services that have the same service functionalities and belong to the same candidate IoT service set.

When an unstable IoT service s_n are identified, the QoS prediction method for IoT services proposed in [21] is firstly used to estimate the QoS of s_n. Then the estimated QoS value and Euclidean distance are employed to find m similar IoT services of s_n from the same candidate set which contains services with values of prior. Finally, the prior of s_n is calculated by:

$$P(c_{d+}/s_n) = \sum_{i=1}^{m} w_i \times P(c_{d+}/s_i)$$
$$w_i \in W_i = \frac{Z_i^{-1} 1_m}{1_k^T Z_i^{-1} 1_m} \tag{9}$$

here, w_i is the reconstruction weights based on QoS values of similar IoT services by referencing the Locally Linear Embedding method, and $Z_i = (f(x_n) - f(s_i))^T (f(x_n) - f(s_i))$.

With the above-defined formulations, for a specific service request in c_d, candidate IoT services and solutions that have the historical data about c_d, can have a prior value used to identify the probability of services and solutions to satisfy the request. At the same time, the high-frequency schemas of IoT service can be found from the perspectives of business correlation and QoS correlation. These schemas are additional supplements for the discovered high-probability services. In addition, based on the identified probabilities and similarity, the prior value of unstable and newly connected IoT services can be estimated.

The identified prior values and correlation schemas are the EK of IoT services, which means that IoT services with a higher prior value and a high-frequency or QoS correlation schema could be selected to composite the local optimal solutions or even the global optimal solution. Consequently, based on the EK of IoT services, the large-scale candidate service set can be reasonably divided. Moreover, appropriate EK-oriented search strategies of algorithms can be designed to dramatically promote the performance of algorithms for searching for the global optimal solution. Next, we define the division of service space and discuss the design strategies of algorithms influenced by the EK of IoT services.

B. SERVICE SPACE DIVISION BASED ON EK OF IOT SERVICES

The whole candidate service set $S_p = \cup S_i$ for a service request in c_d consists of candidate service sets S_i of each subtask (path node) in the execution path. With identified prior values and correlation schemas, each candidate IoT service set S_i can be divided into the optimal Service Subset $OptS(S_i, c_d)$ and General Service Subset $GenS(S_i, c_d)$. Accordingly, the whole service space is divided into two subspaces $OptS(S_p, c_d) = \cup OptS(S_i, c_d)$ and $GenS(S_p, c_d) = \cup GenS(S_i, c_d)$. The formal description of these two subsets can be expressed as follows.

1) $OptS(S_i, c_d) = PriS(S_i, c_d) \cup CorS(S_i, c_d)$

where $PriS(S_i, c_d) = \{s \in S_i | P(c_{d+}/s) \geq P_t\}$, $CorS(S_i, c_d) = \{s \in S_p | C_b(c_{d+}/(s_{i,j}, s_{k,t}, \cdots)) \geq C_{b0} \cup (C_q(c_{d+}/(s_{i,j}, s_k)) \geq C_{q0} \cap \phi(c_{d+}/(s_{i,j}, F)) \leq n_t)\}$ are prior service set and correlation set respectively, P_t is the threshold of prior value of candidate services based on the partition function $U(h) \geq T$; empirically, $P_t = 0.5$ might be the lower limit value with a certain value T, C_{b0} and C_{q0} are thresholds of correlation vlaue, $C_{b0} = 0$ might be the lowest limits, and $C_{q0} = 1$ must be the upper limits when $\phi(c_{d+}/(s_{i,j}, F)) \leq n_t$ is smaller than the threshold n_t, n_t should be smaller than 0.5.

2)$GenS(S_i, c_d) = S_i - OptS(S_i, c_d)$.

When each S_i is divided into two subsets $OptS(S_i, c_d)$ and $GenS(S_i, c_d)$, the completeness of subset $OptS(S_i, c_d)$ can be further discussed.

Definition 4 (Completeness): The completeness is a relative property for the subset $OptS(S_i, c_d)$, which means that $OptS(S_i, c_d)$ is complete when it has at least one IoT service better than all IoT services in $GenS(S_i, c_d)$.

Based on the above definition, a complete subset $OptS(S_i, c_d)$ belonged to different subtasks in different workflow patterns must meet a certain condition. Based Eq.3 and functions in table 4, for subsets belonged to a subtask in sequential, selective, and circular patterns, they should have at least one IoT service that meets the following condition.

$$\sum_{s_k \in OptS_i} w_i q_{i,s_k} - \max\{\sum_{s_t \in GenS_i} w_i q_{i,s_t}\} > 0 \tag{10}$$

Analogously, one subset $OptS_i$ belonged to a subtask in the parallel pattern, should have the minimum value of negative

attributes and the maximum value of positive attributes compared with IoT services in $GenS(S_i, c_d)$.

In this way, when all subsets $OptS_i$ are complete, the optimal subspace $OptS(S_p, c_d)$ is considered complete as well, and the global optimal solution of ISOC should be located in it.

Theorem 1: If the optimal subspace $OptS(S_p, c_d)$ is complete, the global optimal solution O_p is located in it.

Proof: Proof by contradiction.

Case (1): we have a assumption that all IoT services of O_p are from $GenS(S_p, c_d) = \cup GenS(S_i, c_d)$. Without loss of generality, we can change the IoT service from $GenS(S_i, c_d)$ with a better IoT service from $OptS(S_i, c_d)$ and get a new solution O_n. With the utilization function in Eq.3 and aggregation functions in table4, we can get that $f(O_n) > f(O_p)$ (when the selected services is belonged to subtasks in the sequential, selective, and circular patterns) or $f(O_n) \geq f(O_p)$(when the selected services is belonged to subtasks in the parallel pattern). Consequently, O_p is not the global optimal solution, and the assumption is invalid.

Case (2): we have a assumption that IoT services of O_p are partly from $GenS(S_p, c_d) = \cup GenS(S_i, c_d)$. Similar to, we can get that this assumption is also invalid.

According to these two cases, we can conduct that the global optimal solution O_p is located in $OptS(S_p, c_d)$. □

Once the service space is divided, the completeness of subspace $OptS(S_p, c_d)$ can be analyzed. If the subspace $OptS(S_p, c_d)$ is complete, intelligent optimization algorithms only need to search it for finding the global optimal solution. This can effectively reduce the searching time of algorithms. However, conditions of completeness are very strict. Specifically, if a subset $OptS(S_i, c_d)$ belonged to a subtask in the parallel pattern, it is difficult to find a IoT service that satisfies the condition. Usually, the subspace $OptS(S_p, c_d)$ is just considered to have a higher probability to contain the global optimal solution, which should be searched emphatically.

The procedure for dividing the service space is shown in algorithm 1.

The above algorithm can be used to divide the service space properly. The thresholds are set with $P_t = 0.5$, $C_{b0} = 0$, $C_{q0} = 1$ (only the best QoS relevant service is selected) and $n_t = 0.5$. They rely on the threshold T; if a more strict value T (a larger one if T is positive or otherwise a small one) is used, then P_t and C_{b0} should be set with smaller value, whereas greater values might be appropriate. The threshold n_t is set to 0.5, which is a basic value for the difference degree of QoS correlations, where a small value should be used if prominent QoS correlations are expected. Hence, the threshold T determine the scales of $OptS(S_p, c_d)$.

In order to effectively and efficiently search the divided service space, optimization strategies based on the EK of IoT services for solving ISOC problems should be further developed to improve the searching ability of intelligent optimization algorithms.

Optimization strategies based on EK can be summarized as follows:

Algorithm 1 Processing the Service Space

Require: Historical data H, $U(h)$, T, candidate service set S_p;
Ensure: two subsets;
1: $OptS(S_p, c_d)$ and $GenS(S_p, c_d) \leftarrow emptyset$;
2: $H^+, H^- \leftarrow$ employ $U(h) \geq T$ to divide H;
3: $C_s \leftarrow$ employ Apriori algorithm to get service schemes from H;
4: **for all** $s \in (S_p \cap H)$ **do**
5: $P_s \leftarrow$ employ Eq.4 to calculate a prior value for s;
6: **if** $P_s \geq 0.5$ **then**
7: add s to $OptS(S_p, c_d)$ and add P_s to value set V_p;
8: **end if**
9: **end for**
10: **for all** $cs \in C_s$ **do**
11: $C_i \leftarrow$ employ Eq.6 to a calculate correlation value for cs;
12: **if** $C_i > 0$ **then**
13: $OptS(S_p, c_d) = OptS(S_p, c_d) \cup cs$ and add C_i to value set V_p;
14: **end if**
15: **end for**
16: **for all** $s \in OptS(S_p, c_d)$ **do**
17: $F, S \leftarrow$ construct two neighboring service sets for s;
18: **if** $|F| \geq 2$ **then**
19: $\phi \leftarrow$ employ Eq.7 to calculate the degree value;
20: **end if**
21: **if** $\phi \leq 0.5$ **then**
22: add the service in F that has the value 1 calculated by Eq.8;
23: **end if**
24: **if** $|S| \geq 2$ **then**
25: $\phi \leftarrow$ employ Eq.7 to calculate the degree value;
26: **end if**
27: **if** $\phi \leq 0.5$ **then**
28: add the service in S that has the value 1 calculated by Eq.7;
29: **end if**
30: **end for**
31: **for all** $s \in S_p = S_p - (S_p \cap H)$ **do**
32: $P_s \leftarrow$ employ Eq.9 to estimate a prior value for s;
33: **if** $P_s \geq 0.5$ **then**
34: add s to $OptS(S_p, c_d)$ and add P_s to value set V_p;
35: **end if**
36: **end for**
37: $GenS(S_p, c_d) = S_p - OptS(S_p, c_d)$;
38: **Return** $GenS(S_p, c_d)$, $OptS(S_p, c_d)$;

1) Initialize an appropriate number of superior individuals for the initial population from different subspaces to obtain better initial positions;

2) Control the search direction of the algorithms for widely searching the subspace $OptS(S_p, c_d)$ first; and

3) Improve the fitness function to fully search subspaces and to reduce deception problems.

V. EMPIRICAL KNOWLEDGE-ORIENTED GENETIC ALGORITHM

Considering these strategies, this paper proposes an empirical knowledge-oriented genetic algorithm (EK-GA) based on improving the initialization of the initial population, genetic operators and fitness function.

A. INITIAL POPULATION OF EK-GA

The searching strategy used by EK-GA is to search for the global optimal solution in both two subspaces simultaneously, which means that individuals in the initial population should be initialized from two services subspaces $OptS(S_p, c_d)$ and $GenS(S_p, c_d)$. To retain the diversity of the population, we initialize better individuals and deeply search each subspace, and a certain number of individuals should be generated from each subspace. The number of individuals from each set can be equal or correspond to the proportion of subspaces. In this paper, the proportion of each subspace is used as the proportion of individuals generated from each subspace. All individuals can be expressed as $X = (x_1, x_2, \cdots, x_n)$, where x_n is a real serial number of selected IoT service for a subtask, and the length of an individual is the number of subtasks involved in an ISOC execution path.

Three strategies are used to initialize individuals in two subspaces. The first strategy is to randomly generate individuals. The second strategy is to generate individuals with $\max \prod (1 + v_j)$ (one IoT serivce can be selected only once), and v_j is the prior or correlation value of selected service j. The third strategy is to reuse the existing solution, which has the highest probability with prior values based on Eq.5. Use probabilities of these strategies are equal. Algorithm. 2 shows the common procedure for generating individuals from two subspaces.

B. OPERATORS OF EK-GA

The crossover of EK-GA is a single-point crossover that selects a location of two individuals to exchange their IoT services after the selected location. To search the subspaces as far as possible, three strategies are used to generate one offspring based on two selected parent individuals. The first strategy is to search the location of two randomly selected parent individuals for crossover to generate one offspring with $\max \prod (1 + v_j)$. The second strategy is to find the location of two selected parent individuals (one is randomly selected from the last generation, the other is randomly generated from the optimal subspace with $\max \prod (1 + v_j)$) for crossover to generate one offspring with the maximum value of fitness $\max f(X_i)$. The third one is to randomly select a location based on the correlation of IoT services for crossover two randomly selected parent individuals and to generate an offspring which has correlation services with the highest correlation value (if IoT services in parent individuals are not correlated, then reselect parent individuals or generate from the optimal subspace). The probability of being used for each strategy is equal. Algorithm. 3 shows the reproduction algorithm with two selected parents.

Algorithm 2 Individuals Generation

Require: Candidate service set S, value set V_p, and number of individuals N;
Ensure: a set of individuals;
Individual set $I_s \leftarrow$ empty set;
for 1 to N **do**
  $x \leftarrow$ a random integer from 1 to 3;
  switch (x)
  case 1**:**
    $X_i \leftarrow$ randomly select an IoT service x_n from S for each subtask to generate an individual;
    add X_i to I_s, break;
  case 2**:**
    $X_i \leftarrow$ select an IoT service x_n from S for each subtask to generate an individual with $\max \prod (1 + v_j)$;
    add X_i to I_s, break;
  case 3**:**
    $X_i \leftarrow$ select an existing solution with $\max v_j$ from S to generate an individual;
    add X_i to I_s, break;
  end switch
end for
Return a set of individuals I_s;

Algorithm 3 Reproduction Individuals

Require: The value sets V_p and number of individuals N;
Ensure: a set of individuals;
Individual set $I_s \leftarrow$ empty set;
for 1 to N **do**
  $x \leftarrow$ a random integer from 1 to 3;
  switch (x)
  case 1**:**
    $X_i \leftarrow$ randomly select two parent individuals from the last generation to generate two offsprings;
    add the one child with $\max \prod (1 + v_j)$ to I_s, break;
  case 2**:**
    $X_i \leftarrow$ select an individual from the last generation and genetate an individual from $OptS(S_p, c_d)$ with $\max \prod (1 + v_j)$ to generate two offsprings;
    add the one child with best fitness to I_s, break;
  case 3**:**
    $X_i \leftarrow$ randomly select two generate parent individuals from the last generation to generate two offsprings;
    add the one that has correlation services with the highest correlation value to I_s, break;
  end switch
end for

The mutation operator is also a single-point mutation that selects a position of an individual and exchanges the IoT

service in the selected position with the other service in the candidate service list. To increase the diversity of new generations, two simple strategies are used to mutate the selected individual. The first one is to randomly select a position in the newly generated individual and randomly replace the selected candidate service with another one randomly from the same candidate list in $OptS(S_p, c_d)$. The second strategy is to replace the selected candidate service based on the correlation of IoT services, specifically, the IoT services which have the highest correlation value with other services of the selected individual is selected to replace the IoT service. Algorithm 4 shows the algorithm for the mutation operator.

Algorithm 4 Mutate

Require: An individual X_i;
Ensure: a new individual;
1: randomly select a service x_j from X_i;
2: $x \leftarrow$ a random integer from 1 to 2;
3: **switch** (x)
4: **case** 1**:**
5: Exchange x_j with a randomly selected IoT service, break;
6: **case** 2**:**
7: Exchange x_j with a service from $OptS(S_p, c_d)$ that has the highest correlation value with other services in X_i, break;
8: **end switch**
9: **Return** a set of individuals I_s;

C. FITNESS FUNCTION OF EK-GA

For EK-GA, the fitness of an individual should not only reflect QoS utility, but also indicate how the individual can satisfy EK constraints for searching dividing subspaces. Consequently, the optimization problem and EK constraints should all be considered. The fitness function can be defined as follows:

$$F(X_i) = \alpha f(X_i) + \beta \frac{\eta}{T} f_{EK}(X_i) \tag{11}$$

where $f(X_i)$ is the objective function, $f_{EK}(X_i) = \prod(1 + v_j)$ is the EK constraint function, v_j is the feature value of service j in the individual X_i. $0 \leq \eta \leq \frac{\max f(X_i) - \min f(X_i)}{\max f_{EK}(X_i)}$ is the balanced coefficient for the EK function; T is the number of iterations; and $\alpha + \beta = 1$ is the weight. The coefficient $\frac{\eta}{T}$ becomes smaller with each iteration, which ensures that later iterations focus on the true fitness of individuals and prevent the algorithm from trapping in the local optimum in a certain subspace. In this paper, we set $\alpha = \beta = 0.5$.

However, the divided service space based on EK makes the search landscape more rugged. For searching the global optimal solution more effectively, the optimal subspace that might contain the global optimal solution must be fully explored. The fitness-sharing niching technology is used for GA to explore more searches in each subspace and to maintain the diversity of a population to obviate premature convergence. The amendatory fitness function is defined as follows:

$$F_{nic} = \frac{F(X_i)}{\sum_{h \in \text{population}} S_{h,i}} \tag{12}$$

here, $\sum_{h \in \text{population}} S_{h,i}$ represents the crowdedness degree of individual X_i in its niche. $S_(h, i)$ is the sharing value between X_i and X_h and can be calculated as follows:

$$S_{h,i} = \begin{cases} \dfrac{\delta - d_{h,i}}{N}, & \text{if } d_{h,i} < \delta \\ 0, & \text{else} \end{cases} \tag{13}$$

where δ is a defined sharing radius, and $d_{h,i}$ is the number of different services between X_i and X_h.

By using the amendatory fitness, highly similar individuals are discouraged on selection and crossover operators. For example, $X_i = \{x_{12}, x_{23}, x_{37}, x_{42}, x_{51}, x_{66}\}$, $X_j = \{x_{16}, x_{29}, x_{37}, x_{42}, x_{55}, x_{62}\}$ and $X_k = \{x_{16}, x_{29}, x_{37}, x_{43}, x_{55}, x_{62}\}$ are three individuals with fitness values 0.6, 0.8 and 0.9, respectively. $d_{i,j} = 4$, $d_{i,k=5}$ and $d_{j,k} = 1$; if the δ is set with 2, then $S_{i,j} = 0$, $S_{i,k} = 0$, $S_{j,k} = 0.5$ and $S_{i,i} = S_{j,j} = S_{k,k} = 1$. By using Eq.12, $F_{nic}(X_k) = F(X_k)/(S_{i,k} + S_{j,k} + S_{k,k}) = 0.9/1.5 = 0.6$. Similarly, we can have $F_{nic}(X_i) = 0.6$ and $F_{nic}(X_j) = 0.533$. Hence, the two closer individuals X_j and X_k are suppressed to some extent, and individual X_i, which uniquely exploits areas of the search space, is encouraged for evolution.

Based on the above-designed strategies, algorithm.5 shows the pseudocode of EK-GA.

Algorithm 5 Processing the Service Space

Require: N, r, m and S_p ;
Ensure: a solution;
1: preprocessing the service space: employ algorithm 1 to process the service space S_p;
2: initialize initial population P: employ algorithm2 to generate $\frac{|OptS(S_p, c_d)|}{|S_p|} \times N$, and $\frac{|GenS(S_p, c_d)|}{|S_p|} \times N$ individuals from each subspace;
3: fitness evaluation: for each individual, employ Eq.12 to calculate the fitness $F_nic\ (p_i)$;
4: **repeat**
5: new-population $P_n \leftarrow$ empty set;
6: selection: add $(1 - r) \times N$ individuals from P to P_n with the roulette method;
7: crossover: employ algorithm 3 to generate $r \times N$ individuals and add to P_n;
8: mutation: employ algorithm 4 to mutate a new individual with the probability $m\%$;
9: update P with P_n;
10: **until** stopping criteria is satisfied;
11: **Return** the best individual in P, according to $f(p_i)$;

VI. EXPERIMENT AND DISCUSSION

A. OBJECTIVES AND DATASET

The performance of EK-GA is evaluated by using three yardsticks: convergence speed, capability for finding the best

solution with limited iterations and probability of obtaining the global optimum solution. Different objectives are considered to analyse EK-GA:

1) Analyse the effect of EK on the performance of the proposed algorithm;

2) Assess the effectiveness of optimization strategies for the proposed algorithm; and

3) Comparatively evaluate the performance of EK-GA against other EK-oriented algorithms.

Consequently, three types of experiments were implemented by using Visual Studio 2010 with *C* sharp language on a PC: Intel(R) I5-6200U @ 2.30 GHz CPU and 8G Memory. One dataset is collected based on a real-world dataset from a private CMfg. This CMfg belongs to one of the largest manufacturers in China, who is trying to take advantage of CMfg to realize mass customization. Based on this CMfg, a specific demand of customized production of household electrical appliances can be easily satisfied by finding an optimal solution based on the service process with 117 subtasks (see Fig.2; black cuboids represent a set of the same type of subtasks). For each subtask, there are 226 qualified candidate IoT services (including IoT services provided by SME and individual businesses) on average. Constrains for QoS attributes are $c \leq 700\$$, $e_t \leq 17d(days)$, $r_t \leq 20h$, $a \geq 90$, $r \geq 90$, and $\leq 800 \times 10^3$ mA (only for power utilization equipments).

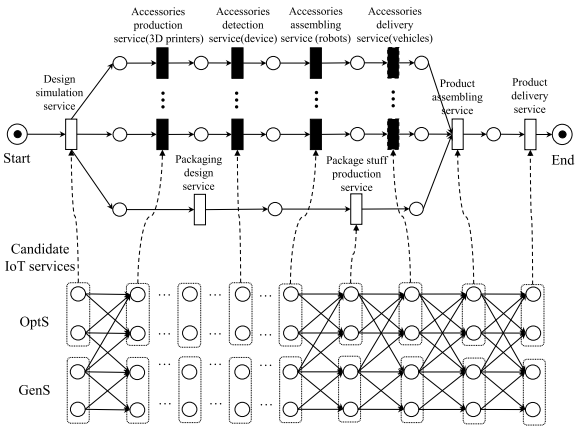


FIGURE 2. IoT service process with subtasks for customized household electrical appliances production.

B. EFFECT OF EK ON THE PROPOSED ALGORITHM

The experiment is designed to analyse the influence of different richness of EK on the effectiveness of EK-GA for solving ISOC problems. Accordingly, this experiment uses different thresholds T to generate EK with different richness and runs EK-GA and Traditional Genetic Algorithm (T-GA) to analyse their performance.

In this experiment, T-GA is initially performed 200 times with 100 iterations to generate historical solutions. Different percentage values of historical solutions are used as threshold T to divide the generated solution into positive and negative solutions for gaining feature sets with different richness. For a certain percentage value p, EK-GA is performed 50 times with 100 iterations to comprehensively assess its convergence speed, capability for finding the best fitness with limited iterations and probability of obtaining the global optimal solution. For all individuals of EK-GA and T-GA, their fitness values are normalized by Eq.2 which transfers the best fitness value (the minimum cost) to 0 and the worst fitness value to 1. Moreover, based on the solutions found by EK-GA and T-GA, solutions belonging to the set $F = \{f_i | f_{best} - f_i \leq \xi = 0.08\}$ (f_{best} is the best solution found by EK-GA) are considered as the global optimal solution. Once the best solution searched by algorithms belong to the set F, the number of finding the global optimal solution is increased. Comparison results are shown in Fig. 3, in which the performance of T-GA is used as a basic line.

Fig.3(a) shows the convergence speed results. The convergence speed of T-GA is the average value of convergence time over 200 times. With the increasing percentage of positive solutions, the average convergence speed of EK-GA increases first and then decreases. When the scales of positive solutions are between 30 and 60 percent, the convergence speed increases dramatically. The reason for this phenomenon might be that, in these cases, the optimal subspace $OptS(S_p, c_d)$ can contain reusable solutions and more-superior candidate IoT services. In other words, the probability that $OptS(S_p, c_d)$ will include the global optimal solution increases. When scales of positive solutions are greater than 70 percent, the convergence speed decreases because the scale of $OptS(S_p, c_d)$ is larger, and more time is needed to search the subspaces.

Fig.3(b) shows the result of the capability of finding the best solution with limited iterations. The best average fitness increases with the increase of positive solutions. When the percentage of positive solutions is up to 60 percent, the best average fitness reaches a maximum and declines with the increase of percentage of positive solutions. This may because when the optimal subspace contains the necessary qualified candidate IoT services for the global optimal solution, EK-GA can find it; however, when the percentage of positive solutions is too large, some unqualified IoT services are assigned to the optimal subspace, which inflences the performance of EK-GA.

Fig.3(c) shows a similar circumstance. When the percentage of positive solutions increases, the probability of obtaining the best optimal solution can reach the highest value of 83% and decline slightly from the latter.

Overall, with the appropriate division of service space, the performance of EK-GA is outstanding relative to T-GA. Experimental results show that EK are always valid even the EK is insufficient.

C. EFFECTIVENESS OF OPTIMIZATION STRATEGIES OF PROPOSED ALGORITHM

Three experiments are designed to verify the effectiveness of optimization strategies for the proposed algorithm. Similarly, 60 per cent is used as the threshold T to process the

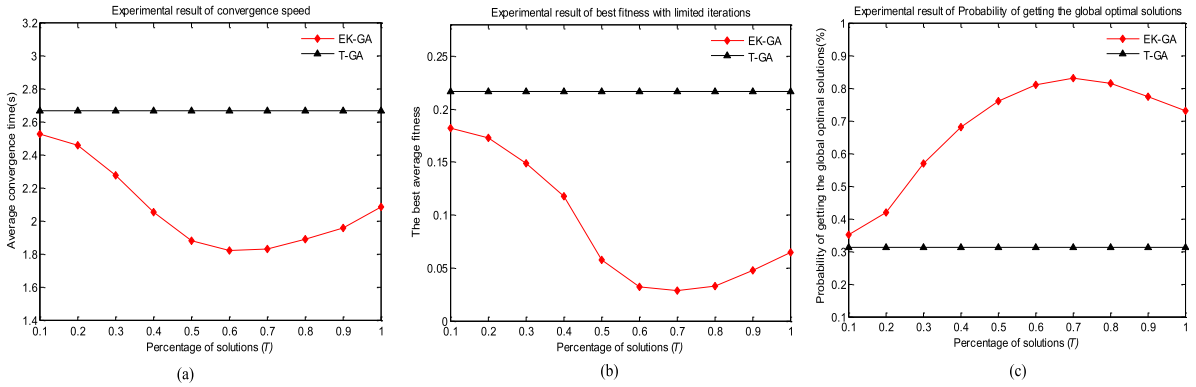


FIGURE 3. Effectiveness of EF-GA with different thresholds p. (a) Convergence speed comparison. (b) Capability comparison. (c) Probability comparison.

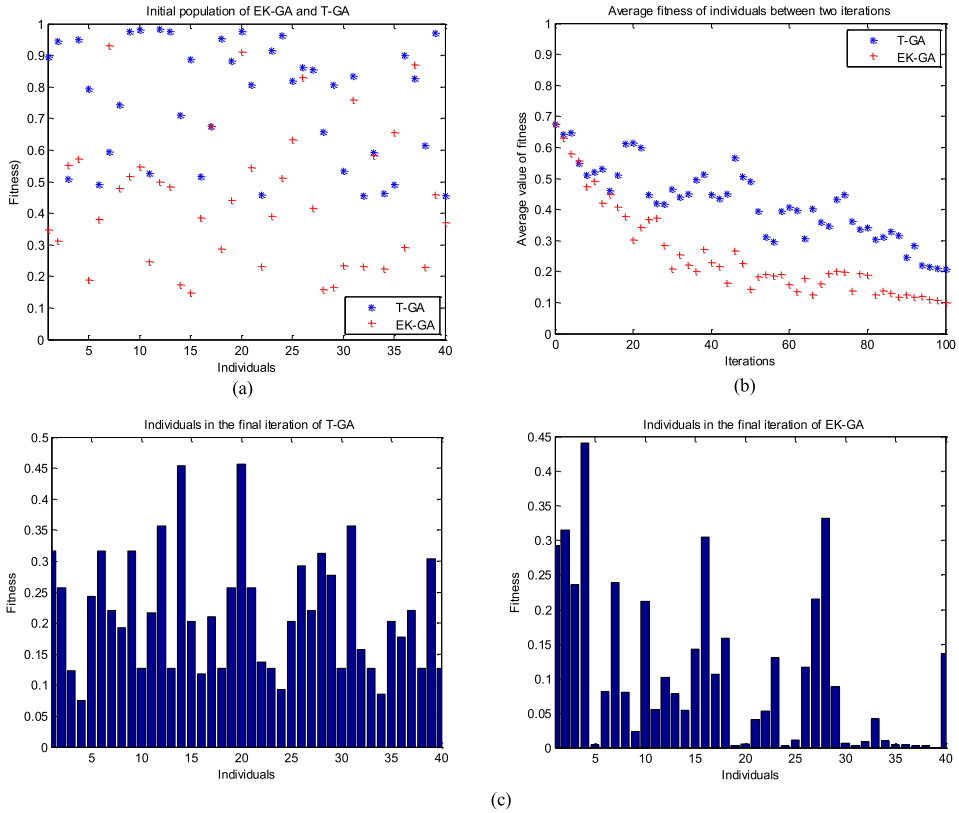


FIGURE 4. Effectiveness of optimization strategies of EK-GA. (a) Effectiveness of initial population. (b) Effectiveness of operators. (c) Effectiveness of fitness function.

service space. Algorithm 2, 3, 4 and 5 are performed separately to analyse each strategy. Algorithms 2 is performed to analyze the strategy of initialization population. Algorithms 3 and 4 are performed together with the normal initial population and normal fitness function of T-GA to verify the improved operators, and algorithm 5 is performed with a normal initial population and operators of T-GA to evaluate the fitness function. Compared experimental results are shown in Fig.4.

Fig.4(a) shows the initial population generated by algorithm 2 and T-GA. Most individuals generated by algorithm 2 are better than solutions initialized by T-GA, which confirms the effectiveness of algorithm 2.

Fig.4(b) shows the average fitness of two iterations of T-GA and T-GA with algorithms 3 and 4. This result confirms that with the EK-oriented operators, EK-GA can fast search the best solution. Whereas EK-GA runs for approximately 20 iterations, new generations of EK-GA can reuse local optimal solutions generated by T-GA to search further for the global optimal solution. This point is consistent with the result shown in Fig.3(b), which also suggests that EK-GA's performance depends heavily upon the quality of EK.

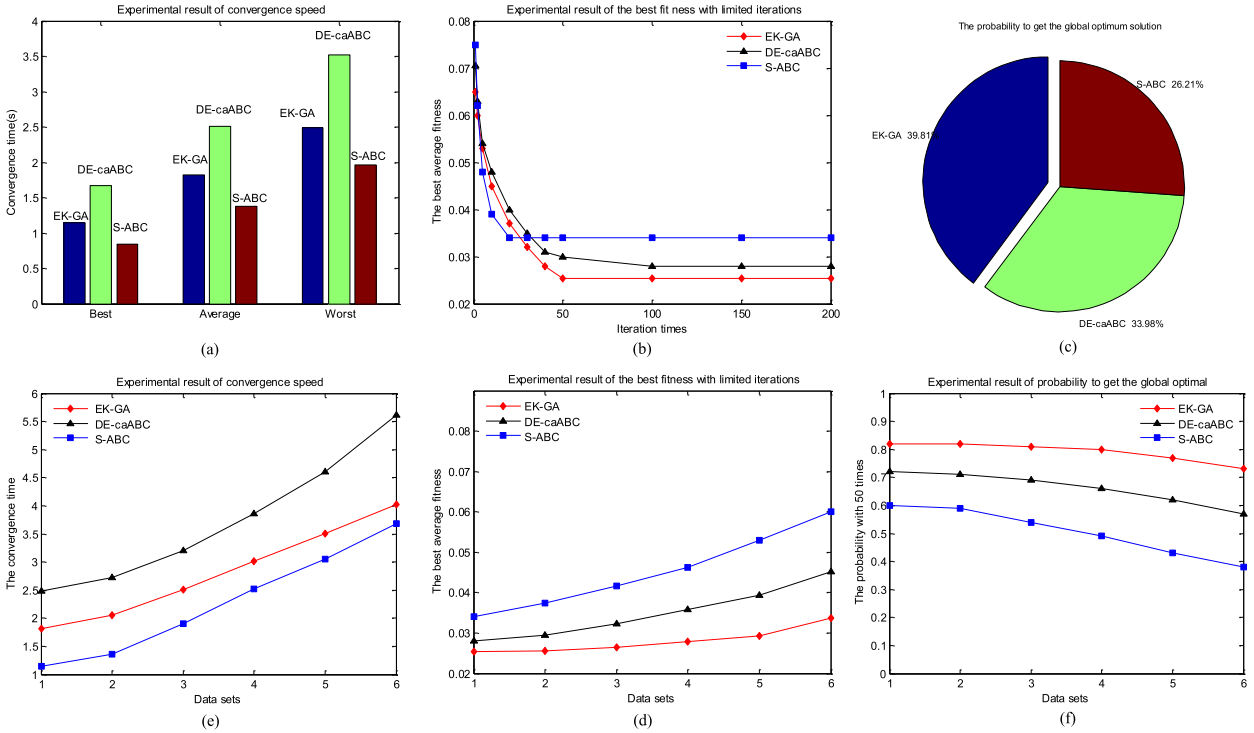


FIGURE 5. Effectiveness of EK-GA compared with other EK-oriented algorithms. (a) Convergence speed comparison. (b) Capability comparison. (c) Probability comparison. (d) Capability comparison. (e) Convergence speed comparison. (f) Probability comparison.

Thus, if the quality of EK is not very good, then the algorithm may converge rapidly to a poor solution around the local optimal solution.

Fig.4(c) contrastively shows the individual fitness of the final iterations of T-GA and EK-GA, which have different fitness functions. As seen from the figure, the population of EK-GA has a greater diversity than does T-GA, which implies the useful effect of fitness-sharing niching technology.

In summary, these experiments verify the effectiveness of optimization strategies. Combining with the results of the these three experiments above, we can conclude that, compared with T-GA, EK-GA has a higher performance for solving ISOC problems.

D. SEARCHING ABILITY OF PROPOSED ALGORITHM COMPARED WITH OTHER EK-ORIENTED ALGORITHMS

Two sets of experiments are designed to assess the performance of EK-GA compared with DE-caABC and S-ABC, which are two existing EK-oriented algorithms and have been proved superior to traditional algorithms such as GA, ABC, and AC, especially for solving service composition problems with a large-scale service space. The setting of DE-caABC and S-ABC are used as the original one in the corresponding study. These two algorithms were first performed 50 times with 200 iterations on the existing dataset. The comparative results with EK-GA are shown in Fig.5 (a)-(c). Moreover, to fully assess the performance of the algorithms, five datasets were synthesized based on the pre-existing dataset. The synthesized datasets are shown in table 6. The first one is the pre-existing one. For each synthesized set, the GA also is used to generate a solution to gain feature sets with the 60 percent threshold. Then, these three algorithms are performed 50 times with 200 iterations. Experimental results are shown in Fig.5(d)-(f).

TABLE 6. Details of the synthetic datasets.

No.	1	2	3	4	5	6
Subtasks	117	120	140	160	180	200
Candidate IoT services	256	300	350	400	450	500

Fig.5(a) shows the convergence speed of the algorithms; the convergence speed of EK-GA is better than that of DE-caABC and worse than that of S-ABC. This could be because that the utilization of prior schemas helps S-ABC to converge faster and the DE stage in DE-caABC needs more time.

Fig.5(b) shows the best average fitness; the performance of EK-GA is the highest. S-ABC can rapidly converge, possibly because abundant prior schemas are used in S-ABC and there is little focus on QoS correlation. DE-caABC has the same drawbacks, but the DE stage can help DE-caABC find better solutions based on generated local optimal solutions.

Fig.5(c) shows the comparative probabilities of obtaining the global optimal solution. As seen from the figure, the performance of EK-GA is best, and DE-caABC is better than S-ABC.

According to above figures, we can conclude that EK-GA has superior performance. Although S-ABC has the best

convergence speed, it always finds the local optimal solutions.

Fig.5(d)-(f) show that with the increasing number of subtasks and candidate services, the performance of DE-caABC and S-ABC with three yardsticks decrease dramatically. In contrast, the performance of EK-GA has a slow decline. EK-GA performs well in all synthesized datasets, which indicates that the proposed algorithm has q better searching capability both in local and global search.

In summary, the above experimental results prove the superiority of EK-GA. It has better local and global searching abilities, whereas DE-caABC and S-ABC are similar to focus on local searching and has a better local search ability. Consequently, the proposed algorithm has better effectiveness in solving ISOC problems with a large-scale of candidate IoT service set.

E. DISCUSSION

With the above experiments, EK-GA is validated to have good performance. Compared with T-GA and other EK-oriented algorithms, with a certain amount of empirical knowledge, it has superior performance in a capability for finding the best solution and a probability of finding the global optimal solution, and has a preferable convergence speed. The reasonable discovery of EK of IoT services plays an important role for EK-GA to search the global optimal solution and might be the key reason that EK-GA can perform better than can other EK-oriented algorithms.

The first experiment has shown that the scale and quality of optimal subspace can influence the convergence speeds, capability of finding the best solution and probability to obtain global optimal solutions. Fig. 5 (b) and (c) shows that when the percentage T is greater than 30%, the probability of optimal subspace containing services of the global optimal solution increases and EK-GA performs well gradually; when the optimal subspace covers sufficient services for the global optimal solution, EK-GA reaches its top performance. Then its performance declines when the optimal subspace contains more unqualified IoT services. The second and third experiments have verified the superiority of EK-GA and further revealed some features of EK-GA. The better local and global searching abilities of EK-GA rely heavily on the quality of optimal subspace, which means that the threshold T of the utility function should be set to an appropriate value for acquiring sufficient prior knowledge. Moreover, thresholds for dividing service space should be adjusted with the threshold T.

The superiority of EK-GA makes it suitable for the ISOC problems in CMfg, particularly for a IoT service composition with many subtasks and candidate IoT services. For such ISOC problems, EK-GA can obtain the optimal solution efficiently and effectively with a certain amount of empirical knowledge. For ISOC problems without historical data, T-GA and other intelligent optimization algorithms can first be used to generate the empirical knowledge for setting up the EK-GA to obtain the most optimal solution. With the help of EK-GA and the easily attained EK from historical practices, manufacturers who are willing to take advantage of IoT servces in CMfg for implementing mass customization or the new industrial revolution based on flexible IoT service composition can arrive at their goals safely and efficiently, with high quality compositional services and products.

VII. CONCLUSIONS AND FUTURE WORK

It is becoming a promising way to implement mass customization by building IoT Service Optimal Compositions (ISOC) in CMgf. However, with ever-increasing candidate IoT services and more complicated IoT execution path, the solving of ISOC problems in CMfg is inefficient.

In this paper, an empirical knowledge-oriented genetic algorithm (EK-GA) is proposed in response to the deficiency of existing methods in addressing this problem. More specifically, based on service domain features and distinctive features of IoT services, some formulas for analysing EK of IoT services are defined, and an algorithm for processing the service space is proposed. Then, with the divided service space, EK-GA is presented with optimization strategies of initial population, operators and fitness function. The effectiveness and efficiency of EK-GA are verified through three types of experiment with ISOC problems in CMfg. EK-GA can be used not only for manufacturers who are attempting to employ IoT service compositions in CMfg for implementing mass customization to cope with the pressure of lower costs and to improve the core competitive ability in the current ''globalization of manufacturing'' environment but also for providing some guidance and insights to CMfg designers.

The main contributions of this paper are summarized as follows:

1) With the purpose of using the empirical knowledge of IoT services to solve ISOC problems, the Ek of IoT services is explored based on SDFs to reasonably divide the service space of ISOC problems.

(2) In order to quickly and fully search the divided service space of ISOC problems in obtaining optimal solutions, we proposed EK-GA algorithm with EK-based optimization strategies. The performance comparisons indicated that the proposed algorithm is better than other EK-based algorithms for solving ISOC problems, especially for solving the ISOC problem with a large-scale search space.

In the future, our work will further refine the definitions of SDFs and strategies of EK-GA by studying the division of subspace for various specific ISOC problems. We will apply EK-GA to solve ISOC problems in several practical manufacturing domains (e.g., automobile manufacturing, electric manufacturing, and zipper manufacturing) to further explore its practical value.

REFERENCES

[1] F. Tao, Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li, ''CCIoT-CMfg: Cloud computing and Internet of Things-based cloud manufacturing service system,'' *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1435–1442, May 2014.

[2] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 147–164, 2016.

[3] Y. Zhang, G. Zhang, Y. Liu, and D. Hu, "Research on services encapsulation and virtualization access model of machine for cloud manufacturing," *J. Intell. Manuf.*, vol. 28, no. 5, pp. 1109–1123, 2017.

[4] J. Mai, L. Zhang, F. Tao, and L. Ren, "Customized production based on distributed 3D printing services in cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 71–83, 2016.

[5] L. Guo and J. Qiu, "Combination of cloud manufacturing and 3D printing: Research progress and prospect," *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 5–8, pp. 1929–1942, 2018.

[6] C. Yang, S. Lan, W. Shen, G. Q. Huang, X. Wang, and T. Lin, "Towards product customization and personalization in IoT-enabled cloud manufacturing," *Cluster Comput.*, vol. 20, no. 2, pp. 1717–1730, 2017.

[7] L. Zhou, L. Zhang, L. Ren, and Y. Laili, "Matching and selection of distributed 3D printing services in cloud manufacturing," in *Proc. 43rd Annu. Conf. Ind. Electron. Soc. (IECON)*, Oct./Nov. 2017, pp. 4728–4733.

[8] C. Yang, W. Shen, T. Lin, and X. Wang, "IoT-enabled dynamic service selection across multiple manufacturing clouds," *Manuf. Lett.*, vol. 7, pp. 22–25, Jan. 2016.

[9] H. Li, K. C. C. Chan, M. Liang, and X. Luo, "Composition of resource-service chain for cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 211–219, Feb. 2016.

[10] X. Xu, Z. Liu, Z. Wang, Q. Z. Sheng, J. Yu, and X. Wang, "S-ABC: A paradigm of service domain-oriented artificial bee colony algorithms for service selection and composition," *Future Gener. Comput. Syst.*, vol. 68, pp. 304–319, Mar. 2017.

[11] J. Zhou and X. Yao, "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.*, vol. 90, nos. 1–4, pp. 1085–1103, 2017.

[12] J. Im, S. Kim, and D. Kim, "IoT mashup as a service: Cloud-based mashup service for the Internet of Things," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun./Jul. 2013, pp. 462–469.

[13] H. Jin, X. Yao, and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," *J. Intell. Manuf.*, vol. 28, no. 8, pp. 1947–1960, 2017.

[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. IEEE 7th Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Nov. 2014, pp. 65–72.

[15] F. Tao, Y. Zuo, L. Da Xu, and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1547–1557, May 2014.

[16] R. Y. Zhong, L. Wang, and X. Xu, "An IoT-enabled real-time machine status monitoring approach for cloud manufacturing," *Procedia CIRP*, vol. 63, pp. 709–714, Jul. 2017.

[17] J. Wan, M. Yi, D. Li, C. Zhang, S. Wang, and K. Zhou, "Mobile services for customization manufacturing systems: An example of industry 4.0," *IEEE Access*, vol. 4, pp. 8977–8986, 2016.

[18] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[19] H.-L. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 68–76, Mar./Apr. 2015.

[20] Y. Lu and J. Cecil, "An Internet of Things (IoT)-based collaborative framework for advanced manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 5–8, pp. 1141–1152, May 2016.

[21] G. White, A. Palade, and S. Clarke, "Qos prediction for reliable service composition in IoT," in *Proc. Int. Conf. Service-Oriented Comput.* Cham, Switzerland: Springer, 2017, pp. 149–160.

[22] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, Dec. 2016.

[23] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, and R. Buyya, "An energy-aware service composition algorithm for multiple cloud-based IoT applications," *J. Netw. Comput. Appl.*, vol. 89, pp. 96–108, Jul. 2017.

[24] I.-R. Chen, J. Guo, and F. Bao, "Trust management for SOA-based IoT and its application to service composition," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 482–495, May/Jun. 2016.

[25] G. Chen, J. Huang, B. Cheng, and J. Chen, "A social network based approach for IoT device management and service composition," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun./Jul. 2015, pp. 1–8.

[26] A. Urbieta, A. González-Beltrán, S. B. Mokhtar, M. A. Hossain, and L. Capra, "Adaptive and context-aware service composition for IoT-based smart cities," *Future Gener. Comput. Syst.*, vol. 76, pp. 262–274, Nov. 2017.

[27] M. E. Khansari, S. Sharifian, and S. A. Motamedi, "Virtual sensor as a service: A new multicriteria QoS-aware cloud service composition for IoT applications," *J. Supercomput.*, 2018, pp. 1–28.

[28] N. Wanigasekara, J. Schmalfuss, D. Carlson, and D. S. Rosenblum, "A bandit approach for intelligent IoT service composition across heterogeneous smart spaces," in *Proc. ACM 6th Int. Conf. Internet Things.*, 2016, pp. 121–129.

[29] H. Zheng, Y. Feng, and J. Tan, "A hybrid energy-aware resource allocation approach in cloud manufacturing environment," *IEEE Access*, vol. 5, pp. 12648–12656, 2017.

[30] D. Arellanes and K.-K. Lau, "Algebraic service composition for user-centric IoT applications," in *Proc. Int. Conf. Internet Things*. Cham, Switzerland: Springer, 2018, pp. 56–69.

[31] R. Yang, B. Li, and C. Cheng, "A petri net-based approach to service composition and monitoring in the IOT," in *Proc. Asia–Pacific. Services Comput. Conf. (APSCC)*, Dec. 2014, pp. 16–22.

[32] L. Zhang, S. Yu, X. Ding, and X. Wang, "Research on IOT RESTful Web service asynchronous composition based on BPEL," in *Proc. 6th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, vol. 1, Aug. 2014, pp. 62–65.

[33] G. Tzortzis and E. Spyrou, "A semi-automatic approach for semantic IoT service composition," in *Proc. Workshop Artif. Intell. Internet Things Conjunction SETN*, 2016, pp. 1–6.

[34] J. Liu *et al.*, "Research on Web service dynamic composition based on execution dependency relationship," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun./Jul. 2016, pp. 113–117.

[35] F. Xiang, G. Z. Jiang, L. Xu, and N. Wang, "The case-library method for service composition and optimal selection of big manufacturing data in cloud manufacturing system," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 59–70, 2016.

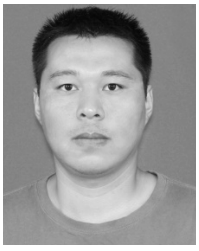
[36] N. S. Van, H. D. Vo, and P. N. Hung, "A correlation-aware negotiation approach for service composition," in *Proc. ACM 6th Int. Symp. Inf. Commun. Technol.*, 2015, pp. 210–216.

[37] G. Hua *et al.*, "A discovery method of service-correlation for service composition in virtual enterprise," *Eur. J. Ind. Eng.*, vol. 8, no. 5, pp. 579–618, 2014.

[38] M. Bravo, "Similarity measures for Web service composition models," *Int. J. Web Service Comput.*, vol. 5, no. 1, pp. 495–505, 2014.

[39] R. Karim, C. Ding, and A. Miri, "End-to-end QoS prediction of vertical service composition in the cloud," in *Proc. IEEE 8th Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2015, pp. 229–236.

[40] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 382–393, May/Jun. 2016.



TIANYANG LI received the M.S. degree from the School of Computer and Information Technology, Northeast Petroleum University, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include service computing, cloud manufacturing, Internet of Things, and intelligent optimization algorithms.

TING HE received the Ph.D. degree from the Department of Mechanical and Electronic Engineering, Harbin Institute of Technology, in 2000. He was with the School of Computer Science and Technology, Harbin Institute of Technology, from 2000 to 2016. He is currently a Professor with the College of Computer Science and Technology, Huaqiao University, where he is involved in focuses on artificial intelligence and service computing.

ZHONGJIE WANG (M'08) received the Ph.D. degree in computer science from the Harbin Institute of Technology in 2006. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. He has authored over 40 publications. His research interests include services computing, mobile and social networking services, and software architecture.

YUFENG ZHANG received the B.Eng. degree in mechanical engineering and the B.Eng. degree in international finance from Shanghai Jiao Tong University in 1998, the Dipl.Ing. and M.Sc. degrees (Hons.) in global production engineering from the Technical University of Berlin in 2004, and the Ph.D. degree in engineering from the University of Cambridge in 2008. His early industrial career included responsibilities for project management and new business development. He continues working very closely with industry in his research and teaching activities. He is currently an Associate Professor (SL) in operations management for global engineering networks with the University of Birmingham.