Int J Adv Manuf Technol (2017) 88:2757–2771
DOI 10.1007/s00170-016-8992-7

ORIGINAL ARTICLE

QoS-aware service composition for cloud manufacturing based on the optimal construction of synergistic elementary service groups

Bo Liu[1] **· Zili Zhang**[1]

Received: 17 November 2015 / Accepted: 30 May 2016 / Published online: 8 June 2016

Abstract Cloud manufacturing (CMfg) has drawn extensive attentions from industrial community and academia. Quality of service (QoS)-aware service composition is critical to the on-demand use of distributed manufacturing resources and capabilities in CMfg systems. However, most previous work plainly composed composite services by the approach of one-to-one mapping-based service composition (OOM-SC), which leads to drawbacks to both the overall QoS of composite services and the success rate of service composition. To circumvent this, an approach of *synergistic elementary service group-based service composition* (*SESG-SC*) is proposed in this paper. It releases the assumption of one-to-one mapping between elementary services and subtasks, allowing a free combination of multiple functionally equivalent elementary services into a *synergistic elementary service group* (*SESG*) to perform each subtask collectively, thereby bettering the overall QoS and achieving more acceptable success rate. To introduce an optimal construction of SESGs into the optimization model of QoS-aware service composition, three kinds of redundant structures within SESGs are discussed and the corresponding QoS evaluation formulas are also proposed. To deal with the increasing computing complexity of the optimization model, an algorithm named matrix-coded genetic algorithm with collaboratively evolutional populations (MCGA-CEP) is designed in the current study. The experimental results indicate that the proposed SESG-SC approach significantly outperforms the previous approaches, and the proposed MCGA-CEP is sound on performance-wise.

Keywords Cloud manufacturing (CMfg) · Service composition · Quality of service (QoS) · Synergistic elementary service group (SESG) · Genetic algorithm

1 Introduction

Cloud manufacturing (CMfg) is a promising new manufacturing paradigm [1–3], which evolves from some advanced manufacturing models, such as agile manufacturing (AM) [4], application service provider (ASP) [5], and manufacturing grid (MGrid) [6]. It reaps the benefits of cloud computing [7, 8], service-oriented technologies (SOTs) [9], Internet of things (IoT) [10–15], and big data [16–18] to provide fully scale sharing, free circulation and transaction, and on-demand use of globally distributed manufacturing resources and capabilities [19].

As a cloud-based manufacturing paradigm, CMfg enriches the philosophy of "everything as a service" in cloud computing, e.g., infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), by introducing new thoughts of "manufacturing resource as a service" and "manufacturing capability as a service" into CMfg systems [20]. In a CMfg system, a great number of manufacturing resources and capabilities provided by various *service providers* (*SPs*) are encapsulated and virtualized as elementary *manufacturing cloud services* (*MCSs*). Afterward, functionally equivalent MCSs are classified into the same *candidate MCS set* (*CMCSS*). As a *service demander* (*SD*) submits his manufacturing task to the system and requests manufacturing services, a series of MCSs are selected from respective CMCSSs and assembled into a virtual manufacturing

✉ Bo Liu
liubocq@swu.edu.cn

1 Faculty of Computer and Information Science, Southwest University, No. 2 Tiansheng Road, Beibei District, Chongqing 400715, People's Republic of China

environment (i.e., a *manufacturing composite cloud service* (*MCCS*) in CMfg systems) to respond and execute that task [21].

Similar to the MGrid system [22], manufacturing tasks in a CMfg system are mostly *multi-functionality manufacturing tasks* (*MFMTs*). To execute such kind of tasks, an iterative process of task decomposition is firstly performed to divide the MFMT into a series of subtasks, until at least one candidate MCS is available to any of the subtasks. To perform each subtask and ultimately complete the MFMT, appropriate selection of MCSs should be made from respective CMCSSs and followed by an integration of the selected MCSs into a MCCS, which embraces both the multi-functionality requirements and the quality of service (QoS) constraints (e.g., time to market, cost, reliability) [23]. That is the process of QoS-aware service composition.

As the number of CMCSSs involved in the above process is large and that of MCSs in each CMCSS may be even larger, a decision should be made to determine which MCSs from the respective CMCSSs should be selected and orchestrated into a MCCS, such that the overall QoS of the MCCS is optimized and the QoS constraints given by a SD are fulfilled. That results in an optimization problem of QoS-aware service composition in CMfg systems.

In the past few years, extensive investigations regarding this optimization problem have been launched in both CMfg systems and other service-oriented architecture (SOA)-based systems [24]. However, most of the previous studies obtained the optimal composite service based on the assumption of *one-to-one mapping* (*OOM*) between tasks and services; we call it the approach of *one-to-one mapping-based service composition* (*OOM-SC*) in the current study. The OOM-SC approach has two drawbacks as it is applied to CMfg systems:

(1) *Insufficient optimization of the overall QoS of MCCSs*. Based on the basic assumption in OOM-SC, only one appropriate MCS is allowed to perform each subtask of a MFMT, while other functionally equivalent MCSs available to the same subtask are kept on the shelf. Since making the best use of available MCSs fails, the optimal MCCS obtained by the OOM-SC approach can hardly touch the ceiling of the real "best QoS." The findings of our previous work suggest that combining composite services (MCCSs) as a whole to perform each MFMT can significantly improve the outcome of overall QoS [25]. Accordingly, an underlying presumption can be drawn out analogically that binding elementary services (MCSs) together to perform each subtask may also contribute to the same dramatic or even further improvement of the overall QoS.
(2) *Apt to fail in the scenarios of severe QoS constraints* (e.g., limited cost budget or stringent time limit). The severe QoS constraints indicate that the given QoS constraints outnumber the overall QoS values of all plainly composed MCCSs. A plainly composed MCCS stands for a MCCS without any redundant structures in each of its component services. In other words, each component service of a MCCS is directly linked to a MCS; as a kind of elementary service, one single MCS has relatively limited QoS parameters comparing with the redundant structures which combine several MCSs as a whole. Accordingly, if a SD submits a MFMT with severe QoS constraints, no solutions can be identified by the approach of OOM-SC and service composition fails. Actually, the scenarios of severe QoS constraints are quite common in real-world manufacturing business, especially as the MFMTs are big deals and the SPs of available MCSs are small and medium enterprises (which are known to be of very limited resources and capabilities).

Therefore, in order to further improve the overall QoS of the ultimately obtained MCCS and achieve a higher success rate of service composition especially in the scenarios of severe QoS constraints, an approach of *synergistic elementary service group-based service composition* (*SESG-SC*) is proposed in this paper. It releases the assumption of OOM, allowing a free integration of multiple functionally equivalent MCSs into a *synergistic elementary service group* (*SESG*) to perform each subtask, thereby composing a MCCS based on the optimal construction of SESGs to complete a MFMT with better QoS outcome and more acceptable success rate. Additionally, in order to deal with the increasing computing complexity of the problem of QoS-aware service composition coming with the SESG-SC approach, an algorithm named matrix-coded genetic algorithm with collaboratively evolutional populations (MCGA-CEP) is also proposed in the current study.

The remainder of the paper is organized as follows: Section 2 analyses the related work. Section 3 presents an example to illustrate the framework and problem of SESG-SC. Section 4 establishes a problem model of QoS-aware service composition based on the SESG-SC approach. Section 5 presents a MCGA-CEP-based implementation to solve the problem. The experimental results are documented in Section 6, and Section 7 concludes this work.

2 Related works

CMfg is proposed in response to the increasing needs for IT-reliant, globalized, distributed, and agile-demanding manufacturing business. As a promising new paradigm, CMfg has drawn extensive attentions from industrial community and academia [26–29]. Notable efforts have been expended on the investigation of the following: (1) concept and architecture [30], (2) enabling technologies [31–34], and (3) application prototype platform [35–37].

Recently, inspired by the application of SOT to service-oriented computing/manufacturing systems [38–40], applying SOT to CMfg systems has been of wide theoretical and application-based interests, where the issue of QoS-aware service composition for CMfg has become an active field of research [41–43]. As the vital factors to reflect the performance of QoS-aware service composition, overall QoS and success rate placed a constant emphasis in CMfg systems. In order to improve the overall QoS of a composite service (MCCS) and the success rate of service composition, research efforts have been undertaken in two main directions: (1) composition approaches and (2) optimization models.

The first direction addresses the problem of how to compose a composite service. Traditionally, a composite service was plainly composed in a OOM pattern, which contained the following features [44, 45]: (1) each component service on the execution path of the composed composite service strictly matches the corresponding one subtask in the workflow of the given task, (2) the execution path has the same topology as the workflow, and (3) each component service is plainly related to an appropriately chosen elementary service (MCS). Although plain composition was widely employed in most cases of QoS-aware service composition, a plainly composed composite service can hardly make rapid improvements of overall QoS and success rate [25].

Addressing the drawback of plain composition, two methods were mainly used: (1) creating interior redundant structures in a composite service. It indicated that the whole execution path of the composed composite service can be evolved by adding parallel or selective subpaths. This method was initially introduced in the realm of dynamic evolution of composite services [46]. A set of evolution operations, such as choice addition and parallel addition, were proposed to create redundant execution paths, thereby guaranteeing and improving the QoS properties of reliability and availability. The above findings enlightened our study on how to construct SESGs and provided the usable construction operations. However, the related research above placed more weight on the theoretical verification of the soundness of redundant structures, and little emphasis was attached to the issue of how to evaluate the created redundant structures with QoS properties. As a result, interior redundant structures still remained absent in the optimization models of QoS-aware service composition. (2) Creating exterior redundant structures among several composite services. Actually, it was a service binding method that combined incompetent composite services as a whole to perform each given task, which has been verified to be of better overall QoS and success rate. The approach of "multi-composition for each task"-based service composition (MCET-SC) was such a typical use [25]. Nevertheless, exterior redundant structures or MCET-SC was a kind of coarse granularity service binding, which required combining the entire MCCSs rather than just binding the needed elementary services. Accordingly, the effectiveness of MCET-SC may decrease as incompetent MCSs were included in the bound MCCSs.

For the second direction, optimization models have a direct impact on the overall QoS and success rate. Previous investigations built their optimization models mainly based on (1) local planning [44, 47], (2) global planning [19, 22, 39], and (3) hybrid planning [48–51], on a basis of the OOM-SC approach. The local planning-based model separately selects the elementary service of the best QoS from respective candidate services set to perform each subtask. This model has an advantage of efficiency (i.e., time-saving) but a disadvantage of effect (i.e., poor overall QoS). In contrast, the global planning-based model selects an elementary service according to its contribution to the overall QoS of a composite service. This model has been explored in extensive studies and verified to be highly beneficial to achieving better QoS of a composite service. Just as pointed out by Ardagna and Pernici [38], however, the existed global planning-based models failed to deal with a task with severe QoS constraints. The recently proposed hybrid method built a holistic optimization model based on the two-staged planning: (1) decomposition of global QoS constraints into local constraints and (2) distributed local optimal selection under local QoS constraints. Although the model benefits from both advantages of local and global planning in time-saving and fine overall QoS, respectively, it cannot contribute to the better outcome of overall QoS (theoretically less than or equal to) as opposed to the global planning-based model.

Specially, to tackle the problem of success rate decline under severe QoS constraints, various kinds of compensation mechanism were proposed in response to the service composition failure. Ardagna and Pernici [38] designed a negotiation mechanism to bargain QoS constraints as no feasible solution can be found. Lin et al. [52] presented a similar mechanism of QoS constraints relaxation, and He et al. [53] introduced an iterative multi-attribute combinatorial auction-based quality-aware service selection to improve the success rate of service composition. However, the aforementioned mechanism of negotiation, relaxation, and auction succeeded in ensuring success rate but failed in preserving overall QoS, because the OOM-SC approach was used as the footstone to generate feasible solutions (i.e., composite services). The drawback of insufficient optimization within an OOM-SC approach resulted in an "improvable area of QoS," which lied in between the traditional unfeasible condition and the real unfeasible condition. In fact, it is highly probable that feasible solutions can be found in such area via the SESG-SC approach without executing any compensation mechanism.

In consideration of the algorithms to optimize the QoS-aware service composition problem, integer programming

(IP) was widely used [38, 49]. Theoretically, QoS-aware service composition problem is NP-hard. As a result, the IP-based approaches can hardly fulfill the requirement of scalability for realistic problems with large search spaces. Therefore, a lot of heuristic algorithms with polynomial or pseudo-polynomial time complexity are designed to find acceptable near-to-optimal solutions, such as particle swarm-based approaches [22], tabu search and simulated annealing integrated approaches [54], and genetic algorithm (GA)-based approaches [51, 55–57]. With regard to the GA-based approaches, many improved algorithms were proposed, such as GA combined with other heuristic algorithms [51] and matrix-coded genetic algorithm (MCGA) [58]. The MCGA-based approaches were reported to be of superior performance to the one-dimensional coded GAs in solving the QoS-aware service composition problem [58]. Moreover, the idea of parallel design of algorithms in multi-core processors made it possible to achieve improved performance [19]. Benefiting from the previous work, extension is made by this paper via designing new operators in consideration of the thorough information exchange based on the matrix coding manner, and introducing parallel populations into the algorithm, in order to further improve the performance.

3 Motivating example

In order to illustrate the issue addressed in this work, an example about "online motorcycle production (OMP)" is presented in the context of CMfg. The flow of OMP is shown in Fig. 1, where six key sections (subtasks) are considered.The proposed framework of SESG-SC approach for OMP is shown in Fig. 2.As a SD submits an OMP task (T), the following steps are performed firstly:

- *Task decomposition*: the step is conducted to divide a MFMT into several subtasks, such that at least one candidate MCS can be discovered for each subtask. According to the flow of OMP shown in Fig. 1, six subtasks can be identified in the decomposition process as follows: (1) frame production and preassembly (FPP), (2) engine production and preassembly (EPP), (3) parts production and preassembly (PPP), (4) final assembly (FA), (5) motorcycle testing (MT), and (6) motorcycle packing (MP). Let ST_i denotes the ith subtask of task T, $i \in \{1, 2, \ldots, I\}$, then T can be expressed as $T = \{ST_1, ST_2, \ldots, ST_I\}$. if $I = 1$, the task T become a single-functionality manufacturing task and can be simply responded by optimal selection of one MCS [22]. Obviously, T is a MFMT because $I = 6$ in the current scenario and $\mathrm{ST}_1, \ldots, \mathrm{ST}_6$ denote the subtasks of FPP,…, MP, respectively.
- *Service discovery*: this step is carried out to find out all candidate MCSs for each subtask and pool them into the related candidate service set. Let CMCSS_i denotes the candidate service set for the ith subtask ST_i, and MCS_{ij} indicates the jth MCS in CMCSS_i, $j \in \{1, 2, \ldots, J\}$, then $\mathrm{CMCSS}_i = \{\mathrm{MCS}_{i1}, \mathrm{MCS}_{i2}, \ldots, \mathrm{MCS}_{iJ}\}$

The above two steps are the preparation procedures for service composition. According to the approach of OOM-SC, the next step is service composition. In this step, the composite service MCCS is composed by appropriate selections of a series of MCSs from each CMCSS, respectively. Different from the approach of OOM-SC, the proposed SESG-SC approach introduces an additional step before executing the step of service composition as shown in Fig. 2.

- *SESG construction* (*additional step*): the step generates an appropriate SESG for each subtask based on the related CMCSS. This step has two operations: (1) *selection of elementary services*: let SESG_i denotes the corresponding synergistic group for the ith subtask ST_i, and then several MCSs from CMCSS_i are appropriately selected into the set of SESG_i, i.e., $\mathrm{SESG}_i = \{\mathrm{MCS}_{ij} | \mathrm{MCS}_{ij} \in \mathrm{CMCSS}_i\}$ s.t., $\mathrm{SESG}_i \subseteq \mathrm{CMCSS}_i$. (2) *Decision on the structure of SESG*: based on the selected MCSs, a SESG can be constructed into three kinds of redundant structures, that is, parallel, selective, and hybrid. A wise choice should be made to identify which kind of structure embraces the most appropriate QoS criteria to undertake the subtask ST_i.
- *Service composition based on SESGs* (*adapted step*): different from the approaches of OOM-SC and MCET-SC, which both compose a MCCS by directly using elementary MCSs, the SESG-SC approach creates a MCCS based on the optimal construction of SESGs for each subtask and integration of these SESGs into the MCCS, such that the overall QoS of the MCCS is optimized and the QoS requirements of SDs are fulfilled.

In order to perform OMP in a CMfg system by using the SESG-SC approach, the following issues must be considered:

(1) How to optimally construct a SESG with MCSs and how different redundant structures influence the QoS evaluation of a SESG.
(2) How to establish the optimization model based on SESGs rather than elementary services (MCSs).
(3) How to deal with the increasing computing complexity of the optimization model coming with the SESG-SC approach.

This paper emphasizes on solving the above issues.

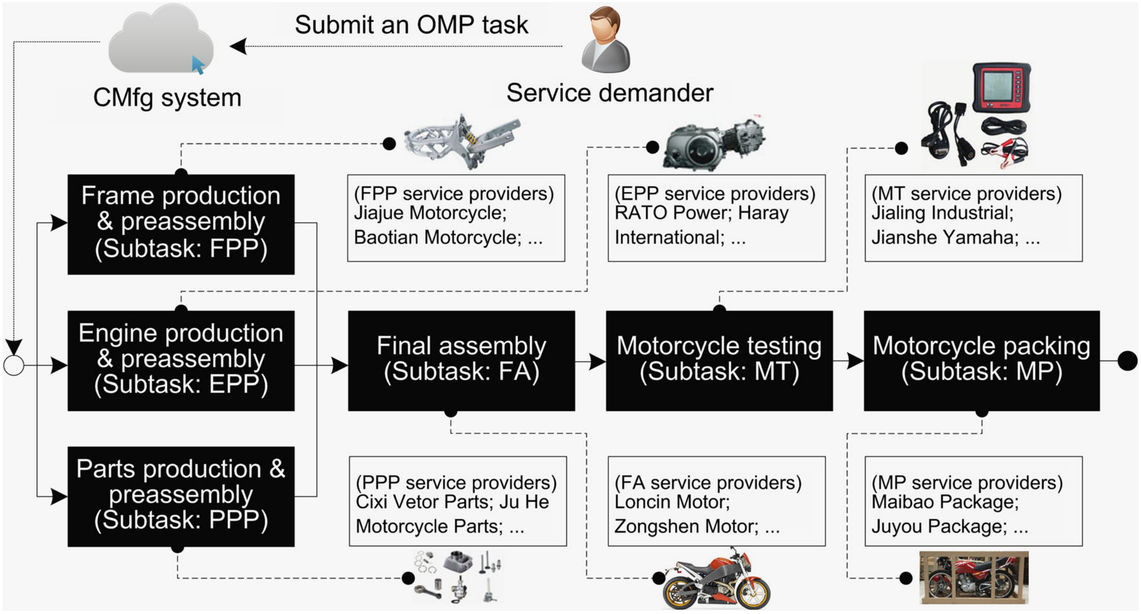


Fig. 1 Flow of online motorcycle production (*OMP*)

4 Formulation of SESG-SC in CMfg

We consider a SESG-based service composition in a CMfg system. In line with the principle of the new approach, several functionally equivalent MCSs can be combined as a whole (SESG) to execute a subtask of a MFMT, which removes the assumption of one-to-one mapping between MCSs and subtasks, so as to enhance the QoS performance and success rate of service composition.

For the sake of illustration, typical QoS properties in [45] (i.e., time, cost, and reliability) are used as the criteria to evaluate the QoS of MCSs, SESGs, and MCCS. More properties can also be added without fundamentally altering the model and related algorithm.

4.1 Notations

To begin modeling the problem, let us use the following notations throughout the paper:

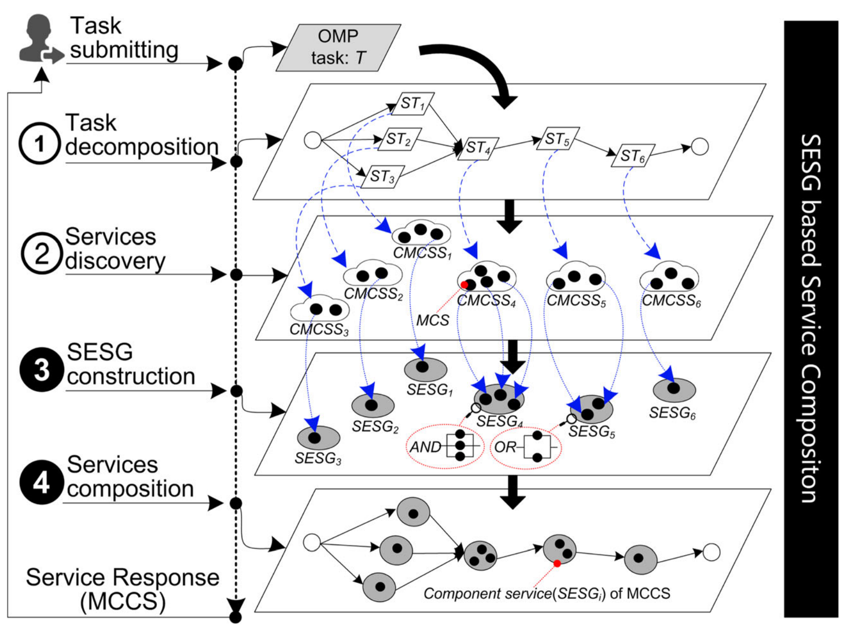
(1) Indices

i 1,…, I: index of subtasks or CMCSSs;

j 1,…, J: index of MCSs in a CMCSSs;

k 1,…, K: index of QoS properties;

Fig. 2 Framework of the SESG-SC approach



(2) Decision variable

x_{ij} is equal to 1, if the jth elementary service MCS_{ij} is selected from the ith candidate service set $CMCSS_i$ and combined into the part of the *selective structure* in the synergistic group $SESG_i$ to execute the corresponding subtask ST_i;

It is equal to −1, if the jth elementary service MCS_{ij} is selected to join the part of the *parallel structure* in the synergistic group $SESG_i$;

It is equal to 0, if the jth elementary service MCS_{ij} is not selected.

(Note that we consider a more general case of *hybrid structure* in the construction of any $SESG_i$, which contains two parts of selective and parallel structures inside. If there is no selective structure within $SESG_i$, the hybrid structure becomes a parallel structure and vice versa. More details about the structures of SESGs can be found in Section 4.2).

(3) Parameters

$Q(T)$ is the vector of QoS requirements on a manufacturing task MFMT submitted by SDs. Let $q_k(T)$ denotes the value of a QoS property, then $Q(T) = (q_1(T), q_2(T), \ldots, q_K(T))$. As we consider the QoS properties of time, cost, and reliability in this model, the vector can be expressed as $Q(T) = (q_1(T), q_2(T), q_3(T))$, where $q_1(T)$, $q_2(T)$, and $q_3(T)$ represent the QoS requirements of a MFMT on time, cost, and reliability, respectively.

$Q(MCS_{ij})$ is the vector of QoS evaluation values of a MCS. Let $q_m(MCS_{ij})$ denotes the evaluation value on a QoS property, then $Q(MCS_{ij}) = (q_1(MCS_{ij}), q_2(MCS_{ij}), q_3(MCS_{ij}))$, where $q_1(MCS_{ij})$, $q_2(MCS_{ij})$, and $q_3(MCS_{ij})$ stand for the QoS values of a MCS on time, cost, and reliability, respectively.

W is the vector of preference weights on the QoS properties of time, cost, and reliability given by a SD's QoS requirements. Let w_r denotes the preference weight (defined as a percentage) on a QoS property, then $W = (w_1, w_2, w_3)$ and $\sum_{r=1}^{3} w_r = 1$, where w_1, w_2, and w_3 are the preference weights on time, cost, and reliability, respectively.

4.2 SESG construction

We assume the preparation procedures for service composition have been performed. Then, the subtasks of the given MFMT have been obtained and the respective CMCSS for each subtask has been produced and pooled the related candidate MCSs. Based on the above outputs, the SESG for each subtask can be constructed.

Definition 1 (*synergistic elementary service group*, SESG): a SESG is a redundant structure of middleware service (compared to the final form of composite service MCCS and the initial form of elementary service MCS) for the execution of a subtask, which allows several functionally equivalent MCSs to be combined as a whole to perform each subtask of a MFMT collectively, in order to obtain better QoS performance.

Two operations proposed by Zeng et al. [46], namely the choice and parallel additions, which were once used to create redundant structures for a composite service, are now introduced into construction of SESGs. Two basic redundant structures (i.e., selective and parallel structures) as well as an extended redundant structure (i.e., hybrid structure) can be produced in a SESG by using those operations. Furthermore, a quantitative measure of the redundant structures is introduced in the current study to evaluate SESGs with QoS properties. Let "‖" and "∘ ∘" denote the choice addition and parallel addition, respectively, and the formulas of redundant structures are designed as follows:

(1) *Formula of selective structure* (for the SESGs constructed by choice addition): suppose that a selective structural $SESG_i$ for ST_i is combined with several elementary MCSs selected from the candidate service set $CMCSS_i$, then $SESG_i = \left\{ \underset{j=1..J}{\|} MCS_{ij} \middle| MCS_{ij} \in CMCSS_i \text{ and } x(i,j) = 1 \right\}$, and each MCS in the $SESG_i$ has a probability of being selected to undertake the subtask ST_i independently. Let percentage p_{ij} denotes the probability of a MCS_{ij} to undertake the subtask ST_i, where $p_i = \sum_{j=1}^{J} p_{ij} = 1$, indicating that there is always a MCS selected for ST_i when viewed as a whole. Accordingly, as the QoS evaluation values of each MCS_{ij} are given by parameters, the QoS of $SESG_i$ which is constructed with these MCSs by choice addition can be expressed as follows:

$$\text{Define} \quad x_{\text{seq}} = \begin{cases} x_{ij}, & \text{if } x_{ij} = 1 \\ 0, & \text{otherwise} \end{cases}, \quad \text{then}$$
$$Q(SESG_i) = (q_1(SESG_i), q_2(SESG_i), q_3(SESG_i))$$
$$\text{where} \begin{cases} q_1(SESG_i) = \sum_{j=1}^{J} q_1(MCS_{ij}) \times x_{\text{seq}} \times p_{ij} \\ q_2(SESG_i) = \sum_{j=1}^{J} q_2(MCS_{ij}) \times x_{\text{seq}} \times p_{ij} \\ q_3(SESG_i) = \sum_{j=1}^{J} q_3(MCS_{ij}) \times x_{\text{seq}} \times p_{ij} \end{cases} \tag{1}$$

(2) *Formula of parallel structure* (for the SESGs constructed by parallel addition): suppose a parallel structural $SESG_i$ for ST_i is combined with a set of $SESG_i = \left\{ \underset{j=1..J}{\circ\circ} MCS_{ij} \middle| MCS_{ij} \in CMCSS_i \text{ and } x(i,j) = -1 \right\}$. We assume that each MCS has the equal access to execute ST_i, and the load of ST_i is divided and assigned to each MCS totally according to its time efficiency to complete

the ST_i. Therefore, the QoS of SESG_{ik} constructed by parallel addition is calculated using the following formula:

$$\text{Define} \quad x_{\mathrm{par}} = \begin{cases} |x_{ij}|, & \text{if } x_{ij} = -1 \\ 0, & \text{otherwise} \end{cases}, \quad \text{then}$$
$$Q(\mathrm{SESG}_i) = (q_1(\mathrm{SESG}_i), q_2(\mathrm{SESG}_i), q_3(\mathrm{SESG}_i))$$
$$\text{where} \begin{cases} q_1(\mathrm{SESG}_i) = 1 \Big/ \left(\sum_{j=1}^{J} \left(\dfrac{1}{q_1(\mathrm{MCS}_{ij})} \times x_{\mathrm{par}} \right) \right) \\ q_2(\mathrm{SESG}_i) = \sum_{j=1}^{J} \left(q_2(\mathrm{MCS}_{ij}) \times \dfrac{q_1(\mathrm{SESG}_i)}{q_1(\mathrm{MCS}_{ij})} \times x_{\mathrm{par}} \right) \\ q_3(\mathrm{SESG}_i) = \prod_{j=1}^{J} \delta; \ \delta = \begin{cases} 1, & \textit{if } x_{\mathrm{par}} = 0 \\ q_3(\mathrm{MCS}_{ij}), & \text{otherwise} \end{cases} \end{cases} \tag{2}$$

(3) *Formula of hybrid structure* (for the SESGs constructed by both parallel and choice additions): the hybrid structure is an extended redundant structure formed by both selective and parallel structures. As a more general case than the SESGs constructed by basic redundant structures, a hybrid structural SESG_i can be divided into two parts: the part of the selective structure and the part of the parallel structure. If any part of selective structure and parallel structure is not existed, the hybrid structure is specialized into a basic structure of selective or parallel. We can calculate the QoS of each part independently by using the formulas of selective and parallel structures, and obtain the overall QoS of SESG_i via combining these two QoS values by using the formula of parallel structure (because these two parts form a SESG through a parallel addition on the whole).

$$\text{Let } \mathrm{sel}(\mathrm{SESG}_i) \text{ and } \mathrm{par}(\mathrm{SESG}_i) \text{ denote the selective and parallel parts of } \mathrm{SESG_i} \text{ respectively, then,}$$
$$Q(\mathrm{SESG}_i) = Q(\mathrm{sel}(\mathrm{SESG}_i) \ \lozenge \lozenge \ \mathrm{par}(\mathrm{SESG}_i)) = (q_1(\mathrm{SESG}_i), q_2(\mathrm{SESG}_i), q_3(\mathrm{SESG}_i))$$
$$\text{where} \begin{cases} q_1(\mathrm{SESG}_i) = 1 \Big/ \left(\dfrac{1}{q_1(\mathrm{sel}(\mathrm{SESG}_i))} + \dfrac{1}{q_1(\mathrm{par}(\mathrm{SESG}_i))} \right) \\ q_2(\mathrm{SESG}_i) = q_2(\mathrm{sel}(\mathrm{SESG}_i)) \times \dfrac{q_1(\mathrm{SESG}_i)}{q_1(\mathrm{sel}(\mathrm{SESG}_i))} + q_2(\mathrm{par}(\mathrm{SESG}_i)) \times \dfrac{q_1(\mathrm{SESG}_i)}{q_1(\mathrm{par}(\mathrm{SESG}_i))} \\ q_3(\mathrm{SESG}_i) = q_3(\mathrm{sel}(\mathrm{SESG}_i)) \times q_3(\mathrm{par}(\mathrm{SESG}_i)) \end{cases} \tag{3}$$

4.3 MCCS composition

In the approach of SESG-SC, The composition of MCCS is based on the optimal selection of SESGs that are constructed in the former step. A well-structured execution path of a MFMT can be specified as a collection of instances of four basic aggregation patterns, that is, sequence pattern, parallel pattern, selective pattern, and circular pattern. Accordingly, the MCCS has the same structure of aggregation patterns as the MFMT, because each subtask in the execution path always has a corresponding component service to undertake it. In the approaches of OOM-SC and MCET-SC, a component service is linked to an elementary service (MCS), while in SESG-SC, a component service is dynamically bond to a SESG.

It should be noted that the parallel or selective pattern here is not the same concepts as the aforementioned selective or parallel structure within a SESG. That is because the former (aggregation patterns) expresses the structural relationships between different subtasks or services with *different functionalities* in a MFMT or MCCS; by contrast, the latter (redundant structures) characterizes the structural properties of several *functionally equivalent* elementary services (MCSs) within a SESG. The aggregation formulas of those patterns have been studied by Tao et al. [22] and Cardoso et al. [45] and applied to calculate the QoS of a MCCS composed with elementary services (MCSs). Let $\mathrm{in}_{\mathrm{seq}}$, $\mathrm{in}_{\mathrm{par}}$, $\mathrm{in}_{\mathrm{sel}}$, and $\mathrm{in}_{\mathrm{cir}}$ denote the instance of sequence, parallel, selective, and circular patterns, respectively. As the SESGs take the place of elementary services (MCSs) in composing a MCCS, the aggregation formulas can be adapted as shown in Table 1 to calculate the QoS of SESG-based MCCSs.

A MCCS may contain several instances of aggregation patterns, and these instances form the MCCS following a sequence pattern on the whole. Taking OMP as an example, two instances of sequence pattern and parallel pattern make up the MCCS of OMP and are composed as a whole based on the overall sequence pattern (shown in Fig. 1). Let in_i denotes an

Table 1 Adapted aggregation formulas for QoS evaluation of the SESG-based MCCS

Aggregation pattern	Aggregation formula	
Sequence pattern	if $\mathrm{in}_{\mathrm{seq}}$ of the sequence pattern is constructed with the set of $\{\mathrm{SESG}_{i_1}, \ldots, \mathrm{SESG}_{i_n}\}$ $Q(in_{\mathrm{seq}}) = \begin{cases} q_1(\mathrm{in}_{\mathrm{seq}}) = \sum_{i=i_1}^{i_n} q_1(\mathrm{SESG}_i) \\ q_2(\mathrm{in}_{\mathrm{seq}}) = \sum_{i=i_1}^{i_n} q_2(\mathrm{SESG}_i) \\ q_3(\mathrm{in}_{\mathrm{seq}}) = \prod_{i=i_1}^{i_n} q_3(\mathrm{SESG}_i) \end{cases}$	(4)
Parallel pattern	if $\mathrm{in}_{\mathrm{par}}$ of the parallel pattern is constructed with the set of $\{\mathrm{SESG}_{i_1}, \ldots, \mathrm{SESG}_{i_n}\}$ $Q(\mathrm{in}_{\mathrm{par}}) = \begin{cases} q_1(\mathrm{in}_{\mathrm{par}}) = \max\limits_{i=i_1 \ldots i_n} (q_1(\mathrm{SESG}_i)) \\ q_2(\mathrm{in}_{\mathrm{par}}) = \sum_{i=i_1}^{i_n} q_2(\mathrm{SESG}_i) \\ q_3(\mathrm{in}_{\mathrm{par}}) = \prod_{i=i_1}^{i_n} q_3(\mathrm{SESG}_i) \end{cases}$	(5)
Selective pattern	if $\mathrm{in}_{\mathrm{sel}}$ of the selective pattern is constructed with the set of $\{\mathrm{SESG}_{i_1}, \ldots, \mathrm{SESG}_{i_n}\}$ Let λ_i denote the probability of $\mathrm{SESG_i}$ being selected, s.t. $\sum_{i=i_1}^{i_n} \lambda_i = 1$ $Q(\mathrm{in}_{\mathrm{sel}}) = \begin{cases} q_1(\mathrm{in}_{\mathrm{sel}}) = \sum_{i=i_1}^{i_n} q_1(\mathrm{SESG}_i) \times \lambda_i \\ q_2(\mathrm{in}_{\mathrm{sel}}) = \sum_{i=i_1}^{i_n} q_2(\mathrm{SESG}_i) \times \lambda_i \\ q_3(\mathrm{in}_{\mathrm{sel}}) = \sum_{i=i_1}^{i_n} q_3(\mathrm{SESG}_i) \times \lambda_i \end{cases}$	(6)
Circular pattern	if $\mathrm{in}_{\mathrm{cir}}$ of circular pattern is constructed with the set of $\{\mathrm{SESG}_{i_1}, \ldots, \mathrm{SESG}_{i_n}\}$ Let ω denote the circle number, then $Q(\mathrm{in}_{\mathrm{cir}}) = \begin{cases} q_1(\mathrm{in}_{\mathrm{cir}}) = \omega \times \left(\sum_{i=i_1}^{i_n} q_1(\mathrm{SESG}_i)\right) \\ q_2(\mathrm{in}_{\mathrm{cir}}) = \omega \times \left(\sum_{i=i_1}^{i_n} q_2(\mathrm{SESG}_i)\right) \\ q_3(\mathrm{in}_{\mathrm{cir}}) = \prod_{i=i_1}^{i_n} q_3(\mathrm{SESG}_i) \end{cases}$	(7)

instance of any aggregation patterns that are constructed with SESGs; suppose a MCCS is composed with a set of instances of aggregation patterns $\{\mathrm{in}_1, \mathrm{in}_2, \ldots, \mathrm{in}_N\}$; as the QoS of a SESG can been obtained in SESG construction, the QoS of the instances is then calculated according to Eqs. (4)–(7) in Table 1, thereby working out the QoS of a MCCS

$$Q(\mathrm{MCCS}) = (q_1(\mathrm{MCCS}), q_2(\mathrm{MCCS}), q_3(\mathrm{MCCS})) \\ \text{where} \begin{cases} q_1(\mathrm{MCCS}) = \sum_{i=1}^{N} q_1(\mathrm{in}_i) \\ q_2(\mathrm{MCCS}) = \sum_{i=1}^{N} q_2(\mathrm{in}_i) \\ q_3(\mathrm{MCCS}) = \prod_{i=1}^{N} q_3(\mathrm{in}_i); \end{cases} \tag{8}$$

4.4 Formulation of SESG-SC problem

4.4.1 Normalization phase

A simple additive weighting (SAW) technique is used to deal with the different magnitudes of QoS properties and obtain a score from diverse QoS dimensions [38, 44]. Positive and negative properties are scaled in different ways, as defined in Eqs. (9) and (10), respectively:

$$\overline{q}(\cdot) = \begin{cases} \dfrac{q(\cdot) - \min(q(\cdot))}{\max(q(\cdot)) - \min(q(\cdot))} & \text{if} \max(q(\cdot)) \neq \min(q(\cdot)) \\ 1 & \text{if} \max(q(\cdot)) = \min(q(\cdot)) \end{cases} \tag{9}$$

$$\overline{q}(\cdot) = \begin{cases} \dfrac{\max(q(\cdot)) - q(\cdot)}{\max(q(\cdot)) - \min(q(\cdot))} & \text{if} \max(q(\cdot)) \neq \min(q(\cdot)) \\ 1 & \text{if} \max(q(\cdot)) = \min(q(\cdot)) \end{cases} \tag{10}$$

Apparently, the time property and cost property are both negative while the reliability property is positive. Based on Eqs. (9) and (10), the QoS scores of time, cost, and reliability property of a MCCS can be calculated, respectively. And the overall QoS score of a MCCS can be expressed as follows:

$$\overline{Q}(\mathrm{MCCS}) = \overline{q}_1(\mathrm{MCCS}) \times w_1 + \overline{q}_2(\mathrm{MCCS}) \times w_2 \\ + \overline{q}_3(\mathrm{MCCS}) \times w_3 \tag{11}$$

4.4.2 Optimization problem formulation

The problem of SESG-SC can be tackled by solving the following model:

$$\mathrm{Max}\left(\overline{Q}(\mathrm{MCCS})\right) \tag{12}$$

Subject to:

$$q_1(\mathrm{MCCS}) \leq q_1(T) \tag{13}$$

$$q_2(\mathrm{MCCS}) \leq q_2(T) \tag{14}$$

$$q_3(\mathrm{MCCS}) \geq q_3(T) \tag{15}$$

$$\sum_{j=1}^{J} |x(i,j)| \geq 1 \tag{16}$$

The constraint in Eqs. (13)–(15) indicates a MCCS should satisfy the QoS requirements of a MFMT. If it fails to fulfill any of the constraints, the service composition is failed. The constraint in Eq. (16) indicates at least one MCS should be combined into the corresponding SESG for each subtask (the OOM-SC approach rigorously restricts $\sum_{j=1}^{J} |x(i,j)| = 1$).

5 Algorithm design for SESG-SC

In this section, we design a matrix-coded genetic algorithm with collaboratively evolutional populations to solve the problem of SESG-SC.

5.1 The basic GA

GA is a search heuristic used to generate useful solutions to optimization and search problems. It mimics the process of natural selection. GA conducts search via a “population,” which contains a collection of “individuals.” Each individual should be encoded by a string of values, called a “chromosome” (each value within a chromosome is called a “gene”), representing a candidate solution to an optimization problem. All individuals have fitness values computing by a fitness function (which is required to be optimized) according to their chromosomes. Three operators of selection, crossover, and mutation are usually provided to enable individuals to evolve toward better solutions generation by generation, ultimately producing the acceptable near-to-optimal solutions.

In view of the irregular and large-scale solution spaces of the SESG-SC problem, we try to design a new GA, which further improves our previously developed hybrid operator-based matrix-coded genetic algorithm [25] by introducing a new feature of collaboratively evolutional populations, aiming at obtaining highly efficient decision for service composition in CMfg. The framework of the MCGA-CEP is shown in Fig. 3.

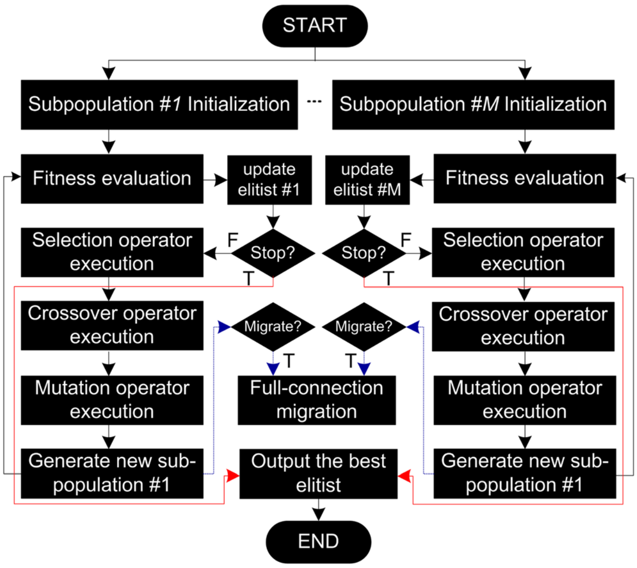


Fig. 3 Framework of the MCGA-CEP

5.2 Chromosomal representation

We encode the chromosome of individuals (solutions) by a matrix

$$X = \begin{bmatrix} x_{11} & \ldots & x_{1j} & \ldots & x_{1J} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \ldots & x_{ij} & \ldots & x_{iJ} \\ \vdots & & \vdots & & \vdots \\ x_{I1} & \ldots & x_{Ij} & \ldots & x_{IJ} \end{bmatrix}, \text{ where any gene } x_{ij} \in \{1,-1,0\} \tag{17}$$

The explanation of the chromosome is as follows:

- Any gene x_{ij} in the chromosomal matrix indicates the decision of whether the elementary service MCS_{ij} is combined into the SESG_i and what kind of redundant structures it joins. If $x_{ij}=1$, it means MCS_{ij} is selected to form the part of the selective structure of SESG_i; if $x_{ij}=-1$, it means MCS_{ij} is selected to form the part of the parallel structure of SESG_i; if $x_{ij}=0$, it means MCS_{ij} is not selected into the SESG_i.
- Any row vector $(x_{i1},\ldots,x_{ij},\ldots,x_{iJ})$ stands for the ith CMCSS_i for the ith subtask ST_i. At least one gene in the vector is nonzero (−1 or 1), according to the constraint in Eq. (16). That means at least one MCS should be combined into the corresponding SESG for each subtask. It should be noted that each CMCSS may have hardly the same number of candidate MCSs in real-world use; in other words, the length J of each row vector is different. In order to simplify the chromosomal representation, we define each row vector of the same length J, because a step of preselection of MCSs from each CMCSS can be designed and performed before service composition. Preselection aims to find out the QoS top-J MCSs in each CMCSS.

5.3 Fitness function

Let X_n denotes any solution (individual) to the problem of SESG-SC. Based on Eqs. (12)–(16), we define the fitness function as follows:

$$\mathrm{Fitness}(X_n) = \left(\overline{Q}(\mathrm{MCCS})\right)_{X_n} \times p(X_n) \tag{18}$$

In the above expression, the term $\overline{Q}(\mathrm{MCCS})$ reflects the objective function (Eq. (12)) and the term $p(X_n)$ reflects a penalty function. Let λ denotes the penalty coefficient, $\lambda\in(0,1]$, then the penalty function is defined as follows:

$$p(X_n) = p_1(X_n) \times p_2(X_n) \times p_3(X_n)$$
$$\text{where}\begin{cases} p_1(X_n) = \begin{cases} 100\%, & \text{if } q_1(\mathrm{MCCS}) \leq q_1(T) \\ \lambda \times q_1(T) \big/ q_1(\mathrm{MCCS}), & \text{otherwise} \end{cases} \\ p_2(X_n) = \begin{cases} 100\%, & \text{if } q_2(\mathrm{MCCS}) \leq q_2(T) \\ \lambda \times q_2(T) \big/ q_2(\mathrm{MCCS}), & \text{otherwise} \end{cases} \\ p_3(X_n) = \begin{cases} 100\%, & \text{if } q_3(\mathrm{MCCS}) \geq q_3(T) \\ \lambda \times q_3(\mathrm{MCCS}) \big/ q_3(T), & \text{otherwise} \end{cases} \end{cases} \tag{19}$$

Equation (19) reflects the constraints in Eqs. (13)–(15). Any individual (solution) that satisfies the constraints in Eqs. (13)–(15) is regarded as a successful service composition, and the value of penalty function $p(X_n)$ is assigned 100 %. Any case that violates the constraints in Eqs. (13)–(15) results in failure of service composition. In the case of failure, we do not simply assign 0 % to $p(X_n)$, instead a percentage reflecting how close it approaches the fulfillment of a constraint, because these individuals may also have useful genes though they violate some constraints.

5.4 Population initialization

According to the framework of MCGA-CEP, $M(M\geq 2)$ subpopulations are generated and evolve in parallel during the execution of the algorithm. Each subpopulation contains N individuals. Each individual is randomly generated gene by gene. For each gene of the chromosome of an individual, a number from the set of {1,−1,0} is randomly selected.

In order to satisfy the constraint in Eq. (16), a row of all zero values is illegal. Then, a *secondary procedure of repair* is design as follows:

Step 1 1

Suppose any row $(x_{i1},\ldots,x_{ij},\ldots,x_{iJ})$ of an individual is initialized (or changed by genetic operators of crossover or mutation) to be a row of all zero value. Then, randomly select a S-length segment $\left(x_{ij_1},\ldots,x_{ij_S}\right)$ from the row $(x_{i1},\ldots,x_{ij},\ldots,x_{iJ})$;

Step 2 Change each gene of the segment $\left(x_{ij_1}, \ldots, x_{ij_S}\right)$ into a randomly selected number from the set of $\{1,-1\}$.

5.5 Operators

5.5.1 Selection operator

A fitness-based selection operator is usually designed to choose the fine individuals to breed a new generation. A roulette wheel method is implemented in the selection operator of MCGA-CEP combined with an elitist model. According to the roulette wheel method, individuals which have the chance to breed are selected based on their respective probability of selection. The selection probability of any individual X_n is defined as follows:

$$\mathrm{prob}(X_n) = \frac{\mathrm{Fitness}(X_n) - \chi \times \mathrm{min\_Fitness}}{\sum_{n=1}^{N} \mathrm{Fitness}(X_n) - \chi \times \mathrm{min\_Fitness}} \tag{20}$$

In Eq. (21), min_Fitness denotes the minimal fitness in current subpopulation and $\chi(0 \leq \chi < 1)$ denotes the pressure coefficient of selection.

According to the elitist model, the individuals of top-N_0 fitness (N_0 will be given as an algorithm parameter) can be regarded as elitist, which are selected to breed at a probability of 100 %.

5.5.2 Crossover operator

A matrix-coded chromosome-based multi-segment crossover is proposed to enable a thorough information exchange between individuals. The designed crossover operator combines the two operators of row-based crossover and point-based crossover proposed by Liu et al. [25] into one integrated operator, which contributes to less complexity in computing and implementation.

We design the multi-segment crossover as follows:

Step 1 For any two individuals X_1 and X_2, generate a crossover template (CT) (CT is a $I \times 2$ matrix). Each row of CT contains two indices, i.e., *start index* and *end index*, indicating the segment that is conducted crossover operation ranges from start index to end index. The value of the indices is randomly selected from the set $\{1, 2, \ldots, J\}$, which are subject to start index < end index.

Step 2 For each row of X_1 and the corresponding row of X_2, exchange the genes that locate in the segment between the start index and the end index, according to the CT. And two new individuals X_1' and X_2' are generated.

Step 3 Check the constraint violation (a row of all zero values is illegal) of X_1' and X_2' in Eq. (16) and fix the possible violation with the secondary procedure of repair provided in Section 5.4.

5.5.3 Mutation operator

Similar to the multi-segment crossover, a multi-segment mutation is designed as follows:

Step 1 Generate a $I \times 2$ matrix as the mutation template for any individual X. Each row of the $I \times 2$ matrix includes two indices of the start index and end index which randomly obtain their value from the natural numbers 1 to J, subject to start index < end index.

Step 2 Use randomly selected $x_{ij}' \in \{1, -1, 0\} - \{x_{ij}\}$ to replace any gene x_{ij} that locates in the segment between the start index and the end index and generate a new individual X'.

Step 3 Check the constraint violation of X' in Eq. (16) and fix the possible violation with the secondary procedure of repair.

5.6 Collaborative evolution of populations

In order to take the advantage of multi-core computing resources, we design a multi-subpopulation-based GA. Accordingly, communication among subpopulations should be considered. We study two factors about this issue:

(1) Connected topology: three kinds of topologies can be used in subpopulation communication, that is, ring, grid, and full-connection. The full-connection topology is thought to be of higher searching capability, because every two subpopulations have a channel to exchange information directly. Moreover, a communication mode constructed with full-connection topology is reported to be quite advantageous as implemented with message passing interface (MPI) [19]. So, we implement the full-connection communication model in the current algorithm.

(2) Migration mechanism: in consideration of the high time consumption of full-connection model, an adaptive migration mechanism is designed to ensure that only the necessary communications are conducted among subpopulations, thereby reducing the communication cost of redundant migration. The number of progression-free generation can be used as an index to estimate when a migration is needed [19]. Progression-free generation means a subpopulation keep its best individual invariable for generations. Let Inv_m denotes the number of

progression-free generation of the mth subpopulation. Inv_m is initialized to be zero. When the next generation obtains the same best individual as the last generation, $\mathrm{Inv}_m = \mathrm{Inv}_m + 1$; otherwise, $\mathrm{Inv}_m = 0$. As the Inv_m of any two subpopulations outnumber than a given upper-bound Inv_Max, a migration should be conducted between these subpopulations. Let c_m denotes the migration coefficient, $0 < c_m \le 1$; then, the expression of Inv_Max can be defined as a function of c_m and the number of individuals in all subpopulations:

$$\mathrm{Inv\_Max} = (M \times N)^{-\exp(c_m)} \quad (21)$$

Equation (21) indicates that the probability of migration increases with the growing number of migration coefficient (because the Inv_Max declines accordingly and thus providing more chances to trigger the execution of migration) and decreases with the expanding of the scale of overall individuals.

6 Experiments

Simulation experiments were performed based on the motivating example of OMP (as described in Section 3), and six subtasks were included in an OMP task. Accordingly, six candidate sets (CMCSSs) of elementary services (MCSs) should be provided to compose a composite service (MCCS) to execute the task.

Two series of experiments were designed to evaluate the proposed SESG-SC approach and the designed MCGA-CEP, respectively. All the experiments were implemented and executed in MATLAB R2011b on a dual-core 2.40 GHz PC with 4 GB RAM under Windows 7.

6.1 Experiments and discussion on SESG-SC approach

6.1.1 Parameter initialization of problem model

$Q(\mathrm{MCS}_{ij})$ is the QoS data of a MCS that were randomly generated. The QoS-time property of MCS values ranges from 10 to 100 h, the QoS-cost values range from 100 to 1000 dollars, and the QoS-reliability values range from 60 to 100 %.

$Q(T)$ is the vector of QoS requirements on a manufacturing task T that was randomly generated based on the QoS data of MCSs. The minimum and maximum values of each QoS property of a MCS were firstly found out in respective CMCSSs. And these values were used to compute the absolute minimum/maximum QoS of the manufacturing task T (by using the aggregation patterns and formulas proposed by Cardoso [45]). Let $\min Q(T)$ and $\max Q(T)$ denote the absolute minimum/maximum QoS of the manufacturing task T, respectively, and then the QoS requirements of T values can range from $\min Q(T)$ to $\max Q(T)$. The computation of $\min Q(T)$ and $\max Q(T)$ is polynomial [44].

W is randomly given in $w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$, indicating that the ranking of a user's QoS preference is time > cost > reliability.

6.1.2 Evaluation on the success rate and overall QoS gained by SESG-SC

The process of this experiment was design as follows:

Step 1 Sixteen MCSs for each CMCSS were initialized. The QoS values of each MCS were generated randomly as described in Section 6.1.1. Eight groups of experiments were carried out with the given MCSs for each CMCSS varying from 2 to 16 (with an increment of 2 in each group), and each given MCS was randomly chosen from the initialized MCSs.

Step 2 One hundred manufacturing tasks were randomly generated (as described in Section 6.1.1) in each group of experiments and requested the CMfg system one by one. In order to reflect the possibly emerged unfeasibility condition (i.e., a task with severe QoS constraints), the first 50 % of these tasks randomly generated their QoS data among the range from $(\min Q(T) + \max Q(T))/2$ to $\max Q(T)$, and the other 50 % of the tasks generated their QoS data among the range from $\min Q(T)$ to $(\min Q(T) + \max Q(T))/2$. It should be noted that $\min Q(T)$ and $\max Q(T)$ were calculated based on the QoS parameters of the total number of 16×6 MCSs (the number 16 indicate 16 MCSs for each CMCSS; the number 6 indicate 6 CMCSSs).

Step 3 For each request of manufacturing task, the approaches of SESG-SC, OOM-SC, and MCET-SC with MCGA-CEP were implemented to generate composite service responses (i.e., MCCS), respectively. The parameters of MCGA-CEP were given as follows: the number of subpopulation M is set to 2 (considering the dual-core processor of the experimental environment). The size of each subpopulation N is set to 200 (as suggested in [19]). The multi-segment crossover probability (CP) is equal to 0.90, the multi-segment mutation probability (MP) is equal to 0.04, the pressure coefficient of selection (χ) is equal to 0.99, and the penalty coefficient (λ) is equal to 0.80 (as suggested in our previous work [25]). In the elitist model, 5 % of individuals ($5\,\% \times 100 = 10$ individuals) with the best fitness in a subpopulation were regarded as an elitist, considering that keeping too many individuals in the elitist model had a negative impact on the individual diversity in evolution of the subpopulation. The migration coefficient c_m is set to 0.3 because too frequent migration could disturb the evolutionary direction of subpopulations and result in unnecessary time consumption; thus, the upper bound of the number of progression-free generation Inv_Max is set to $\sqrt[exp0.3]{200 \times 2} \approx 85$. The total number of evolution cycles (max generation) is set to 1000.

Step 4 The number of successfully responded tasks in each approach was recorded, the success rate was computed, and the fitness of MCCS obtained by different approaches was recorded, computing the accumulative total fitness of 100 tasks, which reflects the corresponding overall QoS of MCCS.

The experimental results are shown in Figs. 4 and 5.

The experimental data in Fig. 4 indicated that

- From the overall QoS point of view, the ranking of these three service composition approaches discussed in this paper from low to high was OOM-SC < MCET-SC < SESG-SC.
- The accumulative total fitness of OOM-SC was gradually improved as the number of MCSs increased, because expansion of the scale of candidate MCSs may bring some MCSs with fine QoS parameters which bettered the overall QoS by substituting for the relatively incompetent MCSs.
- By comparison, both the approaches of MCET-SC and SESG-SC gained a rapid improvement on accumulative total fitness, because both of them employed similar policies of service binding. Concretely speaking, the SESG-SC binds incompetent MCSs together while the MCET-SC binds several MCCSs together, thereby generating solutions that make the best use of available resource and capabilities.
- Moreover, SESG-SC did better than MCET-SC, because MCS-based binding in SESG-SC is a kind of fine granularity combination while MCCS-based binding in MCET-SC is coarsely granular. That means SESG-SC orchestrates the competent MCSs more accurately, while binding a whole MCCS in MCET-SC may import incompetent MCSs which exert a negative effect on the overall QoS.
- Furthermore, the rapid improvement of accumulative total fitness in both SESG-SC and MCET-SC only happened before the number of MCSs reached 12; after that, the trend of improvement in both approaches slowed down. That is because at the early stage, with the increasing number of MCSs, more MCSs are integrated into MCCS (or MCCSs) that contributes to a rapid improvement on the QoS properties of time and reliability. Then, the QoS constraint on cost property (i.e., constraint in Eq. (14)) starts to function, because employing more MCSs results in higher cost, which prohibits the continuous binding of services; accordingly, the further improvement on fitness is attributed to the replacement of some relatively less competent MCSs by those new available MCSs with better QoS properties.

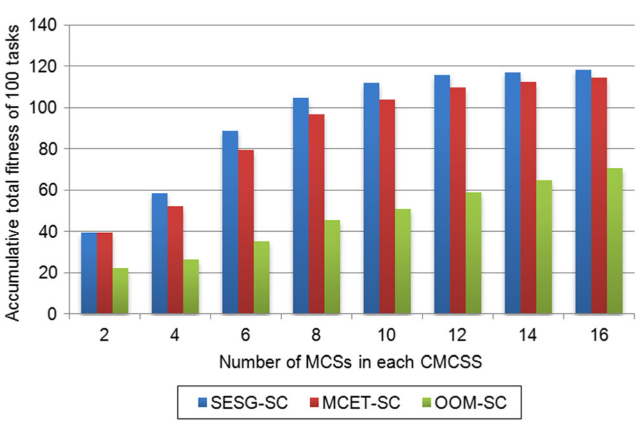


Fig. 4 Fitness comparison among the different approaches

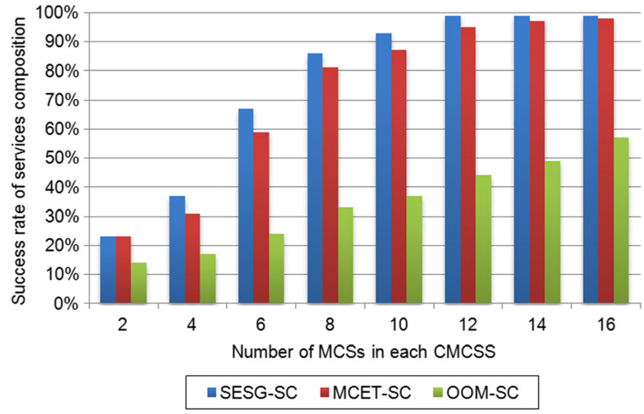


Fig. 5 Success rate comparison among the different approaches

The experimental data in Fig. 5 indicated that

- From the success rate point of view, the ranking of the three approaches from low to high was also OOM-SC < MCET-SC < SESG-SC.
- When the number of MCSs was small (from 2 to 8), none of the approaches can obtain an acceptable success rate of 80 %, because the available manufacturing resource and capabilities were in relatively short supply as compared with the QoS requirements of tasks. As that number increased, the success rate of SESG-SC and MCET-SC approach arose rapidly and achieved the rate of more than 80 % successively or even being very close to 100 %, because both of these approaches use a kind of service binding policy that was verified to be capable of gaining better overall QoS outcomes as discussed above.
- Moreover, SESG-SC was in an advantageous position in comparison with MCET-SC. That is because SESG-SC can achieve a higher overall QoS outcome than MCET-

SC, and thus, SESG-SC handles more cases of unfeasibility conditions of severe QoS constraints.

- Unfortunately, although the number of MCSs kept increasing, OOM-SC can hardly reach the 80 % of the success rate. That is because there are always some tasks with relatively severe QoS constraints that cannot be solved by the OOM-SC model, due to its strict limitation on the relationship between MCSs and subtasks. OOM-SC can get a higher success rate only if there are always some super MCSs with absolutely fine QoS parameters available for each subtask of requesting tasks, which is unpractical in real manufacturing business.

In conclusion, the SESG-SC approach makes a significant progress on the overall QoS outcome and success rate of service composition in CMfg systems.

6.2 Experiments and discussion on MCGA-CEP

We conducted this series of experiments to evaluate the performance of MCGA-CEP. The single-population-based MCGA (MCGA-SP) is used as a reference algorithm [58]. The process of this experiment was design as follows:

Step 1 Five hundred MCSs for each CMCSS was initialized. The QoS values of each MCS were generated randomly as described in Section 6.1.1.

Step 2 Five test cases which varied the number of MCSs from 100 to 500 in each CMCSS was generated. Each MCS in a CMCSS was randomly selected from the corresponding 500 MCSs which were initialized for the CMCSS in step 1. Then, the solution spaces range from 100^6 to 500^6. As suggested in [19], in industrial information systems, many manufacturing tasks contain usually 5 to 10 subtasks with dozens to hundreds of candidate services. Obviously, the scales of the experiments were in accordance with the practical situation in CMfg.

Step 3 Ten to fifty MFMTs for the first to fifth test case was generated, respectively. In any test case, each MFMT requested the CMfg system *one by one*. The fitness for each MFMT was recorded, and the accumulative total fitness was used to evaluate the effect of these two algorithms to search near-to-optimal solutions. The convergence generation and convergence time for each MFMT was recorded, and the average convergence generation and average convergence time was computed, to evaluate the efficiency of the two algorithms.

The experimental results are shown in Figs. 6 and 8. The experimental data suggested that

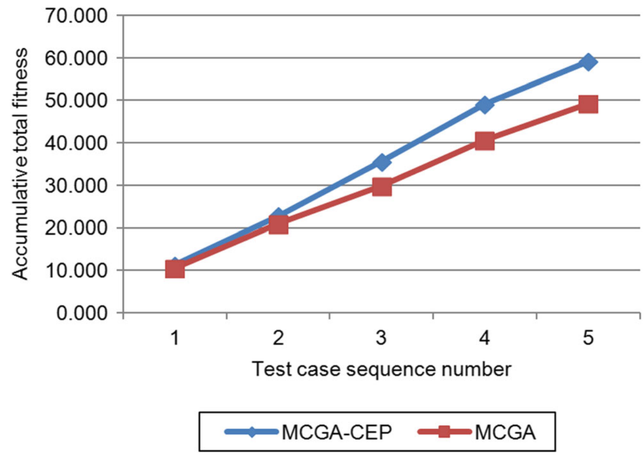


Fig. 6 Fitness comparison between MCGA-CEP and MCGA

- From the angle of the effect, MCGA-CEP showed a significant superiority over the other algorithm, as shown in Fig. 6. The first reason is attributed to the collaboration between the two subpopulations in MCGA-CEP. These subpopulations evolved in parallel and thus provided more chances to keep the diversity of individuals. Consequently, it is more possible for MCGA-CEP to overcome the prematurity in a large-scale search space. And the second reason is that multi-segment-based crossover and mutation operators were designed in MCGA-CEP instead of the traditional point-based or single-segment-based operators in MCGA-SP, which enabled a thorough information exchange between individuals based on their matrix coding manner and thus contributed to systematic search around the current solutions.
- From the angle of efficiency, MCGA-CEP was slightly inferior compared with MCGA-SP, as shown in Fig. 7. Although the migration between the two subpopulations may result in an extra cost of time, the performance gap between MCGA-CEP and MCGA-SP was considerably narrowed as a result of the following two factors: firstly,

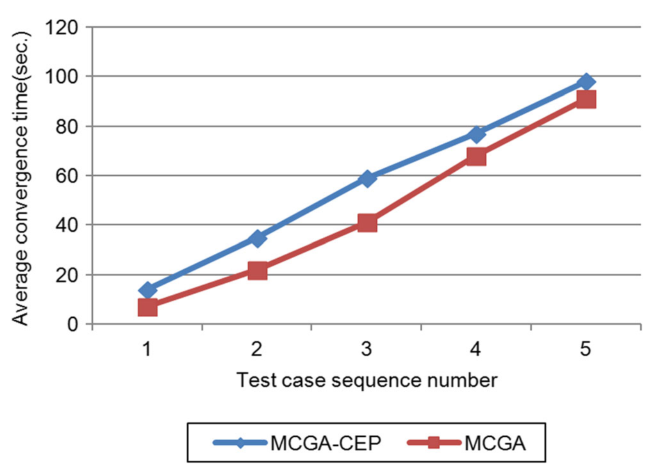


Fig. 7 Average convergence time comparison between MCGA-CEP and MCGA

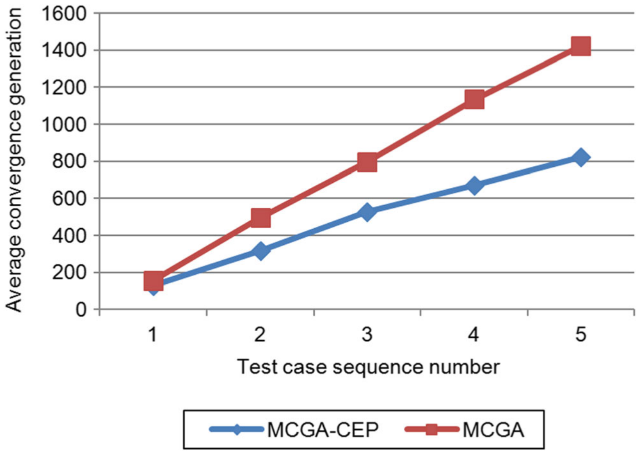


Fig. 8 Average convergence generation comparison between MCGA-CEP and MCGA

the migration mechanism designed in the MCGA-CEP played a vital role in controlling unnecessary communications among subpopulations. The number of progression-free generation reflected the real-time status of population evolvement and enabled only valuable migrations to be conducted, thereby reducing the extra cost on subpopulation communication. Secondly, the multi-segment-based operators were more efficient than the point-based or single-segment-based operators. Being capable of thorough information exchange between individuals, the operators in MCGA-CEP contributed to a great decrease on convergence generations (as shown in Fig. 8).

In a word, the proposed MCGA-CEP achieves a significant improvement of the fitness of solutions at a cost of marginal extra time as opposed to MCGA-SP in solving the problem of SESG-SC.

7 Conclusion and future work

The present study advances the state-of-the-art for QoS-aware service composition in cloud manufacturing based on the following main contributions: (1) an approach of SESG-SC is proposed, which releases the assumption of one-to-one mapping between elementary services and subtasks in traditional service composition methods, and allows a free combination of multiple functionally equivalent elementary services into a SESG to perform each subtask collectively, thereby bettering the overall QoS and achieving more acceptable success rate, especially considering the scenarios of severe QoS constraints. (2) Three kinds of redundant structures within SESGs are discussed, and the corresponding QoS evaluation formulas are also proposed, which make it possible to introduce an optimal construction of SESGs into the optimization model of QoS-aware service composition, thereby contributing to a better outcome of overall QoS and success rate. (3) An algorithm of MCGA-CEP is designed to solve the more complex optimization problem of SESG-SC, and experimental data suggest the designed MCGA-CEP achieves a significant improvement of the fitness of solutions at a marginal extra time cost as opposed to the MCGA-SP.

Our study has opened up a promising avenue to further improve the overall QoS of composite services and the success rate of service composition, which enhances customer satisfaction to a new level in CMfg systems. Also, the current study sets the stage for the further work of investigation and application of the SESG-SC approach to model and solve the more general problem of multi-task-oriented QoS-aware service composition in CMfg systems. Additionally, further improvement of the heuristic algorithm is also included in further work.

Acknowledgments This work is financially supported by the Fundamental Research Funds for the Central Universities (No. XDJK2014C042 and No. SWU113028) in China.

References

1. Li B, Zhang L, Wang S, Tao F, Cao J, Jiang X, Song X, Chai X (2010) Cloud manufacturing: a new service-oriented networked manufacturing model. Comput Integr Manuf Syst 16(1):1–7
2. Zhong RY, Lan S, Xu C, Dai Q, Huang GQ (2016) Visualization of RFID-enabled shopfloor logistics big data in cloud manufacturing. Int J Adv Manuf Technol 84(1):5–16
3. Zhang Y, Xi D, Li R, Sun S (2016) Task-driven manufacturing cloud service proactive discovery and optimal configuration method. Int J Adv Manuf Technol 84(1):29–45
4. Dubey R, Gunasekaran A (2015) Agile manufacturing: framework and its empirical validation. Int J Adv Manuf Technol 76(9):2147–2157
5. Ren L, Cui J, Wei Y, LaiLi Y, Zhang L (2016) Research on the impact of service provider cooperative relationship on cloud manufacturing platform. Int J Adv Manuf Technol. doi:10.1007/s00170-016-8345-6
6. Tao F, Hu YF, Zhou ZD (2008) Study on manufacturing grid & its resource service optimal-selection system. Int J Adv Manuf Technol 37(9):1022–1041
7. Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28(1):75–86
8. Wang C, Bi Z, Xu LD (2014) IoT and cloud computing in automation of assembly modeling systems. IEEE Trans Ind Inf 10(2):1426–1434
9. Tao F, Zhang L, Lu K, Zhao D (2012) Research on manufacturing grid resource service optimal-selection and composition framework. Int J Enterp Inf Syst 6(2):237–264
10. Tao F, Cheng Y, Da Xu L, Zhang L, Li B (2014) CCIoT-CMfg: cloud computing and Internet of things-based cloud manufacturing service system. IEEE T Ind Inform 10(2):1435–1442
11. Tao F, Wang Y, Zuo Y, Yang H, Zhang M (2016) Internet of things in product life-cycle energy management. J Ind Inf Integ. doi:10.1016/j.jii.2016.03.001
12. Xu LD, He W, Li S (2014) Internet of things in industries: a survey. IEEE Trans Ind Inf 10(4):2233–2243

13. Li S, Xu LD, Zhao S (2015) The internet of things: a survey. Inf Syst Front 17(2):243–259
14. Cai H, Xu LD, Xu B, Xie C, Qin S, Jiang L (2014) IoT-based configurable information service platform for product lifecycle management. IEEE Trans Ind Inf 10(2):1558–1567
15. Bi Z, Xu LD, Wang C (2014) Internet of things for enterprise systems of modern manufacturing. IEEE Trans Ind Inf 10(2):1537–1546
16. Liu Z, Wang Y, Cai L, Cheng Q, Zhang H (2016) Design and manufacturing model of customized hydrostatic bearing system based on cloud and big data technology. Int J Adv Manuf Technol 84(1):261–273
17. Chen Y, Chen H, Gorkhali A, Lu Y, Ma Y, Li L (2016) Big data analytics and big data science: a survey. J Manag Analyt 3(1):1–42
18. Chong D, Shi H (2015) Big data analytics: a literature review. J Manag Analyt 2(3):175–201
19. Tao F, LaiLi Y, Xu L, Zhang L (2013) FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. IEEE T Ind Inform 9(4):2023–2033
20. Li B, Zhang L, Ren L, Chai X, Tao F, Luo Y, Wang Y, Yin C, Huang G, Zhao X (2011) Further discussion on cloud manufacturing. Comput Integr Manuf Syst 17(3):449–457
21. Zhang L, Luo Y, Tao F, Li B, Ren L, Zhang X, Guo H, Cheng Y, Hu A, Liu Y (2014) Cloud manufacturing: a new manufacturing paradigm. Int J Enterp Inf Syst 8(2):167–187
22. Tao F, Zhao D, Hu Y, Zhou Z (2008) Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. IEEE T Ind Inform 4(4):315–327
23. Xu LD (2011) Enterprise systems: state-of-the-art and future trends. IEEE Trans Ind Inf 7(4):630–640
24. Viriyasitavat W (2016) Multi-criteria selection for services selection in service workflow. J Ind Inf Integ. doi:10.1016/j.jii.2016.03.003
25. Liu W, Liu B, Sun D, Li Y, Ma G (2013) Study on multi-task oriented services composition and optimisation with the 'multi-composition for each task' pattern in cloud manufacturing systems. Int J Comput Integr Manuf 26(8):786–805
26. He W, Xu LD (2015) A state-of-the-art survey of cloud manufacturing. Int J Comput Integr Manuf 28(3):239–250
27. Tao F, Zuo Y, Xu LD, Zhang L (2014) IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. IEEE Trans Ind Inf 10(2):1547–1557
28. Tao F, Zhang L, Liu Y, Cheng Y, Wang L, Xu X (2015) Manufacturing service management in cloud manufacturing: overview and future research directions. J Manuf Sci E-T Asme 137(4):040912
29. Qu T, Lei SP, Wang ZZ, Nie DX, Chen X, Huang GQ (2016) IoT-based real-time production logistics synchronization system under smart cloud manufacturing. Int J Adv Manuf Technol 84(1):147–164
30. Fan W, Xiao T (2011) Integrated architecture of cloud manufacturing based on federation mode. Comput Integr Manuf Syst 17(3):469–476
31. Ma C, Ren L, Teng D, Wang H, Dai G (2011) Ubiquitous human-computer interaction in cloud manufacturing. Comput Integr Manuf Syst 17(3):504–510
32. Ren L, Zhang L, Zhang Y, Tao F, Luo Y (2011) Resource virtualization in cloud manufacturing. Comput Integr Manuf Syst 17(3):511–518
33. Yin C, Huang B, Liu F, Wen L, Wang Z, Li X, Yang S, Ye D, Liu X (2011) Common key technology system of cloud manufacturing service platform for small and medium enterprises. Comput Integr Manuf Syst 17(3):495–503
34. Zhang L, Luo Y, Tao F, Ren L, Guo H (2010) Key technologies for the construction of manufacturing cloud. Comput Integr Manuf Syst 16(11):2510–2520
35. Yin S, Yin C, Liu F, Li X (2011) Outsourcing resources integration service mode and semantic description in cloud manufacturing environment. Comput Integr Manuf Syst 17(3):525–532
36. Zhang Q, Qi D (2011) Service-oriented collaborative design platform for cloud manufacturing. J South China Univ Technol (Natural Science) 39(12):75–81
37. Ren L, Zhang L, Tao F, Zhao C, Chai X, Zhao X (2015) Cloud manufacturing: from concept to practice. Int J Enterp Inf Syst 9(2): 186–209
38. Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE T Software Eng 33(6):369–384
39. Guo H, Tao F, Zhang L, Su S, Si N (2010) Correlation-aware Web services composition and QoS computation model in virtual enterprise. Int J Adv Manuf Technol 51:817–827
40. Zhang H, Hu Y (2011) A hybrid chaotic quantum evolutionary algorithm for resource combinatorial optimization in manufacturing grid system. Int J Adv Manuf Technol 52:821–831
41. Wang ZJ, Liu ZZ, Zhou XF (2011) An approach for composite Web service selection based on DGQoS. Int J Adv Manuf Technol 56(9–12):1167–1179
42. Huang B, Li C, Tao F (2014) A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system. Int J Enterp Inf Syst 8(4):445–463
43. Cao Y, Wang S, Kang L, Gao Y (2015) A TQCS-based service selection and scheduling strategy in cloud manufacturing. Int J Adv Manuf Technol. doi:10.1007/s00170-015-7350-5
44. Zeng L, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H (2004) QoS-aware middleware for Web services composition. IEEE T Software Eng 30(5):311–327
45. Cardoso J, Sheth A, Miller J, Arnold J, Kochut K (2004) Quality of service for workflows and Web service processes. J Web Semant 1(2004):281–308
46. Zeng J, Sun H, Liu X, Deng T, Huai J (2010) Dynamic evolution mechanism for trustworthy software based on service composition. Int J Softw 21(2):261–276
47. Menasce DA (2002) QoS issues in Web services. IEEE Internet Comput 6(6):72–75
48. Li J, Zhao Y, Sun H, Zheng Z, Ma D (2011) DH4SS: a distributed heuristic for QoS-based service selection. Int J Web Grid Serv 7(4): 388–409
49. Alrifai M, Risse T, Nejdl W (2012) A hybrid approach for efficient Web service composition with end-to-end QoS constraints. ACM T Web 6(2):7:1–7:31
50. Richter J, Baruwal Chhetri M, Kowalczyk R, Bao VQ (2012) Establishing composite SLAs through concurrent QoS negotiation with surplus redistribution. Concurr Comp-Pract E 24(9):938–955
51. 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
52. Lin C, Sheu RK, Chang Y, Yuan S (2011) A relaxable service selection algorithm for QoS-based Web service composition. Inf Softw Technol 53(12):1370–1381
53. He Q, Yan J, Jin H, Yang Y (2014) Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction. IEEE T Software Eng 40(2):192–215
54. Ko JM, Kim CO, Kwon IH (2008) Quality-of-service oriented Web service composition algorithm and planning architecture. J Syst Softw 81(11):2079–2090
55. Li F, Xu LD, Jin C, Wang H (2012) Random assignment method based on genetic algorithms and its application in resource allocation. Expert Syst Appl 39(15):12213–12219
56. Li F, Xu LD, Jin C, Wang H (2011) Intelligent bionic genetic algorithm (IB-GA) and its convergence. Expert Syst Appl 38(7):8804–8811
57. Li F, Xu LD, Jin C, Wang H (2011) Structure of multi-stage composite genetic algorithm (MSC-GA) and its performance. Expert Syst Appl 38(7):8929–8937
58. Ma Y, Zhang C (2008) Quick convergence of genetic algorithm for QoS-driven Web service selection. Comput Netw 52(5):1093–1104