UNIVERSITY OF BIRMINGHAM

University of Birmingham
Research at Birmingham

An improved discrete bees algorithm for correlation-aware service aggregation optimization in cloud manufacturing

Xu, Wenjun; Tian, Sisi; Liu, Quan; Xie, Yongquan ; Zhou, Zude; Pham, Duc

DOI:
10.1007/s00170-015-7738-2

License:
None: All rights reserved

Document Version
Peer reviewed version

Citation for published version (Harvard):
Xu, W, Tian, S, Liu, Q, Xie, Y, Zhou, Z & Pham, D 2016, 'An improved discrete bees algorithm for correlation-aware service aggregation optimization in cloud manufacturing', *The International Journal of Advanced Manufacturing Technology*, vol. 84, pp. 17-28. https://doi.org/10.1007/s00170-015-7738-2

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
The final publication is available at Springer via http://dx.doi.org/10.1007/s00170-015-7738-2

Checked 25/8/2016

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of ‘fair dealing’ under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 10. May. 2024

An Improved Discrete Bees Algorithm for Correlation-Aware Service Aggregation Optimization in Cloud Manufacturing

Wenjun Xu[1,2,*], Sisi Tian[1,2], Quan Liu[1,2], Yongquan Xie[1,2], Zude Zhou[1,2], Duc Truong Pham[3]

[1]School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

[2]Key Lab. of Fiber Sensing Technology and Information Processing, Ministry of Education, Wuhan 430070, China

[3]School of Mechanical Engineering, University of Birmingham, Birmingham, UK

xuwenjun@whut.edu.cn, tiansisi@whut.edu.com, quanliu@whut.edu.cn, xyqwhut0728@126.com, zudezhou@whut.edu.cn, d.t.pham@bham.ac.uk

Abstract: As an effective approach to realize value and efficiency-added manufacturing activities, service aggregation usually plays an important role in Cloud Manufacturing (CMfg), and in order to improve the performance of manufacturing service aggregation, the quality of service (QoS) issue should be considered. However, most existing works related to QoS-based service aggregation assume that the services which to be aggregated are independent from each other, and the service aggregation evaluation models ignore the correlation between the services, which directly leading to inaccuracies in the evaluation of QoS of aggregation services. In this paper, three kinds of correlation are considered and the correlation-aware QoS model of aggregation service is presented in three levels. Furthermore, how to select an appropriate service to compose newly and optimal performance service from massive cloud services is discussed, as the service aggregation optimal-selection problem is one of the key issues in CMfg. An improved discrete bees algorithm based on Pareto (IDBA-Pareto) is proposed to solve the problem in this context for CMfg. The presented method adopts a novel neighborhood searching mechanism underpinned by variable neighborhood searching (VNS) to improve the exploitation ability. The dynamic crowding distance adjustment strategy and the Pareto solution acceptance strategy at a certain probability are utilized to maintain diversity of solutions in population, so as to facilitate escaping from local optimum. The simulation results validate the effectiveness, high-efficiency and superiority of IDBA-Pareto due to better population diversity and convergence speed.

Keywords: Cloud Manufacturing, manufacturing service aggregation, QoS correlation model, intelligent optimization, IDBA-Pareto

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

1 Introduction

In recent years, in view of the coexistence problems of resource bottleneck and resource idle [1, 2], how to achieve resource optimal allocation in manufacturing has become one of developmental target of industrial informatics technology. A number of researches have been carried out with the advancement of networked manufacturing mode, such as integrated manufacturing, grid manufacturing, agile manufacturing, etc. which realizing the sharing and collaboration of manufacturing resources. However, new problems show up with these novel technologies and operation modes, such as big data [3] and cloud technology [4], which drag down its performance in transforming into applications.

Driven by requirements in market and advanced information technology, CMfg [5-7] takes advantage of the characteristics and results of existing manufacturing mode. In the CMfg systems, manufacturing resources or manufacturing capacity circulate in form of services. There exist a large number of cloud services in the CMfg system. The existing service has limitations and cannot accomplish the complex tasks. Hence, it is required to have various services with different ability to form aggregation service with complex ability. Users' demand can be satisfied through reusing and aggregating cloud service to form instant strain loosely-coupled cloud. Manufacturing service aggregation realizes on-demand allocation and value-added manufacturing resources, and it also reduces manufacturing cost. The problems in service aggregation optimization are mainly manifested in service aggregation model. In aspect of service aggregation model, most existing studies about QoS-based service aggregation suggests that there is not any dependencies between services in the process of aggregation, which leading to the degradation of accuracy in evaluating the QoS of aggregation service when establishing a service aggregation model. Therefore, the correlation between services is studied in this paper contains the definition and computation model. In order to deal with service aggregation optimization, a highly-efficient service aggregation optimization algorithm is desired to solve the multi-objective CMfg service aggregation optimal selection (MO-CMSAOS) problem and to get high-quality service aggregation set. A more efficient aggregation solution can achieve better optimal allocation and utilization of resources.

The rest of this paper is organized as follows. Section 2 introduces the related works about service aggregation optimization and correlation among services. The MO-CMSAOS problem is described in section 3. Section 4 establishes the correlation-aware QoS model of aggregation service on three levels, which are the basic QoS model, the QoS evaluation model and the QoS correlation model of aggregation service respectively. The proposed IDBA-Pareto algorithm is detailed presented in section 5. Section 6 shows the experiment results and analyzes the efficiency and effectiveness of IDBA-Pareto algorithm. Conclusions are drawn in section 7.

2 Related Works

Tao et al. [8] analyzed the typical characteristics of CMfg and discussed the whole-lifecycle implementation of service aggregation in CMfg. Several key issues for service aggregation were studied in detail, such as modeling, evaluation and optimal-selection. The authors pointed out some future works and provided theoretical foundation for realizing service aggregation.

In term of researches on service aggregation modeling and evaluation, the existing works based on QoS metrics are primary to build an effective QoS evaluation model so as to achieve optimal selection. Raje et al [9] discussed eighteen QoS metrics of elementary services, which are availability, reliability, error rate, etc. Zeng et al [10] presented a five-dimensional evaluation model based on five generic QoS parameters, including execution price, execution duration, reputation, successful execution rate and availability. A six-dimensional manufacturing service evaluation model was established in [11]. Liu et al [12] presented an evaluation model for

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

service aggregation based on four QoS attributes under the four basic composite modes. However, the existing woks about QoS modeling and evaluation ignored the influence of correlations between services for service aggregation, and a few researchers paid more attention to the researches of correlations among services. Tao et al [8] presented the composition correlation relationship briefly. Three kinds of relations between two elementary services were described including the composable relation, business entity relation and statistical relation. A QoS model was established that supports correlation between services by taking relation degree as an important measure of composite service quality [13]. A correlation-aware QoS description model was built to characterize a candidate service having a correlation with other candidate services [14]. The business entity relation between services was studied in particular [15]. Guo et al [16] described the composable correlation, business entity correlation and statistical cooperate correlation among web services, and established correlation-aware QoS computation model of service composition in virtual enterprise. However, the preceding studies define relationship between services and built QoS models, and these models have their own characteristics in terms of definition and relation degree calculation, which were not comprehensive. They can not be used directly in cloud manufacturing environment. In this context, a more profound study of correlation among services is included and a three-level correlation-aware QoS model of service aggregation is built in the paper.

The optimal selection of service aggregation has attracted more attention in recent years, which is a discrete constrained multi-objective optimization problem. As a way to deal with the disadvantages of transforming a multi-objective problem into a single objective problem, the Pareto-optimal concept [15, 17] has been effective in solving multi-objective optimization problem. In addition, a number of intelligent optimization algorithms combined with Pareto-optimal concept have been adopted to solve such problems, such as genetic algorithm (GA) [17], particle swarm optimization (PSO) [15, 18], ant colony optimization (ACO) [19], etc. Liu et al [12] improved the performance of GA, but the optimal solutions are relatively insufficient in solving large-scale service aggregation problem. GA has slow convergence rate and shows weakness in local searching. PSO is simple for implementation, but it may easily get trapped in local optimality. Many attempts have been tried to seek novel algorithm or hybrid algorithm to improve the performance. Bees algorithm (BA) [20, 21] is an optimization algorithm with the combination of neighborhood searching and random global searching by mimics the food foraging behavior of swarms of honey bees. With the advantages of the superior ability of local and global searching, easy operation and available for hybridization with other algorithms, BA has been found to have the ability to solve a set of multi-objective optimization problems [22, 23]. Compared with the optimization algorithms, e.g. GA, PSO, ACO, etc., the performance of BA has shown a great competitiveness [24]. Some researchers improve the performance of BA in aspects of initialization parameters, neighborhood radius, searching strategy to solve complex optimization problems [25, 26]. An improved discrete bees algorithm is proposed to handle the MO-CMSAOS problem in this context. It introduces a novel neighborhood searching mechanism based on variable neighborhood searching (VNS) to improve the exploitation ability, a dynamic crowding distance adjustment strategy to maintain diversity of solutions in population and a Pareto solution acceptance mechanism at a certain probability to maintain diversity of solutions in population, so as to escape from local optimum.

3 Problem Statement

In CMfg environment, usually a single service is difficult to satisfy the needs of user's request, so it is necessary to combine various services together to accomplish the complex tasks cooperatively. The MO-CMSAOS problem can be described as followed. Initially, the user's task is decomposed into n subtasks

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

symbolized by $(ST_1, ST_2, \ldots, ST_n)$ according to certain logic rules. Each subtask can be accomplished by candidate services $(MS_1, MS_2, \ldots, MS_m)$ meeting the function demand. These services form a candidate services set (CSS). Then one service is selected from corresponding CSS for each subtask. In this way, a service aggregation solution is constructed. Finally, one aggregated service as a solution with optimal QoS performance that is up to users' requirements, such as cost minimization, time minimization, lowest reliability and reputation, is selected. With the growing number of candidate services and subtask available, it is troublesome to find an optimal solution. Fig. 1 shows the optimal selection process of service aggregation. The CMfg service platform matches the requirement of user with historic cases in case database of service aggregation firstly, and service aggregation is adopted to search the optimal path.

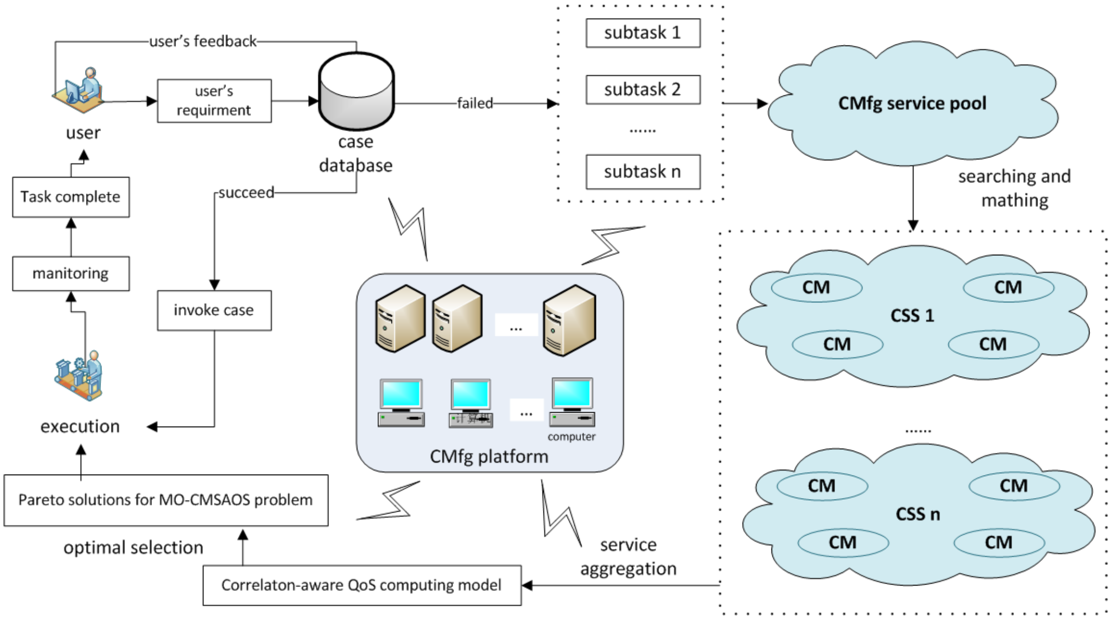

Fig. 1 Optimal selection process of service aggregation

4 Correlation-Aware QoS Model

4.1 Correlation among manufacturing services

Current service evaluation model assumes that the QoS of a candidate service is independent from other candidate services, so that the existing evaluation models cannot deal with the problem in which the QoS of a candidate service has a correlation with other candidate services. Hence, correlation among services is described including definition and computation of correlation degree in this section.

Three kinds of correlation among services are proposed, i.e., interface correlation (InC), business correlation (BuC), statistical correlation (StC). It is assumed there are *n* services in a service aggregation path (SAP) and each service has m candidate services, for any two services (S_i, S_j) in the SAP, the correlation between them is defined as Correlation(S_i, S_j) = <InC, BuC, StC>.

(1) Definition and correlation degree of InC

Provided S_i, is the preorder of S_j, InC is defined that whether there exist data logical correlation between S_i and S_j, i.e., whether there exists a certain similarity in concept, attribute and attribute value between S_i's output (O_i) and S_j's input (I_j), InC can be divided into five level according to the degree of logical correlation:

$$InC(S_i, S_j) = \{< O_i, O_j > | O_i \equiv O_j, O_i \supset O_j, O_i \subset O_j, O_i \cap O_j \neq \{O_i, O_j\} and O_i \cap O_j \neq \varnothing\} \quad (1)$$

Then, interface correlation degree (ICD) is described from three aspects: concept, attribute and attribute

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

values.

(2) Definition and correlation degree of BuC

BuC is defined that whether there exists a certain correlation between service providers in SAP, the correlation may be a cooperation relationship for common interests or a competitive relationship. When BuC cannot be described quantitatively, business correlation degree (BCD) is defined as follow:

$$BCD(S_i, S_j) = \begin{cases} 2 & \text{cooperative correlation} \\ 1 & \text{no correlation} \\ 0.5 & \text{competitive correlation} \end{cases} \quad (2)$$

When BuC is quantitative, BCD is described as in [10], QoS is described as follow:

$$Q = (q^1, q^2, \ldots, q^k) \quad (3)$$

q^i is the ith parameter and described as follow:

$$q^i = \{q(i,0) : default; q(i,1) : correlation(i,1); \ldots; q(i,k_i) : correaltion(i,k_i)\} \quad (4)$$

$correlation(i,k)$ represents the kth correlation. QoS values can be negative or positive, so computing model of q^i is calculated as:

$$q^i = \begin{cases} \max(isExist(i,j) \times q(i,j)) & q^i is \text{ positive} \\ \min(isExist(i,j) \times q(i,j)) & q^i is\ negative \text{ and } \forall isExist(i,j) = 1 \end{cases} \quad (5)$$

where $isExist(i,j) = \begin{cases} 0 & correlation(i,j) = 0 \\ 1 & correlation(i,j) = 1 \end{cases}$, $isExist(i,j)$ describes that whether there is a BuC between services, if there exist BuC between them, the value is 1, otherwise the value is 0.

(3) Definition and correlation degree of StC

StC refers to the services which frequently perform task at the same time. If two services often cooperate together, the reputation and reliability of them are higher. Statistical correlation degree (SCD) is described as always, often, once in a while and never in a qualitative and is described as follow in quantitative:

$$SCD(S_i, S_j) = \gamma / E \quad (6)$$

Where γ and E represent the success rate of S_i and S_j working together the number of times they are invoked in a period in historical data respectively.

4.2 Correlation-aware QoS model of aggregation service

Each component service has a set of service candidates with similar function but differ in QoS, four QoS criterias are considered: execution cost (C), response time (T), availability (Ava) and reliability (Rel).

The correlation-aware QoS model of elementary service is defined as followed:

$$q^i = \{q_0^i : default; q_1^i : SC(InC); q_2^i : SC(BuC); q_3^i : SC(StC)\} \quad (7)$$

where q_0^i, q_1^i, q_2^i and q_3^i represent no relationship, InC, BuC and StC respectively. The corresponding computing model is calculated as followed:

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

$$\begin{cases} q_1^i(S_a) = ICD(S_a,S_b)\times\left(q_0^i\left(S_a\right)+\Delta q_1^i(S_a,S_b)\times\omega^1(S_a,S_b)\right) \\ q_2^i(S_a) = q_0^i\left(S_a\right)+\Delta q_2^i(S_a,S_b)\times\sigma^2(S_a,S_b)\times BCD(S_a,S_b) \\ q_3^i(S_a) = q_0^i\left(S_a\right)+\Delta q_3^i(S_a,S_b)\times\gamma^3(S_a,S_b)\times SCD(S_a,S_b) \end{cases} \tag{8}$$

where $\omega^1(S_a,S_b)=\begin{cases}0 & SC(InC)=0\\1 & SC(InC)=1\end{cases}$, $\sigma^2(S_a,S_b)=\begin{cases}0 & SC(BuC)=0\\1 & SC(BuC)=1\end{cases}$, $\gamma^3(S_a,S_b)=\begin{cases}0 & SC(StC)=0\\1 & SC(StC)=1\end{cases}$, meanning the value is 1 if correlationship exists between them, otherwise the value is 0.

The correlation-aware QoS model of service aggregation under four basic composite modes is shown in Fig. 2 and corresponding computing model is defined in Table 1 [27]. It is assumed that there are *n* service $(MS_1, MS_2, \ldots, MS_n)$ in a SAP, MS are different in parallel structure, *pi* is the probability that a service is chosen among many alternative paths in selection structure, and $\sum_{i=1}^{n} p_i = 1$. *k* is the executions that a service is executed in circulation structure, each QoS parameter can be calculated by Eq. (8)

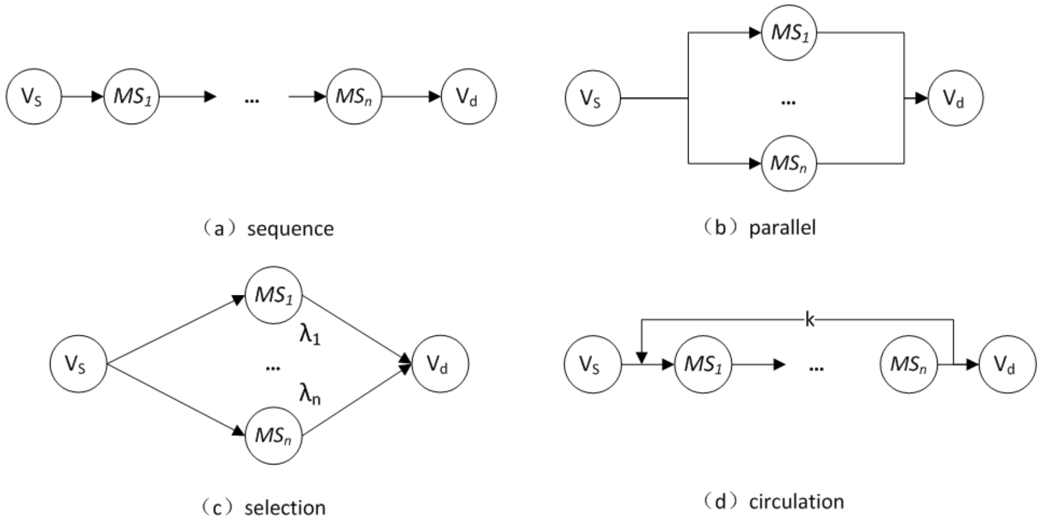

Fig. 2 Four basic composite modes

Table 1 QoS computing model

QoS attribute	Squence	Parallel	selection	circulation
Execution cost(C)	$\sum_{i=1}^{n} C(MS_i)$	$\sum_{i=1}^{n} C(MS_i)$	$\sum_{i=1}^{n} p_i C(MS_i)$	$k\sum_{i=1}^{n} C(MS_i)$
Response time(T)	$\sum_{i=1}^{n} T(MS_i)$	$Max(T(MS_i))$	$\sum_{i=1}^{n} p_i T(MS_i)$	$k\sum_{i=1}^{n} T(MS_i)$
Availability(Ava)	$\prod_{i=1}^{n} Ava(MS_i)$	$\prod_{i=1}^{n} Ava(MS_i)$	$\prod_{i=1}^{n} p_i Ava(MS_i)$	$\prod_{i=1}^{k}\prod_{i=1}^{n} Ava(MS_i)$
Reliability(Rel)	$\prod_{i=1}^{n} \mathrm{Re}l(MS_i)$	$\prod_{i=1}^{n} \mathrm{Re}l(MS_i)$	$\prod_{i=1}^{n} p_i \mathrm{Re}l(MS_i)$	$\prod_{i=1}^{k}\prod_{i=1}^{n} \mathrm{Re}l(MS_i)$

Any SAP can be decomposed by the four basic models, in order to calculate QoS of a SAP, non-sequential structures are transformed to sequence mode at first, the computation model of a SAP is as follow:

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

$$
\begin{aligned}
Q(SAP_i) &= \left\langle T(SAP_i), C(SAP_i), Ava(SAP_i), \mathrm{Re}\,l(SAP_i), \mathrm{Re}\,p(SAP_i) \right\rangle \\
&= f(Q(seq), Q(par), Q(sel), Q(cyc)) \\
&= f_{seq} \left\langle \begin{array}{l} f_{seq}(T^{seq}, C^{seq}, Ava^{seq}, \mathrm{Re}\,l^{seq}, \mathrm{Re}\,p^{seq}), \\ f_{par}(T^{par}, C^{par}, Ava^{par}, \mathrm{Re}\,l^{par}, \mathrm{Re}\,p^{par}), \\ f_{sel}(T^{sel}, C^{sel}, Ava^{sel}, \mathrm{Re}\,l^{sel}, \mathrm{Re}\,p^{sel}), \\ f_{cir}(T^{cir}, C^{cir}, Ava^{cir}, \mathrm{Re}\,l^{cir}, \mathrm{Re}\,p^{cir}) \end{array} \right\rangle
\end{aligned} \tag{9}
$$

5 IDBA-Pareto algorithm for MO-CMSAOS problem

5.1 Formalized description of MO-CMSAOS problem

It is assumed there are n services in a SAP and m candidate services in each CSS. Then it can be inferred that there are $\prod_{i=1}^{n} m$ possible solutions for an MO-CMSAOS problem. The MO-CMSAOS problem is to select the optimal one from all possible solutions under multi-objective and multi-constraints. T and C are considered as two objective functions that conflict each other. Reliability and availability are treated as two constraints. MO-CMSAOS are formalized as follows:

$$
\text{Min} \quad F(SAP) = \{[T(SAP), C(SAP)]\} \tag{10}
$$

$$
\text{s.t.} \quad \begin{cases} \mathrm{Re}\,l_i - \mathrm{Re}\,l_{\min} \geq 0 \\ Ava_i - Ava_{\min} \geq 0 \end{cases} \quad i \in (1, 2, \ldots, 10) \tag{11}
$$

where $T(SAP)$ and $C(SAP)$ are two objective funcitons, $Ava_{\min}$ and $\mathrm{Re}\,l_{\min}$ is lowest of availability and reliability respectively.

Researchers have paid much attention to solve multi-objective optimization problem. On the whole there are two main methods. One is to transform multi-objectives to a single objective problem. The other is based on Pareto-optimal concept. Pareto-based approaches try to optimize all the objectives by balancing conflicted multi-objective simultaneously to get a set of non-dominated solutions which is widely adopted to solve multi-objective optimization problem, while the method transfer multi-objectives to a single one cannot effectively solve multi-objective optimization problem. Pareto-based approach is used to solve MO-CMSAOS problem in this paper.

5.2 IDBA-Pareto algorithm for MO-CMSAOS problem

5.2.1 IDBA-Pareto algorithm description

The proposed improved discrete bees algorithm is based on Pareto-optimal concept and bees algorithm in the paper. In [23], we developed a discrete bees algorithm to realize service optimal selection for resource service aggregation in cloud manufacturing. However, it did not comprehensively consider the service correlations in manufacturing service aggregation, as well as the correlation-aware QoS model. Moreover, the operation mechanism of the algorithm can still be further improved to achieve better performance, and be able to satisfy the requirements of more complex and large scale services aggregation in CMfg. Therefore, based on the previous works in [23], besides the considering of the service correlations in CMfg, some novel mechanisms on neighborhood searching are also developed for such algorithm in this paper, including the neighborhood

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

structure and searching strategy. Three kind of neighborhood structures are designed to generate new individuals by individual disturbance and expand the neighborhood searching space, and VNS is introduced to improve the neighborhood searching performance of bees algorithm, which can escape form local optimization better. Furthermore, a dynamic adjustment mechanism for crowding distance of population diversity preservation strategies is used to achieve better distribution of solutions. The detailed innovations for designing IDBA-Pareto for MO-CMSAOS are followed:

(1) A non-dominated sorting strategy is adopted to get the non-dominated solutions as selected site for neighborhood searching.
(2) A new neighborhood structure is designed and a novel neighborhood searching mechanism based on variable neighborhood searching (VNS) is adopted to improve the exploitation ability.
(3) Dynamic crowding distance adjustment strategy and the Pareto solution acceptance strategy at a certain probability are introduced to maintain diversity of solutions in population and avoid falling into local optimum respectively.

The flow chart of the proposed algorithm is depicted in Fig. 3.

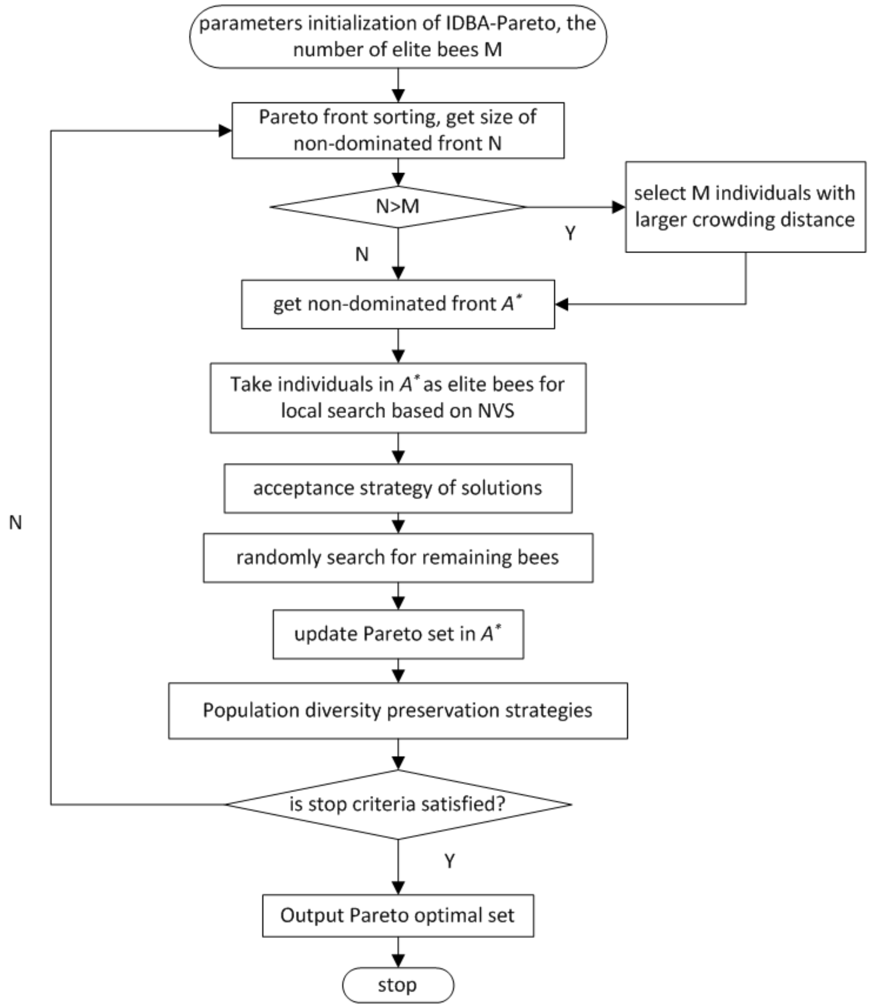

Fig. 3 Flowchart of IDBA-Pareto

5.2.2 Initialization of bees

An individual bee represents a service aggregation path in the algorithm. Integer coding method is applied to encode the candidate services and map the aggregation service into position vectors. It is assumed that an SAP consists of n services. The position vector of a bee is denoted by $X = [x_1, x_2, \dots x_n]$, where x_i represent the ith component node in a sap, and the value of x_i represent the coding of a elementary service selected for the

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

component node. As shown in Fig.4, the corresponding value of position vector is {4, 2, .., 8c.

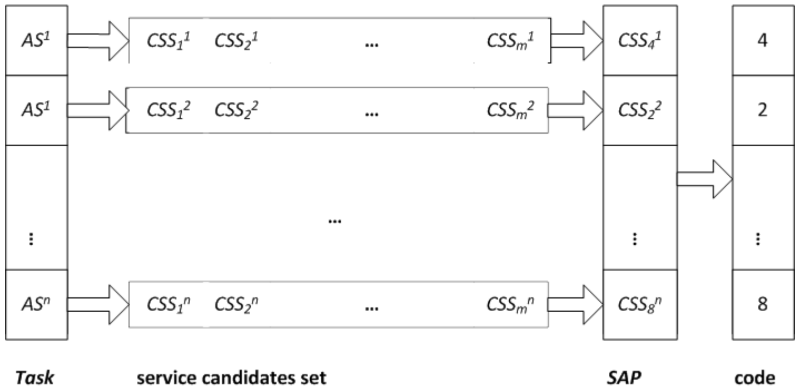

Fig. 4 Initialization of bees

5.2.3 Pareto front sorting

Current population P is a set of candidate solutions. Before neighborhood searching, the non-domination level of individual in P needs to be evaluated. In [17], the pseudo code of Pareto front sorting is shown in Fig. 5 and the solution P is classified to i ($i = 1, 2, \dots$) subpopulations $(F_1, F_2, \dots, F_i)$, F_1 is the Pareto-optimal front in solution P.

Step1.For each individual $p \in P$
  Initialize $S_p = \varnothing$, $n_p = 0$
  For each individual $q \in$ P
    If $p \prec q$ then $S_p = S_p \cup \{q\}$
    else if $p \succ q$ then $n_p = n_p + 1$
  If $n_p = 0$ then $F_1 = F_1 \cup \{p\}$
Step2. Initialize $i = 1$
  While $F_i \neq \varnothing$
    Initialize H = $\varnothing$
    For each individual $p \in F_i$
      For each $q \in S_i$ $n_q = n_q - 1$
        if $n_q = 0$ then H = H $\cup \{q\}$
    $i = i + 1$
    F_i = H

Fig. 5 Pseudo code of sorting of Pareto front

5.2.4 Neighborhood searching

(1) Neighborhood structure

In order to solve MO-CMSAOS problem, a novel neighborhood structure and searching strategy is designed to find better solution. Three neighborhood operators are utilized: swap, inset and inverse. As shown in Fig. 6, swap operator refers to randomly generate two points of individuals to be exchanged and swap genes of the two selected point to form a new individual. Fig. 7 illustrates insert operator. It selects a point and a gene randomly generated is inserted into that point so that a new bee is developed. Inverse operator shown in Fig. 8 randomly generates two mutate point and reverse the genes between the two points.

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

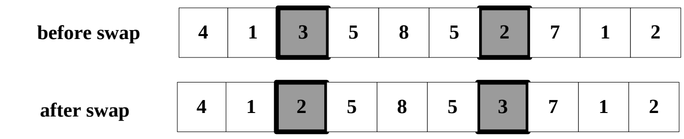

Fig. 6 Swap operator (N_1)

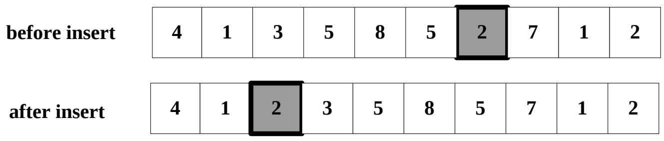

Fig. 7 Insert operator (N_2)

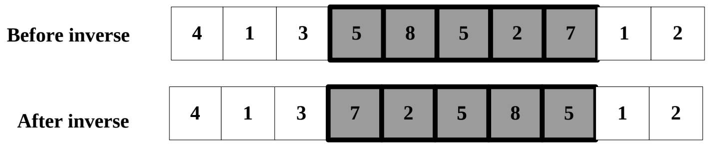

Fig. 8 Inverse operator (N_3)

(2) Neighborhood searching strategy

Variable neighborhood searching (VNS) [28] is a heuristic searching algorithm. It approaches the global optimal solutions by changing neighborhood structure. In doing so it expands the scope of searching to update local optimal solution. VNS is widely used in solving global optimization problems because of its easiness and effectiveness. Neighborhood searching strategy of IDBA-Pareto based on VNS is shown in Fig. 9.

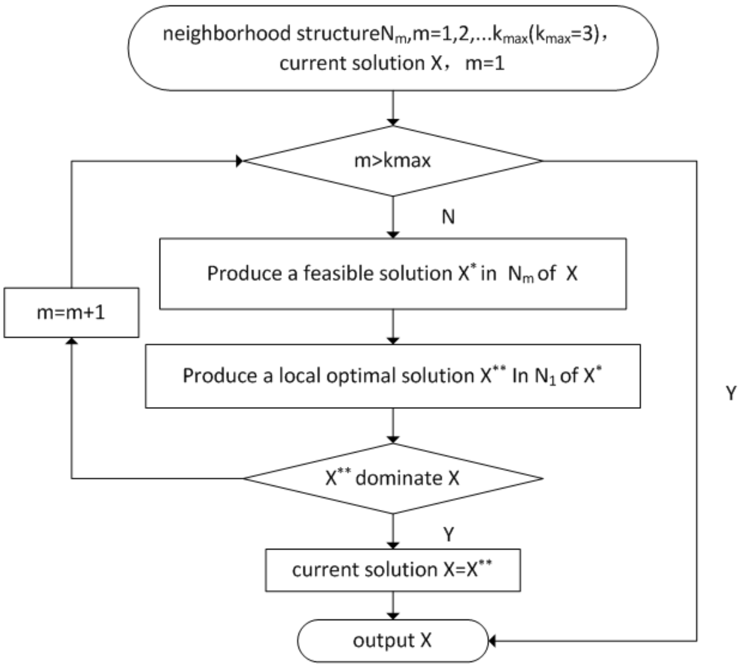

Fig. 9 Neighborhood searching strategy based on VNS

5.2.5 Population diversity preservation strategies

With the algorithm iterates, the local accumulation of individual situations may happen. In this section, a new diversity preservation strategy is proposed to avoid that. The strategy is based on the crowding distance of NSGA-II algorithm and introduces a dynamic adjustment mechanism, and Fig. 10 illustrates the process of the strategy. According to [28], the computation model of crowding distance is as follow:

$$D\left[i\right]_{distance} = \sum_{i=1}^{r}\left(F\left[i+1\right].k - F\left[i-1\right].k\right) \tag{12}$$

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

where F represents objective function of service aggregation problem, r denotes the number of objectives. The greater the crowding distance between individuals is, the better the population diversity is obtained.

Input: Pareto front F_1 with a size of N, M: maximum number of non-dominated solutions

Output: Pareto front F_1 with a size of M

Step1.Initialize the crowding distance of each individual in F_1, $D[i]_{distance} = 0$

Step2.While(N>M)

Step3.calculate the crowding distance of each individual in Eq.(12)

Step4.set the crowding distance of two boundary individual. $D[1]_{distance} = D[M]_{distance} = \infty$

Step5.choose the individual with the smallest distance in F_1 and remove it from F_1

Step6.end while

Fig. 10 Dynamic crowding distance adjustment strategy

The new population diversity preservation strategy removes one individual in each iteration process and adjusts crowding distance of remaining individuals, which avoiding the worse distribution of solutions caused by removing all individuals with small distance at one time.

5.2.6 Acceptance strategy of solution

In order to preserve population diversity and avoid local optima, an acceptance strategy of solution in a certain probability is introduced. Acceptance probability of the solution in the ith iteration is defined as follows:

$$p_i = P_{\max} * e^{-0.02i} \tag{13}$$

where p is the acceptance probability, $P_{\max}$ is the max value of p, and $P_{\max}$ =0.3, i is iterations. The value of p is consistently updated in iterations. A large probability is initialized to accept a large range of solutions to prevent premature convergence and to escape from local optima. Later on, p gradually goes to 0 so that the algorithm convergences quickly. The flow chart of acceptance strategy of solution is shown in Fig. 11.

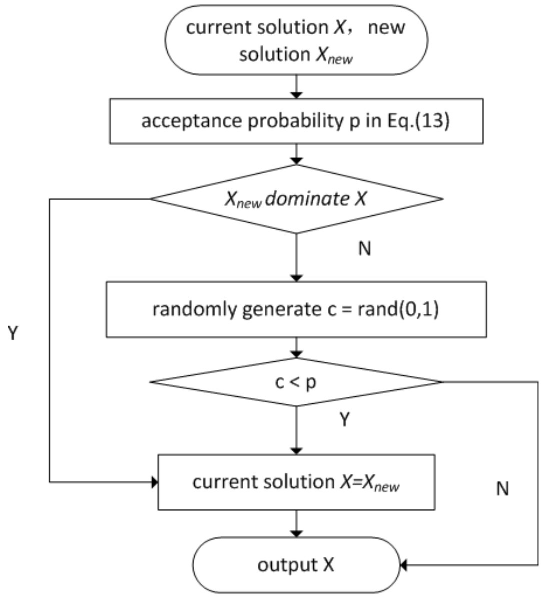

Fig. 11 Flow chart of acceptance strategy of solution

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

6 Case study and performance analysis

6.1 Case study

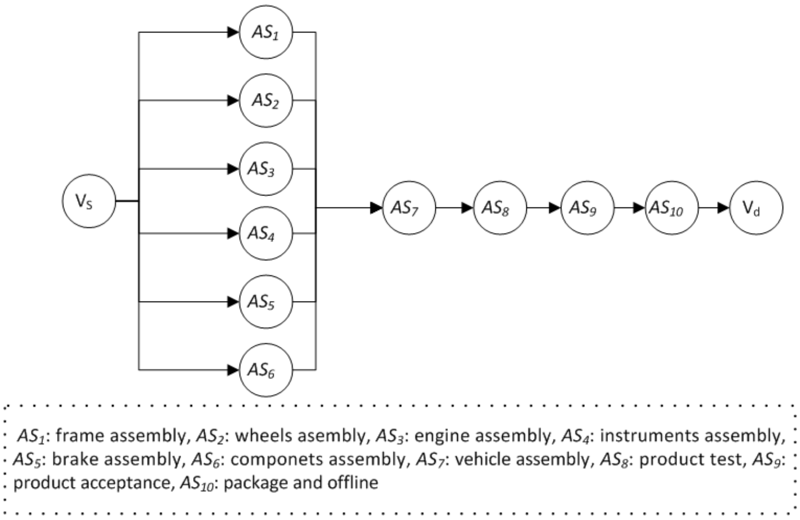

Fig. 12 An application example of service aggregation

In order to verify the proposed algorithm, motorcycle assembly process is considered as an application example of service aggregation optimization problem. Fig. 12 shows the model of motorcycle assembly. It can be seen that six sub-tasks are parallel and the other four sequential. QoS-aware computation model for service aggregation can be described by Eqs. (14) and (15). Time and Cost are considered as two objective functions which conflict each other, reliability and availability are treated as constraints. Where $T\left(SAP\right)$ and $C\left(SAP\right)$ must be smaller than user' requirement $T_{\max}$ and $C_{\max}$ respectively. $Ava_{\min}$ and $\mathrm{Re}\,l_{\min}$ are lower limit of availability and reliability respectively.

$$\text{Minimum} \quad \begin{cases} T\left(SAP\right) = Max\left(T_1, T_2, T_3, T_4, T_5, T_6\right) + T_7 + T_8 + T_9 + T_{10} \\ C\left(SAP\right) = \sum_{i=1}^{10} C_i \end{cases} \tag{14}$$

$$\text{s.t.} \quad \begin{cases} \mathrm{Re}\,l_i - \mathrm{Re}\,l_{\min} \geq 0 \\ Ava_i - Ava_{\min} \geq 0 \\ C_{\max} - C\left(SAP\right) \geq 0 \\ T_{\max} - T\left(SAP\right) \geq 0 \end{cases} \quad i \in \left(1, 2, \ldots, 10\right) \tag{15}$$

The parameters of IDBA-Pareto algorithm are listed in Table 2. Table 3 defines the range of QoS attribute value, the value of each QoS attribute for each candidate service is randomly generated in a range. For simplicity, we assume that each abstract service contains the same number of candidate service, and $C_{\max}$ =500, $T_{\max}$ =30, $Ava_{\min}$ =0.6, $\mathrm{Re}\,l_{\min}$ =0.6. Elite population size is 20. The proposed algorithms are implemented in Matlab environment.

Table 2 Parameters settings for IDBA-Pareto algorithm

Parameter	Symbol	Value

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

Number of population or scout bees	*Pop*	{50,100,150,200}
Number of elite patches	M_{elite}	20
Number of iterations	*Itr*	{100,200,300,400}
Number of service candidates	*SG_Size*	{5,10,20}
Number of subtask	*SG_Num*	10
Maximum of acceptance probability	$P_{\max}$	0.3

Table 3 The range of the QoS attribute value for each candidate service

OoS attribute	Range
Execution cost	[1,100]
Response time	[1,10]
Availability	[0.5,1]
Reliability	[0.5,1]

6.2 Performance analysis

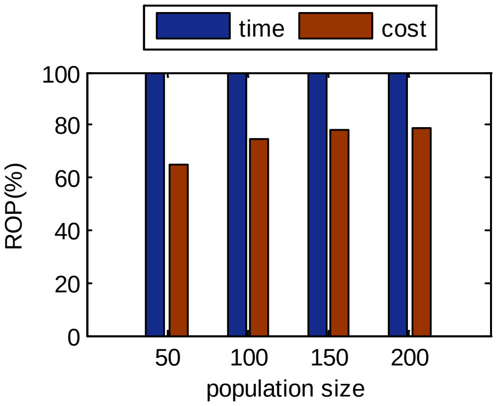

Fig. 13 ROP of IDBA-Pareto

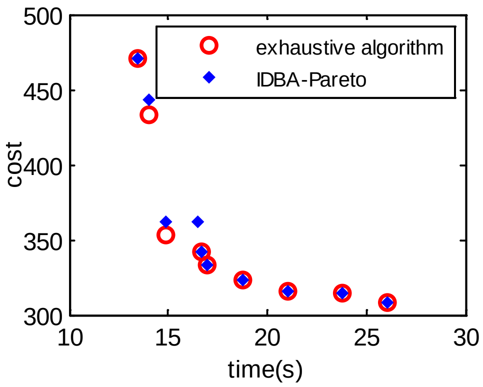

Fig. 14 Comparison of optimal front

We compare exhaustive algorithm with IDBA-Pareto in the rate of optimal result (ROP). When the number of service is 5, IDBA-Pareto algorithm is executed 10 times with the maximum iteration is set to 50, 100, 150 and 200. The average ROP is showed in Fig. 13 and the comparison result is showed in Fig. 14. The ROP is about 74.5 percent when the maximum iteration is 100, which shows the effectiveness of the proposed algorithm in handling MOCMSAOS problem.

The running time when candidate services and populations are in different scale is compared. When the population size is 100, the number of service candidates is 5, 10 and 20 respectively. In each situation the algorithm is executed 10 times with the maximum iterations of 100, 200, 300 and 400. The result is showed in Fig. 15. The number of service candidates is set to 20 and the maximum iteration 400, the average running time is about 7.79s. The results shown in Fig. 16 are obtained on conditions that the number of service candidates is 20, the size of population 50, 100, 150, 200. In each situation the algorithm is repeated 10 times and terminated after 100, 200, 300, 400 iterations. The results display that when the number of population is 200 and iteration limitation is 400, the average running time is about 13.54s. Therefore the results validate the efficiency of IDBA-Pareto in solving MO-CMSAOS problem.

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

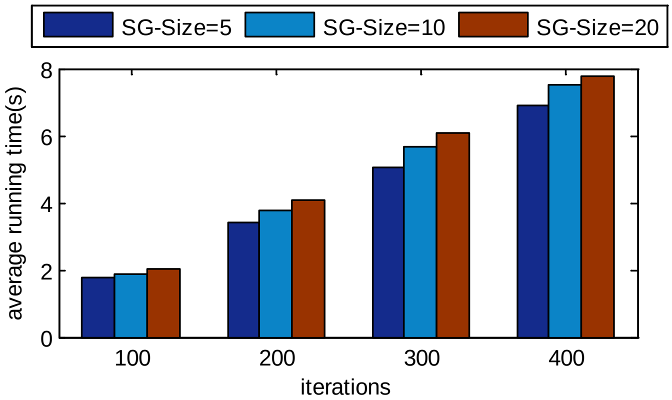

Fig. 15 Average running time in different candidate sizes

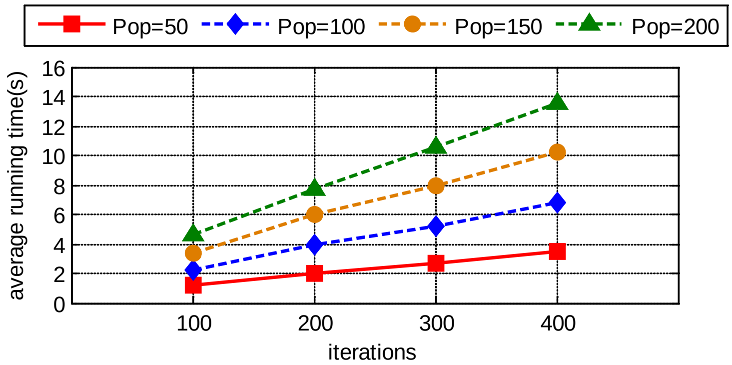

Fig. 16 Average running time in different population sizes

Population diversity preservation strategy proposed in subsection 5.2.5 is compared with basic crowding distance mechanism in NSGA-II [17]. The number of population, service candidates and iterations are set to 100, 20 and 200 respectively. Fig. 17 shows that dynamic crowding distance adjustment strategy can get larger crowding distance between individuals. Therefore the proposed strategy can keep population diversity better.

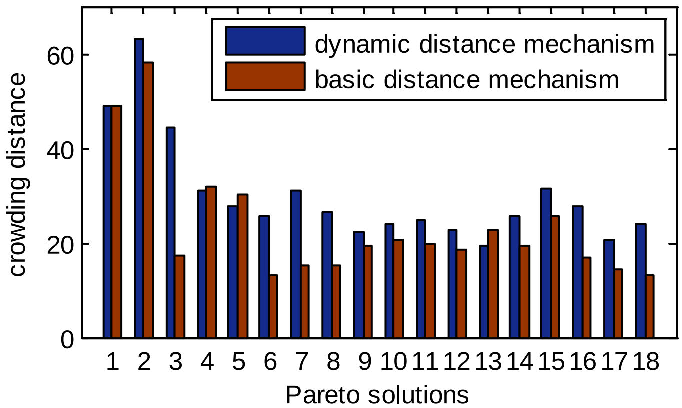

Fig. 17 Comparison of crowding distance

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

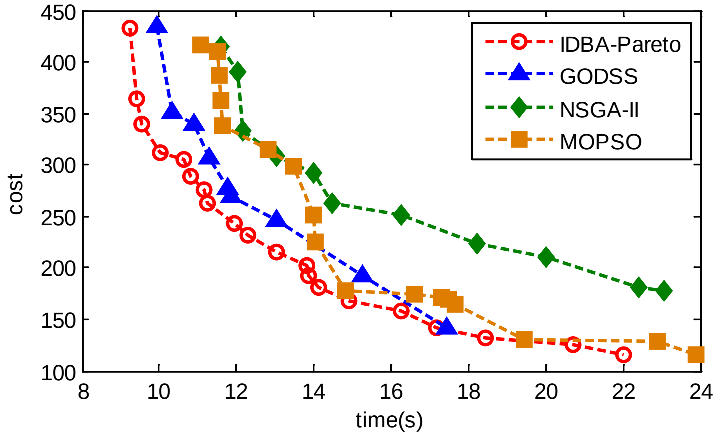

Fig. 18 Comparison of Pareto front

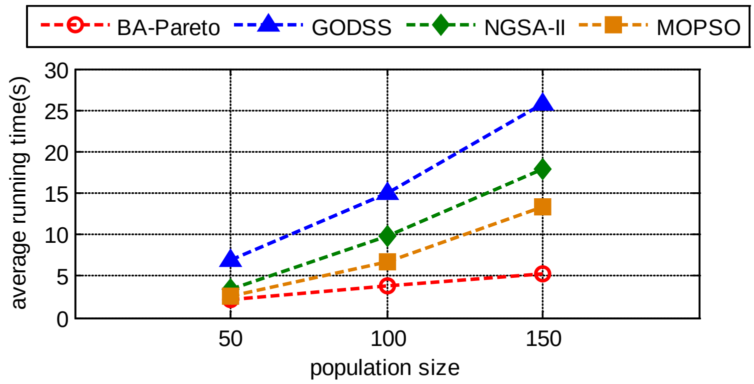

Fig. 19 Comparison of average running time

The proposed algorithm is compared with NSGA-II, MOPSO [18] and GODSS [12] to stress the improved ability of maintaining population diversity. The number of population, candidate services and iterations are 50, 20 and 200 respectively. The results show that IDBA-Pareto can find more effective Pareto optimal solutions, and obtain better solutions as shown in Fig. 18. The crossover and mutation probability of NSGA-II is 0.9 and 0.1 respectively. The crossover probability of GODSS is 1 and mutation probability is 0.15. The value of learning factors of MOPSO is set to 2, while the maximal particle velocity is 0.25 times of the range of values for each variable. According to [29], the inertia weight shrinks linearly when the algorithm proceeds. The maximal inertia weight is 0.9, while the minimal inertia weight is 0.4. Fig. 19 shows IDBA-Pareto has faster convergence speed. In summary, the performance of IDBA-Pareto is superior to NSGA-II, GODSS and MOPSO.

7 Conclusion

The correlation-aware QoS calculation model is built for manufacturing service aggregation in CMfg in this paper. An improved discrete bees algorithm is proposed based on Pareto-optimal concept. An effective neighborhood structure is designed for the algorithm in order to enhance its performance. Population diversity preservation strategy and acceptance strategy of solution are introduced to strengthen the algorithm’s capability of escaping form local optima. A set of experiments have been carried out to verify the efficiency of the proposed IDBA-Pareto. The simulation results demonstrate the effectiveness, high-efficiency and superiority of

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

IDBA-Pareto. Moreover, IDBA-Pareto is able to perform with excellent population diversity and gain competitive convergence speed. The future work would include the dynamic optimization of the parameters of IDBA-Pareto in terms of the dynamics in production processes in CMfg.

Acknowledgement

This research is supported by National Natural Science Foundation of China (Grant No. 51305319), the International Science & Technology Cooperation Program of China (Grant No. 2015DFA70340), and the Key Project of Natural Science Foundation of Hubei Province of China (Grant No. 2013CFA044).

References

1. S. Yin, C. Yin, F. Liu, X. Li. Outsourcing resources integration service mode and semantic description in cloud manufacturing environment. Computer Integrated Manufacturing Systems. 2011. **17**(3): p. 525-532.
2. W.J. Xu, Z.D. Zhou, D.T. Pham, Q. Liu, C. Ji, W. Meng. Quality of service in manufacturing networks: a service framework and its implementation. International Journal of Advanced Manufacturing Technology. 2012. **63**(9-12): p. 1227-1237.
3. Li, Jingran, Fei Tao, Ying Cheng, and Liangjin Zhao. "Big Data in product lifecycle management." The International Journal of Advanced Manufacturing Technology (2015): 1-18.
4. Yang Cao, Shilong Wang, Ling Kang, Changsong Li, and Liang Guo. "Study on machining service modes and resource selection strategies in cloud manufacturing." The International Journal of Advanced Manufacturing Technology (2015): 1-17.
5. B. Li, L. Zhang, S. Wang, F. Tao, J. Cao, X. Jiang, X. Song, X. Chai. Cloud manufacturing: A new service-oriented networked manufacturing model. Computer Integrated Manufacturing Systems. 2010. **16**(1): p. 1-7+16.
6. X. Xu. From cloud computing to cloud manufacturing. Robotics and Computer-Integrated Manufacturing. 2012. **28**(1): p. 75-86.
7. L. Zhang, Y.L. Luo, F. Tao, B.H. Li, L. Ren, X.S. Zhang, H. Guo, Y. Cheng, A.R. Hu, Y.K. Liu. Cloud manufacturing: a new manufacturing paradigm. Enterprise Information Systems. 2014. **8**(2): p. 167-187.
8. F. Tao, L. Zhang, H. Guo, Y.-L. Luo, L. Ren. Typical characteristics of cloud manufacturing and several key issues of cloud service composition. Computer Integrated Manufacturing Systems. 2011. **17**(3): p. 477-486.
9. R.R. Raje, B.R. Bryant, A.M. Olson, M. Auguston, C. Burt. A quality-of-service-based framework for creating distributed heterogeneous software components. Concurrency Computation Practice and Experience. 2002. **14**(12): p. 1009-1034.
10. L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, H. Chang. QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering. 2004. **30**(5): p. 311-327.
11. F. Tao, Y. Hu, D. Zhao, Z. Zhou, H. Zhang, Z. Lei. Study on manufacturing grid resource service QoS modeling and evaluation. The International Journal of Advanced Manufacturing Technology. 2009. **41**(9-10): p. 1034-1042.
12. S. Liu, Y. Liu, F. Zhang, G. Tang, N. Jing. A dynamic web services selection algorithm with QoS global optimal in web service composition. Journal of Software. 2007. **18**(3): p. 646-656.
13. Y. Dai, L. Yang, B. Zhang, Y. Gao. QoS for composite web services and optimizing. Chinese Journal of Computers. 2006. **29**(7): p. 1167-1178.
14. S.-Y. Ye, J. Wei, L. Li, T. Huang. Service-correlation aware service selection for composite service.

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn

Jisuanji Xuebao/Chinese Journal of Computers. 2008. **31**(8): p. 1383-1397.

15. F. Tao, D. Zhao, H. Yefa, Z. Zhou. Correlation-aware resource service composition and optimal-selection in manufacturing grid. European Journal of Operational Research. 2010. **201**(1): p. 129-143.
16. H. Guo, F. Tao, L. Zhang, S. Su, N. Si. Correlation-aware web services composition and QoS computation model in virtual enterprise. International Journal of Advanced Manufacturing Technology. 2010. **51**(5-8): p. 817-827.
17. K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in Parallel Problem Solving from Nature PPSN VI. 2000, Springer Berlin Heidelberg. p. 849-858.
18. C.A. Coello Coello, M.S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. in 2002 Congress on Evolutionary Computation, CEC 2002, May 12, 2002 - May 17, 2002. 2002. Honolulu, HI, United states: IEEE Computer Society.
19. W. Zhang, C.K. Chang, T. Feng, H.-Y. Jiang. QoS-based dynamic web service composition with ant colony optimization. in 34th Annual IEEE International Computer Software and Applications Conference (COMPSAC 2010). 2010. Seoul, Korea, Republic of: IEEE Computer Society.
20. D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi. The bees algorithm - a novel tool for complex optimisation problems. in the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006). 2006.
21. D.T. Pham, M. Castellani. The Bees Algorithm: modelling foraging behaviour to solve continuous optimization problems. Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science. 2009. **223**(12): p. 2919-2938.
22. D.T. Pham, A. Ghanbarzadeh. Multi-objective optimisation using the bees algorithm. in the 3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007). 2007.
23. S. Tian, Q. Liu, W. Xu, J. Yan. A discrete hybrid bees algorithm for service aggregation optimal selection in cloud manufacturing. in 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2013). 2013. Hefei, China: Springer Verlag.
24. B. Yuce, M. Packianather, E. Mastrocinque, D. Pham, A. Lambiase. Honey bees inspired optimization method: The bees algorithm. Insects. 2013. **4**(4): p. 646-662.
25. Q.T. Pham, D.T. Pham, M. Castellani. A modified bees algorithm and a statistics-based method for tuning its parameters. Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering. 2012. **226**(I3): p. 287-301.
26. W. Hussein, S. Sahran, S. Sheikh Abdullah, A new initialization algorithm for bees algorithm, in Soft Computing Applications and Intelligent Systems. 2013, Springer Berlin Heidelberg. p. 39-52.
27. W.-N. Liu, B. Liu, D.-H. Sun. A conceptual framework for dynamic manufacturing resource service composition and optimization in service-oriented networked manufacturing. in 2011 International Conference on Cloud and Service Computing (CSC 2011). 2011. Hong Kong, China: IEEE Computer Society.
28. P. Hansen, N. Mladenović, J. Moreno Pérez. Variable neighbourhood search: methods and applications. 4OR. 2008. **6**(4): p. 319-360.
29. Y. Shi, R. Eberhart. A modified particle swarm optimizer. in the 1998 IEEE International Conference on Evolutionary Computation Proceedings. 1998.

*Corresponding author: Wenjun Xu, xuwenjun@whut.edu.cn