



Research
Intelligent Manufacturing—Article

Dual-Dimensional Manufacturing Service Collaboration Optimization Toward Industrial Internet Platforms



Shibao Pang^{a,b,c}, Shunsheng Guo^{a,b}, Xi Vincent Wang^{c,*}, Lei Wang^{a,b}, Lihui Wang^c

^aSchool of Mechanical and Electrical Engineering, Wuhan University of Technology, Wuhan 430070, China

^bHubei Digital Manufacturing Key Laboratory, Wuhan University of Technology, Wuhan 430070, China

^cDepartment of Production Engineering, KTH Royal Institute of Technology, Stockholm 10044, Sweden

ARTICLE INFO

Article history:

Received 23 February 2022

Revised 15 June 2022

Accepted 13 July 2022

Available online 17 November 2022

Keywords:

Manufacturing service collaboration

Service optimal selection

Service granularity

Industrial Internet platform

ABSTRACT

An Industrial Internet platform is acknowledged to be a requisite promoter for smart manufacturing, enabling physical manufacturing resources to be virtualized and permitting resources to collaborate in the form of services. As a central function of the platform, manufacturing service collaboration optimization is dedicated to establishing high-quality service collaboration solutions for manufacturing tasks. Such optimization is inseparable from the functional and amount requirements of a task, which must be satisfied when orchestrating services. However, existing manufacturing service collaboration optimization methods mainly focus on horizontal collaboration among services for functional demands and rarely consider vertical collaboration to cover the needed amounts. To address this gap, this paper proposes a dual-dimensional service collaboration methodology that combines functional and amount collaboration. First, a multi-granularity manufacturing service modeling method is presented to describe services. On this basis, a dual-dimensional manufacturing service collaboration optimization (DMSCO) model is formulated. In the vertical dimension, multiple functionally equivalent services form a service cluster to fulfill a subtask; in the horizontal dimension, complementary service clusters collaborate for the entire task. Service selection and amount distribution to the selected services are critical issues in the model. To solve the problem, a multi-objective memetic algorithm with multiple local search operators is tailored. The algorithm embeds a competition mechanism to dynamically adjust the selection probabilities of the local search operators. The experimental results demonstrate the superiority of the algorithm in terms of convergence, solution quality, and comprehensive metrics, in comparison with commonly used algorithms.

© 2022 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As Industry 4.0 progresses, modern manufacturing is embracing a conversion from a traditional operation mode to an intelligent mode [1]. Within this trend, various advanced manufacturing technologies and paradigms are emerging [2–4]. As crucial enablers of smart manufacturing, Industrial Internet platforms aim to achieve service-oriented and on-demand production [5,6]. Industrial Internet platforms, which are an extension and implementation of the Internet of Things [7,8] and cloud computing [9,10] into the industrial field, are drawing great interest from both academia and industry [11–13]. With the support of an Industrial Internet

platform, siloed manufacturing resources can be connected and encapsulated as manufacturing services delivered to customers to carry out manufacturing tasks, thereby empowering the unified management and utilization of discrete resources [14]. In general, the fulfillment of a manufacturing task necessitates the collaboration of multiple services [15]. After receiving a task, the platform decomposes it into several subtasks and matches each with a set of candidate services. A *manufacturing service collaboration solution* refers to a portfolio of services selected from candidate sets to fulfill a task. Consequently, one of the main purposes of an Industrial Internet platform is to orchestrate services in order to generate high-quality solutions—that is, to perform manufacturing service collaboration optimization.

During service collaboration optimization, both the functional requirements proposed by the customer and the required

* Corresponding author.

E-mail address: wangxi@kth.se (X.V. Wang).

processing amount of the task must be satisfied. For example, it is common for a task submitted to an Industrial Internet platform to have more than one product to be processed. To enhance the performance of the collaboration solutions, it is rational and practical to select multiple services to fulfill a subtask in parallel. In this way, services are fully utilized, while solution efficiencies significantly increase. Within this context, two dimensions of collaboration coexist: In the vertical dimension, a cluster of services with similar functions collaborate in parallel for a subtask; meanwhile in the horizontal dimension, multiple service clusters with complementary functions collaborate in sequence for the entire task, forming a complex collaboration network. Service collaboration optimization for such a network comprises more than just the optimal selection of services. Amount distribution—that is, how to reasonably distribute the processing amount of one subtask to the multiple selected services—is indispensable. A task may have thousands of alternative service combinations; moreover, within a combination, there may be even more potential schemes for amount distribution among services, resulting in numerous collaboration solutions. To obtain high-quality solutions, neither service selection nor amount distribution can be disregarded.

In addition, to achieve collaboration in two dimensions, an appropriate manufacturing service modeling method is essential. In existing service collaboration optimization, the service invocation time and cost are assumed to be fixed, regardless of the processing amount [16,17]. This assumption cannot be employed due to a violation of the on-demand principle when amount distribution is involved. Moreover, since a subtask is fulfilled collaboratively by multiple services, the amount distribution of the subtask largely hinges on the availability of these services. From the perspective of the entire Industrial Internet platform, manufacturing services exist at intra-enterprise and inter-enterprise levels, resulting in different granularities [18]. For example, a manufacturing service may correspond to a single resource such as a machine tool, a group of resources such as a workstation on a shop floor, or even a complex production process involving multiple resources in different subordinate companies of a conglomerate. The availability of services is limited by their component states and relationships, which changes over time. Unless this factor is taken into consideration during service collaboration optimization, it is difficult to safeguard the feasibility and efficiency of collaboration solutions.

With the aim of generating high-quality collaboration solutions, this paper proposes a novel dual-dimensional manufacturing service collaboration optimization (DMSCO) methodology. The proposed methodology considers the multi-granularity characteristic of manufacturing services and comprises both functional collaboration in the horizontal dimension and amount collaboration in the vertical dimension. The main contributions of this paper are as follows:

(1) By analyzing the components and structures of manufacturing services, a multi-granularity service modeling method is presented, which enables the introduction of complicated services (i.e., composite services and service chains) to service collaboration optimization.

(2) A mathematical model for DMSCO integrating service selection and amount distribution is formulated under the consideration of service granularities, with the objective of optimizing the total cost, reliability, and finishing time of service collaboration solutions.

(3) A multi-objective memetic algorithm is developed to efficiently search for collaboration solutions to the model, in which a competition mechanism is embedded to adjust the selection probabilities of local search operators according to their contributions.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. Section 3 models manufacturing services

and formulates the DMSCO model. In Section 4, the proposed algorithm is depicted. Section 5 reports the numerical experiments, and the conclusions are presented in Section 6.

2. Related work

2.1. Manufacturing service modeling

As the prerequisites of manufacturing service collaboration, physical resource virtualization and manufacturing service modeling have been intensively studied in the past decades. Researchers initially focused on functional and non-functional information descriptions. Shi et al. [19] proposed a manufacturing resource hierarchy model, in which resource information abstracted from the physical resource layer was encapsulated in a resource expressing layer using extensible markup language, and operations to resources were outlined in the resource interface layer in order to shield the internal information communication details of the resources. Vichare et al. [20] established a unified manufacturing resource model for a computer numerical control (CNC) machining system composed of the machine resource and additional auxiliary devices. The model was a standard information representation able to define the elements of the CNC machining system and support the automation of process planning decision-making. Ameri and McArthur [21] extended the manufacturing service description language, an ontology used for modeling service manufacturing capabilities, by means of property inference rules and classification rules, thereby enhancing the ontology semantically and enabling advanced reasoning. Wang and Wang [22] transformed the capabilities of physical resources into manufacturing services utilizing a function block mechanism, thus providing a viable integration method capable of coordinating various manufacturing resources in an interoperable and flexible way.

In fact, manufacturing services occur within a multi-granularity environment [18] and, based on research progress, the construction of multi-granularity manufacturing services has attracted the attention of researchers. The coexistence of services with different granularities empowers the efficient and flexible generation of collaboration solutions. A coarse-grained service can be invoked as a whole, quickly covering most requirements of a task [23,24]. Meanwhile, fine-grained services can flexibly meet other personalized needs. Liu et al. [25] presented multi-granularity resource virtualization strategies consisting of resource aggregation functions and resource clustering algorithms to encapsulate physical resources into cloud services, which bridged the gap between complex manufacturing tasks and underlying resources. Yu et al. [26] designed a multi-level aggregate service planning method based on data mining techniques, thereby determining the number of manufacturing services maintained at different granularities. Zhang et al. [27] proposed a new application pattern—namely, cloud manufacturing for industry alliances—for which a domain-driven development method for multi-granularity manufacturing services was designed, including the two phases of atomic and coarse-grained service development, in order to achieve the on-demand provision of manufacturing resources. Li et al. [28] presented a resource service chain composition evolutionary algorithm based on the dependency between resource services in order to construct optimal coarse-grained services, thereby enhancing the reusability and selection efficiency of the manufacturing services.

2.2. Manufacturing service collaboration optimization

Existing research efforts on manufacturing service collaboration optimization mainly focus on optimal service selection and

configuration. Abundant models have been proposed thus far. Lu and Xu [29] investigated knowledge-based service composition and adaptive resource planning in a cloud manufacturing environment, facilitating fast resource allocation for a given service request through a semantic web-based system. Ren et al. [30] proposed a service selection model considering social relationships among the manufacturing services to maximize the overall synergy effect. Five types of social relationships among services were abstracted from the service interaction and cooperation data, and a service synergy effect model was derived based on a weighted aggregation of relationship strength. Considering logistics services in service collaboration, Zhou et al. [31] analyzed and established a collaborative optimization model of logistics and processing services and took the average task delivery time as the optimization objective. Wu et al. [32] presented a multi-objective integer bi-level multi-follower programming model to concurrently optimize the sustainability (economic, environmental, and social performance) and quality of collaboration solutions. Wang et al. [33] established a multi-objective dynamic service composition reconfiguration model in which eight crucial practical constraints were introduced according to a real-life cloud manufacturing process, thereby narrowing the gap between theory and application.

Due to the non-deterministic polynomial-time hard (NP-hard) nature of optimal service selection [34], intelligent evolutionary algorithms are widely utilized to solve the problem, allowing high-quality solutions to the models to be obtained with fewer computing resources [35]. Bouzary and Chen [36] developed a grey wolf optimizer combined with evolutionary operators of a genetic algorithm; they adapted the continuous structure of the grey wolf optimizer to the optimal service selection problem and avoided local optimal stagnation during the hunting process by providing more exploration strength. Akbaripour and Houshmand [37] were the first to analyze the solution space landscape of the problem, demonstrating that the landscape was rugged and that local optima were gathered in a small area of the search space. Based on their analysis, they designed a hybrid approach integrating an imperialist competitive algorithm with a local search algorithm. Zhang et al. [38] proposed a novel genetic-based hyper-heuristic algorithm in which the genetic algorithm acted as a high-level heuristic to produce appropriate combinations of low-level heuristics operating on the solution domain in order to solve the multi-task-oriented optimal service selection problem in uncertain environments. Zhou and Yao [39] proposed a multi-objective hybrid artificial bee colony algorithm to optimize the quality of service and energy consumption, in which a cuckoo search with a Lévy flight was introduced to maintain the diversity of solutions, and a comprehensive learning strategy was designed to ensure a balance between the exploitation and exploration of the algorithm. Zhang et al. [40] presented an improved non-dominated sorting genetic algorithm-II (NSGA-II) coupled with a Tabu search and enhanced K -means mechanism to optimize the long-/short-term utilities of the providers, consumers, and operators involved in the service collaboration.

To date, the exploration of dual-dimension service collaboration remains limited. Some of the relevant research only focuses on service selection [41,42]. However, distributing the processing amount to the selected services is integral to manufacturing service collaboration optimization when multiple services fulfill the same subtask in parallel. Zhu et al. [43] investigated the amount distribution but ignored the service selection phase; they distributed the amount to all the candidate services. Despite decreasing the finishing time, execution of the collaboration solution becomes difficult due to the massively involved services. Furthermore, the availability of a manufacturing service is simply assumed to be a numerical value in previous references. Although such a solution may be suitable for the field of computer science, the sit-

uation in the manufacturing industry is more complicated. The availability of a manufacturing service is affected by many factors, such as the service structure, internal component services, and scheduled occupancy. Therefore, a single numerical value cannot accurately reflect the availability of a manufacturing service.

2.3. Overview

As mentioned above, both manufacturing service modeling and collaboration optimization have been extensively studied. However, research on service collaboration optimization rarely considers the characteristics of the service granularity, resulting in an incoherence between the service modeling and collaboration optimization. Despite the few efforts that have been made in this area [44,45], a detailed description of the availability of services at different granularities is still missing. Moreover, existing research on service collaboration optimization primarily focuses on the selection of services to satisfy functional requirements, whereas amount-related collaboration is disregarded. Further research on the optimization model and the specific algorithm for dual-dimensional service collaboration is required. To this end, this paper presents a DMSCO methodology that is multi-granularity service-based and that simultaneously optimizes service selection and amount distribution.

3. Problem formulation

3.1. Multi-granularity manufacturing service modeling

In the overall performance of collaboration solutions, the total cost, reliability, and finishing time are of primary concern with regard to customers, respectively corresponding to the unit cost, reliability, and processing speed of services. In addition, the feasibility of solutions is limited by the availability of services. While modeling manufacturing services, this paper primarily focuses on these attributes. It should be noted that the availability of services is measured from the perspective of time—that is, whether the services are available during certain time periods. Manufacturing services exist at both the intra-enterprise level and the inter-enterprise level. According to their internal components and structures, services can be modeled at three granularities: resource services, composite services, and service chains. Notations used for modeling these services are listed in Table 1, and the service expressions and attributes are shown in Table 2.

A *resource service* is a simple service mapped by a single physical resource within an enterprise. Since the resource cannot be available all the time, the availability of the service is expressed

Table 1
Notations used for multi-granularity manufacturing service modeling.

Notation	Description
S	Manufacturing service; referring to a resource service, a composite service, or a service chain
m, n	Indexes of component services in a composite service or a service chain, where $m, n = 1, 2, \dots, N$
k	Index of available durations of a resource service or a composite service, where $k = 1, 2, \dots, K$
c, r, p, a	Unit cost, reliability, processing speed, and availability of S , respectively
d_k	k th available duration of a resource service or composite service
V	Set of component services in a composite service or service chain
E	Set of execution constraints in a service chain
S_m, S_n	m th, n th component services in a composite service or service chain, respectively
c_n, r_n, p_n, a_n	Unit cost, reliability, processing speed, and availability of S_n , respectively

Table 2
Modeling of multi-granularity manufacturing services.

Service Granularity	Expression	Unit cost	Reliability	Processing speed	Availability
Resource service	$S = \{c, r, p, a\}$	c	r	p	$a = \langle d_1, d_2, \dots, d_k, \dots, d_k \rangle$
Composite service	$S = \{c, r, p, a, V\}$	$c = \sum_{S_n \in V} c_n$	$r = \left[\prod_{S_n \in V} r_n \right]^{\frac{1}{n}}$	$p = p_1$	$a = \bigcap_{S_n \in V} a_n$
Service chain	$S = \{c, r, p, a, V, E\}$	$c = \sum_{S_n \in V} c_n$	$r = \left[\prod_{S_n \in V} r_n \right]^{\frac{1}{n}}$	$p = \{p_1, p_2, \dots, p_n, \dots, p_N\}$	$a = \{a_1, a_2, \dots, a_n, \dots, a_N\}$

as a series of available periods of time, where the element d_k indicates the duration of time during which the resource is accessible.

A *composite service* comprises multiple interdependent resource services, which are generally provided by a single enterprise for a specific function. A set of component services $V = \{S_1, S_2, \dots, S_n, \dots, S_N\}$ is invoked at the same time when providing the function to the outside, where a central service S_1 acts as the main implementer of the function, and the rest are auxiliary services collaborating around S_1 . The availability of a composite service is the intersection of the availability of its components.

A *service chain* is a fixed service provided by multiple correlated enterprises, such as conglomerates or alliances. It consists of a set of component (i.e., resource or composite) services $V = \{S_1, S_2, \dots, S_n, \dots, S_N\}$ that collaborate with specific execution constraints $E = \{ \langle S_m, S_n \rangle | S_m \in V, S_n \in V \}$. $\langle S_m, S_n \rangle$ means that S_m is invoked before S_n , and a component service will not be occupied until invoked. The processing speed and availability of a service chain are expressed as the sets of corresponding attributes of the components, respectively.

3.2. The model for DMSCO

In the model for DMSCO, a manufacturing task T contains I subtasks with different function requirements, and each subtask ST_i can be fulfilled by a set of candidate services CSS_i . Regarding the vertical dimension, a cluster of functionally equivalent services $CS_i = \{S_{i1}, S_{i2}, \dots, S_{ij}, \dots, S_{ij}\}$ are selected from CSS_i to fulfill ST_i . Aside

from service selection, the distribution of the processing amount of ST_i to the services in CS_i needs to be determined. In the horizontal dimension, all functionally complementary CS_i compose the final collaboration solution $CS = \{CS_1, CS_2, \dots, CS_i, \dots, CS_I\}$ for T . The following text formulates the model from the two dimensions, where the availability constraints of the services at different granularities are constructed, and the total cost, reliability, and finishing time are adopted as the optimization objectives. Notations used for DMSCO are listed in Table 3.

3.2.1. Collaboration in the vertical dimension

Considering the amount distribution, the total cost C_{ij} , reliability R_{ij} , and finishing time F_{ij} of a single S_{ij} are formulated first. The formulas are different depending on the service granularities. For a resource service or composite service:

$$C_{ij} = Amt_{ij} \cdot c_{ij} \tag{1}$$

$$R_{ij} = r_{ij} \tag{2}$$

$$F_{ij} = B_{ij} + Amt_{ij}/p_{ij} \tag{3}$$

$$\Phi_{ijk} = \begin{cases} 1, & \text{if } [B_{ij}, F_{ij}] \subset d_{ijk} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\sum_{d_{ijk} \in a_{ij}} \Phi_{ijk} = 1 \tag{5}$$

Table 3
Notations used for the DMSCO model.

Notation	Description
i	Index of subtasks, where $i = 1, 2, \dots, I$
j	Index of manufacturing services in CS_i , where $j = 1, 2, \dots, J_i$
T	Manufacturing task
Amt	Total required amount of T
CS	Manufacturing service collaboration solution for T
C, R, F	Total cost, reliability, and finishing time, respectively, of CS to fulfill T
ST_i	i th subtask of T
Amt_i	Total required amount of ST_i
CSS_i	Set of candidate manufacturing services for ST_i
CS_i	Cluster of manufacturing services selected from CSS_i to fulfill ST_i
C_i, R_i, B_i, F_i	Total cost, reliability, beginning time, and finishing time, respectively, of CS_i to fulfill ST_i
S_{ij}	j th manufacturing service in CS_i (S_{ij} is a specific S , which is used in the context in which T is involved)
Amt_{ij}	Processing amount distributed to S_{ij}
$c_{ij}, r_{ij}, p_{ij}, a_{ij}$	Unit cost, reliability, processing speed, and availability of S_{ij} , respectively
$C_{ij}, R_{ij}, B_{ij}, F_{ij}$	Total cost, reliability, beginning time, and finishing time, respectively, of S_{ij} to process Amt_{ij}
d_{ijk}	k th available duration of a resource service or composite service S_{ij}
Φ_{ijk}	Binary variable equal to 1 if Amt_{ij} is processed in the duration d_{ijk} of S_{ij}
V_{ij}	Set of component services in a service chain S_{ij}
E_{ij}	Set of execution constraints in a service chain S_{ij}
S_{ijn}	n th component service in a service chain S_{ij}
p_{ijn}, a_{ijn}	Processing speed and availability of S_{ijn} , respectively
b_{ijn}, f_{ijn}	Beginning time and finishing time, respectively, of S_{ijn} to process Amt_{ij}
d_{ijnk}	k th available duration of S_{ijn}
Φ_{ijnk}	Binary variable equal to 1 if Amt_{ij} is processed in the duration d_{ijnk} of S_{ijn}
$C^{max}, R^{max}, F^{max}$	Maximum total cost, reliability, and finishing time, respectively, of CS
$C^{min}, R^{min}, F^{min}$	Minimum total cost, reliability, and finishing time, respectively, of CS

where Eq. (5) indicates that Amt_{ij} must be processed within an available duration of a_{ij} , ensuring the availability of S_{ij} .

For a service chain, C_{ij} and R_{ij} are calculated in the same way. B_{ij} and F_{ij} are decided by the component services, and are formulated as follows:

$$B_{ij} = \min_{S_{ijn} \in V_{ij}} (b_{ijn}) \tag{6}$$

$$F_{ij} = \max_{S_{ijn} \in V_{ij}} (f_{ijn}) \tag{7}$$

$$f_{ijn} = b_{ijn} + Amt_{ij}/p_{ijn} \tag{8}$$

$$\Phi_{ijnk} = \begin{cases} 1, & \text{if } [b_{ijn}, f_{ijn}] \subset d_{ijnk} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$$\prod_{S_{ijn} \in V_{ij}} \sum_{d_{ijnk} \in a_{ijn}} \Phi_{ijnk} = 1 \tag{10}$$

$$f_{ijn} \leq b_{ijn} \quad \forall \langle S_m, S_n \rangle \in E_{ij} \tag{11}$$

where Eqs. (10) and (11) ensure the availability and correct invocation sequence of the component services, respectively.

For a service cluster CS_i , the total cost C_i , reliability R_i , beginning time B_i , and finishing time F_i are formulated as follows:

$$C_i = \sum_{S_{ij} \in CS_i} C_{ij} \tag{12}$$

$$R_i = \sum_{S_{ij} \in CS_i} \frac{Amt_{ij}}{Amt_i} \cdot R_{ij} \tag{13}$$

$$B_i = \min_{S_{ij} \in CS_i} (B_{ij}) \tag{14}$$

$$F_i = \max_{S_{ij} \in CS_i} (F_{ij}) \tag{15}$$

$$Amt = Amt_i = \sum_{S_{ij} \in CS_i} Amt_{ij} \tag{16}$$

$$CS_i \subset CSS_i \tag{17}$$

where Eq. (16) ensures that the sum of the processing amount distributed to the services in CS_i is equal to Amt_i , and Eq. (17) ensures that CS_i is selected from the candidate service set CSS_i .

3.2.2. Collaboration in the horizontal dimension

Regarding the horizontal dimension, the total cost C , reliability R , and finishing time F of a service collaboration solution CS are formulated as follows:

$$C = \sum_{CS_i \in CS} C_i \tag{18}$$

$$R = \left[\prod_{CS_i \in CS} R_i \right]^{\dagger} \tag{19}$$

$$F = F_i \tag{20}$$

$$B_i \geq F_{i-1} \geq 0 \tag{21}$$

Due to the different measurement methods and units of C , R , and F , objective functions are normalized as follows to optimize the metrics at the same time:

$$\text{minimize } f_1 = \begin{cases} \frac{C - C^{\min}}{C^{\max} - C^{\min}}, & \text{if } C^{\max} \neq C^{\min} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

$$\text{minimize } f_2 = \begin{cases} \frac{R^{\max} - R}{R^{\max} - R^{\min}}, & \text{if } R^{\max} \neq R^{\min} \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

$$\text{minimize } f_3 = \begin{cases} \frac{F - F^{\min}}{F^{\max} - F^{\min}}, & \text{if } F^{\max} \neq F^{\min} \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

4. The proposed memetic algorithm for DMSCO

Service selection has been proved to be an NP-hard problem [34]. Consequently, DMSCO incorporating both service selection and amount distribution is NP-hard. To address this problem, a competition-based multi-objective memetic algorithm (CMOMA) is developed.

4.1. Framework of CMOMA

Memetic algorithms represent a category of metaheuristics that combine evolutionary algorithms with local searches [46]. The framework of the proposed CMOMA is shown in Fig. 1, where a grey wolf optimizer is deployed for the global search, and a competition-based variable neighborhood search is embedded for the local search. For the adaption of DMSCO, a mixed-crossover operator is designed for the global search. In addition, to fully explore the search space, the population is updated through fast non-dominated sorting, and parents (leaders) for a solution are randomly selected from the population.

The local search is of great significance in memetic algorithms and should be tailored to the specific problem. This paper develops two types of local search operators corresponding to service selection and amount distribution, each of which includes four operators. When applying the local search to a solution, one of the operators is selected for execution. In addition, a competition mechanism is designed to update the selection probabilities of the operators.

4.2. Encoding and decoding

To accommodate DMSCO, a two-vector encoding scheme is proposed, which involves a service selection vector \mathbf{X} and an amount distribution vector \mathbf{Y} . Each vector consists of I segments corresponding to I subtasks. Since a subtask ST_i is collaboratively fulfilled by J_i services, the segment \mathbf{Seg}_i in both \mathbf{X} and \mathbf{Y} comprises J_i elements carrying information about the services selected for ST_i . The element x_{ij} of \mathbf{X} is an integer in $[1, L_i]$, representing the index of a selected service in CSS_i (L_i is the length of CSS_i). \mathbf{Y} is used to distribute the amount to the selected services. The element y_{ij} is a real number in $[0, 1]$, representing the weight of the amount distributed to the allelic service on \mathbf{X} . Fig. 2 shows an example of the encoding scheme, in which the task contains three subtasks, each subtask has 20 candidate services, and up to three of these can be selected. Based on the encoding scheme, the population of CMOMA is randomly initialized.

Prior to decoding \mathbf{X} , elements of \mathbf{Y} are inspected. If y_{ij} is equal to 0, the allelic service on \mathbf{X} will not participate in the collaboration. Meanwhile, to prevent the amount distributed to a service from being too small, y_{ij} will be set to 0 when less than 0.1. If all y_{ij} belonging to \mathbf{Seg}_i equal 0, one of them will be set to a random number between 0.1 and 1.0. After inspection, the x_{ij} th candidate service in CSS_i is selected if the allelic y_{ij} is greater than 0. When decoding \mathbf{Y} , the last y_{ij} greater than 0 in each \mathbf{Seg}_i is marked as y_i^* , and the amount distributed to services is then calculated as shown in Eq. (25). Supposing that the total amount of the task in the example in Fig. 2 is 10 000, the decoding result is presented in Fig. 3.

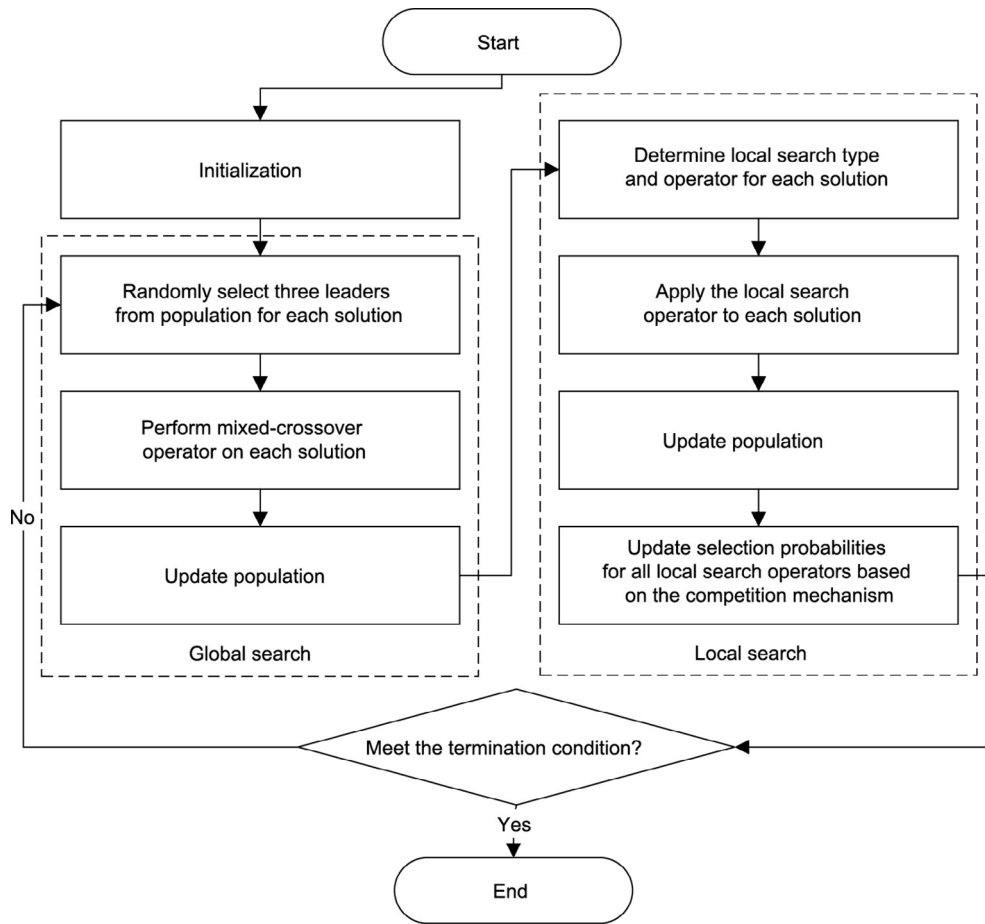


Fig. 1. Framework of CMOMA.

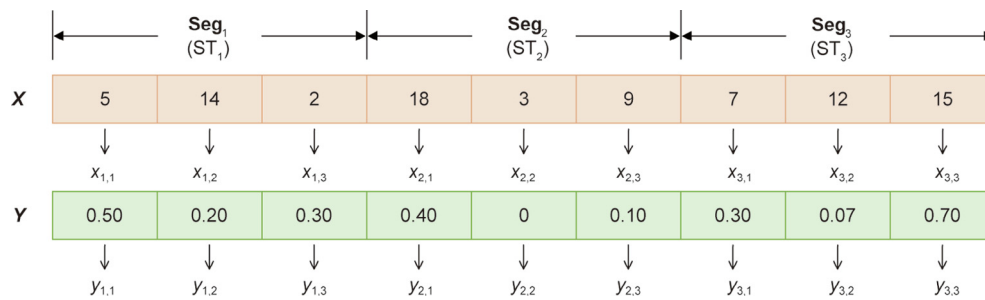


Fig. 2. Example of the encoding scheme.

$$\text{Amt}_{ij} = \begin{cases} \left[\text{Amt}_i \cdot \frac{y_{ij}}{\sum_{y_{ij} \in \text{Seg}_i} y_{ij}} \right], & y_{ij} \neq y_i^* \\ \text{Amt}_i - \sum_{y_{ij} \in \text{Seg}_i} \text{Amt}_{ij}, & y_{ij} = y_i^* \\ y_{ij} \neq y_i^* \end{cases} \quad (25)$$

4.3. Global search

The global search algorithm is extended from the grey wolf optimizer, which is an emerging evolutionary algorithm inspired by the hunting behaviors of grey wolves [47]. In each iteration of the grey wolf optimizer, solutions renew their positions under

the guidance of three leaders (the first three best solutions), α , β , and δ . Compared with similar algorithms, the optimizer is characterized by a fast search speed and easy implementation. It was initially designed for solving continuous optimization problems. As there are both discrete variables \mathbf{X} and continuous variables \mathbf{Y} in the solutions, a mixed-crossover operator is designed in the proposed algorithm. For \mathbf{X} , the crossover operator is formulated as Eqs. (26) and (27), where $\text{rand}()$ is a function returning a random number in $[0, 1]$, and x_{ij}^{new} , x_{ij} , x_{ij}^α , x_{ij}^β , and x_{ij}^δ are \mathbf{X} elements of the new solution, the original solution, α , β , and δ . Fig. 4 presents an example of the crossover operator for \mathbf{X} .

$$r = \text{rand}() \quad (26)$$

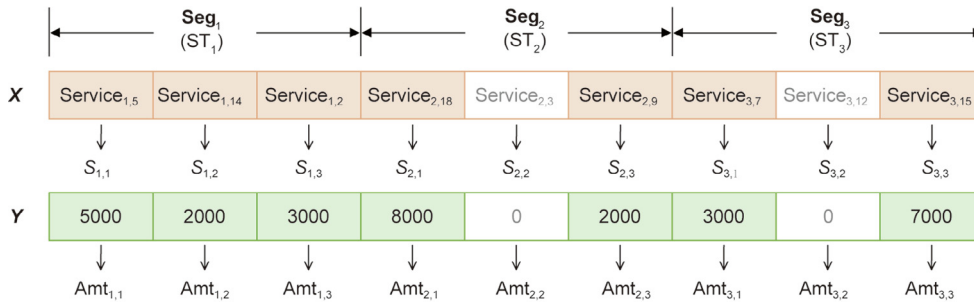


Fig. 3. Example of the decoding scheme.

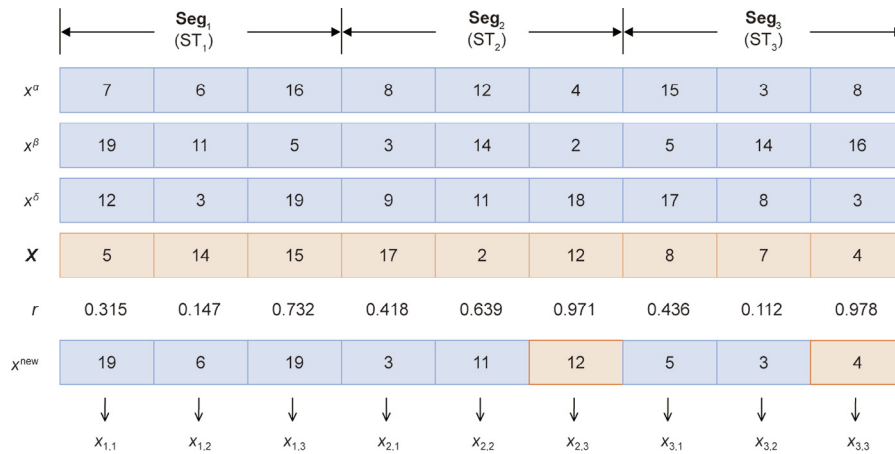


Fig. 4. Example of the crossover operator for X .

$$x_{ij}^{\text{new}} = \begin{cases} x_{ij}^\alpha, & \text{if } r < 0.25 \\ x_{ij}^\beta, & \text{else if } r < 0.50 \\ x_{ij}^\delta, & \text{else if } r < 0.75 \\ x_{ij}, & \text{otherwise} \end{cases} \quad (27)$$

For Y , the original crossover operator in the grey wolf optimizer is adopted, as shown in Eqs. (28)–(31), where a linearly decreases from 2 to 0 during the iteration, and y_{ij}^{new} , y_{ij} , y_{ij}^α , y_{ij}^β , and y_{ij}^δ are the Y elements of the new solution, the original solution, α , β , and δ .

$$y_{ij}^1 = y_{ij}^\alpha - a \cdot (2 \cdot \text{rand}() - 1) \cdot |2 \cdot \text{rand}() \cdot y_{ij}^\alpha - y_{ij}| \quad (28)$$

$$y_{ij}^2 = y_{ij}^\beta - a \cdot (2 \cdot \text{rand}() - 1) \cdot |2 \cdot \text{rand}() \cdot y_{ij}^\beta - y_{ij}| \quad (29)$$

$$y_{ij}^3 = y_{ij}^\delta - a \cdot (2 \cdot \text{rand}() - 1) \cdot |2 \cdot \text{rand}() \cdot y_{ij}^\delta - y_{ij}| \quad (30)$$

$$y_{ij}^{\text{new}} = \frac{1}{3} \sum_{p=1,2,3} y_{ij}^p \quad (31)$$

Regarding multi-objective optimization, most research introduces an external archive into the grey wolf optimizer to preserve the non-dominated solutions obtained so far, and utilizes a roulette-wheel method to select leaders from the archive. However, because only the best solutions are preserved, the leaders selected from the archive are likely to cause the algorithm to fall into a local optimum. Therefore, this paper adopts fast non-dominated sorting to select solutions for the next population.

When selecting leaders, all solutions in the population have the same probability of being selected. In this way, solutions that are not in the first Pareto front have a chance to be selected, which is conducive to jumping out of the local optimal and fully exploring the search space. Details about fast non-dominated sorting can be found in Ref. [48].

4.4. Competition-based variable neighborhood search

The local search is executed by a competition-based variable neighborhood search. Two types of local search operators are individually designed for service selection and amount distribution, and each type contains four operators. When the local search is performed on a solution, the local search type is randomly determined first, and then a specific search operator from the type is selected according to the probabilities updated by a competition mechanism.

4.4.1. Local search for service selection

Since high-quality services can significantly improve collaboration solutions, the local search for service selection is designed to replace low-quality services with high-quality services. This type includes three objective-oriented operators, OS_1 , OS_2 , and OS_3 , and a hybrid operator, OS_4 . Each objective-oriented operator corresponds to one objective to be optimized and performs the same related operation on all segments in a solution. The hybrid operator randomly performs the corresponding operations of different objectives on each segment in a solution and is dedicated to maintaining a balance. The details of the operators are as follows:

(1) **OS₁ for total cost f_1** : For each **Seg_{*i*}** in **X**, the operator finds the service with the highest unit cost in **Seg_{*i*}** and replaces it with a randomly selected service with a lower unit price from **CSS_{*i*}**.

(2) **OS₂ for reliability f_2** : For each **Seg_{*i*}** in **X**, the operator finds the service with the lowest reliability in **Seg_{*i*}** and replaces it with a randomly selected service with higher reliability from **CSS_{*i*}**.

(3) **OS₃ for finishing time f_3** : For each **Seg_{*i*}** in **X**, the operator finds the service with the lowest speed in **Seg_{*i*}** and replaces it with a randomly selected service with a higher speed from **CSS_{*i*}**. More specifically, if the service is a service chain, the lowest speed of the internal component services is regarded as the speed of the service chain.

(4) **OS₄ for all objectives**: For each **Seg_{*i*}** in **X**, the operator randomly selects an objective to be optimized and modifies **Seg_{*i*}** according to the above corresponding operator. In this way, different service selection operations to optimize different objectives can be performed with equal probability on a solution.

4.4.2. Local search for amount distribution

Regarding total cost and reliability, collaboration solutions will be better if the relatively low-quality service among the selected services is distributed to less processing amount of a subtask. Meanwhile, services are expected to be fully utilized to reduce the finishing time. Therefore, it is preferable to distribute the processing amount in a way that is proportional to the speeds of the services selected for a subtask. Accordingly, a local search for amount distribution is designed, including three objective-oriented operators, **OA₁**, **OA₂**, and **OA₃**, and a hybrid operator, **OA₄**. The details of the operators are as follows:

(1) **OA₁ for total cost f_1** : For each **Seg_{*i*}** in **Y**, the operator finds the service with the highest unit cost in **Seg_{*i*}** and decreases the corresponding y_{ij} according to Eq. (32), where y'_{ij} is the value before decreasing.

(2) **OA₂ for reliability f_2** : For each **Seg_{*i*}** in **Y**, the operator finds the service with the lowest reliability in **Seg_{*i*}** and decreases the corresponding y_{ij} according to Eq. (32).

(3) **OA₃ for finishing time f_3** : For each **Seg_{*i*}** in **Y**, the operator adjusts all y_{ij} belonging to **Seg_{*i*}** according to Eq. (33), where $p_{i,x_{ij}}$ is the speed of the current service and x_{iw} is the index of each selected service in **Seg_{*i*}**. Notably, the lowest speed of the internal component services is regarded as the speed of the service chain.

(4) **OA₄ for all objectives**: For each **Seg_{*i*}** in **Y**, the operator randomly selects an objective to be optimized and modifies **Seg_{*i*}** according to the above corresponding operator. As in **OS₄**, different amount distribution operations can be performed with equal probability on a solution.

$$y_{ij} = \text{rand}() \cdot y'_{ij} \tag{32}$$

$$y_{ij} = \frac{p_{i,x_{ij}}}{\sum_{x_{iw} \in \text{Seg}_i} p_{i,x_{iw}}} \tag{33}$$

4.4.3. Competition mechanism

With the iterative execution of the algorithm, the optimization progress of different objectives is usually different. A competition mechanism is developed to assign more computing resources to the objective with more space for improvement. In the mechanism, operators with the same type compete to obtain a higher selection probability. The new probabilities are updated depending on the effects of the operators on the solutions. If an operator improves a solution, it will be incentivized to obtain a higher probability; otherwise, it will be suppressed.

To quantify the effect of operators in the current iteration, the population is updated after the local search, and solutions with the local search operator applied are identified and classified.

Taking the local search for service selection as an example, the effect of **OS_{*g*}** is formulated as Eqs. (34) and (35), where g (or h) is the index of optimization objectives. Eq. (34) is for objective-oriented operators and takes the focused objective as the primary factor. Other objectives are included as secondary factors to avoid their excessive deterioration. η in Eq. (34) is a coordination coefficient used to control the weights among the factors; it is greater than 1/3 in order to highlight the effect on the focused objective. Eq. (35) is for the hybrid operator that considers all the objectives. U_{OS_g} is the set containing the solutions that applied **OS_{*g*}**, CS' is the original solution before the local search operator is applied, $f_g(CS)$ and $f_g(CS')$ are the g th objectives of CS and CS' , $f_h(CS)$ and $f_h(CS')$ are the h th objectives of CS and CS' , and ε is a small number preventing the denominator from being 0.

$$e_{OS_g} = \sum_{CS \in U_{OS_g}} \left[\eta \cdot \left(\frac{f_g(CS') - f_g(CS)}{f_g(CS') + \varepsilon} \right) + \frac{1 - \eta}{2} \cdot \sum_{\substack{h=1 \\ h \neq g}}^3 \left(\frac{f_h(CS') - f_h(CS)}{f_h(CS') + \varepsilon} \right) \right], \tag{34}$$

$$g = 1, 2, 3$$

$$e_{OS_4} = \frac{1}{3} \cdot \sum_{CS \in U_{OS_4}} \sum_{h=1}^3 \left(\frac{f_h(CS') - f_h(CS)}{f_h(CS') + \varepsilon} \right) \tag{35}$$

In the above formulas, e_{OS_g} will increase if **OS_{*g*}** improves a solution and will decrease if a solution deteriorates, so that the incentives or suppressions can be achieved. To prevent e_{OS_g} from being negative or 0, a correction function $H(e_{OS_g})$ is presented in Eq. (36), where μ_{OS} is a small number. μ'_{OS} in Eq. (37) is the value of μ_{OS} in the last iteration. In this paper, μ_{OS} is set to 0.01 at the beginning of the algorithm. Since only considering effects in the current iteration may lead to drastic changes in the selection probabilities and thus result in instability of the algorithm, the historical effects of operators are introduced to alleviate the changes. As the probabilities used in the current iteration carry historical information, they are deployed to update the selection probabilities. Eq. (38) presents the formula for updating the selection probability of **OS_{*g*}**, where p_{OS_g} is the selection probability used in the current iteration, and $p_{OS_g}^*$ is the updated probability to be used in the next iteration. Without loss of generality, the selection probabilities of the operators are equal at the beginning. Similarly, the selection probability of **OA_{*g*}** can be updated.

$$H(e_{OS_g}) = \max(e_{OS_g}, \mu_{OS}) \tag{36}$$

$$\mu_{OS} = \begin{cases} 0.01 \cdot \max_{h=1}^4(e_{OS_h}), & \text{if } \max_{h=1}^4(e_{OS_h}) > 0 \\ \mu'_{OS}, & \text{otherwise} \end{cases} \tag{37}$$

$$p_{OS_g}^* = \frac{\sqrt{p_{OS_g} \cdot H(e_{OS_g})}}{\sum_{h=1}^4 \sqrt{p_{OS_h} \cdot H(e_{OS_h})}} \tag{38}$$

5. Implementation through numerical experiments

Extensive numerical experiments were conducted to test the performance of CMOMA in solving the proposed problem. Each solution of the experiments determined the schemes of the service selection and amount distribution to the selected services, which were simultaneously optimized as the algorithm evolved. All programs for the experiments were coded with Java (JDK 14) and executed on a laptop with 32 GB RAM and a 2.2 GHz Intel Core i7-8750H under a Windows 10 (64) system. Parallel technology was implemented in the decoding stage for speed.

5.1. Performance metric and test instance

In the experiments, four indicators were adopted as performance metrics: generational distance (GD), inverted GD (IGD), hypervolume (HV), and set coverage (SC).

(1) GD is a convergence metric defined as Eq. (39), where U is the non-dominated solution set obtained by a specific algorithm, P is the true Pareto-optimal front, and $d_{p,u}$ is the Euclidean distance between solutions p and u . A smaller GD value indicates a solution set with a better convergence performance.

$$GD = \frac{1}{|U|} \cdot \sqrt{\sum_{u \in U} \min_{p \in P} (d_{p,u})} \tag{39}$$

(2) IGD is a comprehensive metric defined as Eq. (40), which simultaneously reflects the convergence and diversity of a solution set. A smaller IGD value indicates a solution set with a better comprehensive performance.

$$IGD = \frac{1}{|P|} \cdot \sqrt{\sum_{p \in P} \min_{u \in U} (d_{p,u})} \tag{40}$$

(3) HV calculates the volume covered by a solution set in objective space; it is defined as Eq. (41), where c_u is the hypercube constructed by the line segment between u and the reference point as the diagonal. Since Eqs. (22)–(24) have normalized objectives, the reference point in this paper is (1,1,1). A solution set with a bigger HV value is preferable.

$$HV = \text{volume} \left(\bigcup_{u \in U} c_u \right) \tag{41}$$

(4) SC measures the dominance relationship between two non-dominated solution sets U and V , through which the quality of the two sets can be compared. SC is defined as Eq. (42), where $C(U, V)$ is the proportion of solutions in V that are dominated by solutions in U , and $u > v$ means that u dominates v . If $C(U, V)$ is bigger than $C(V, U)$, U has a higher quality.

$$C(U, V) = \frac{|\{v \in V | \exists u \in U : u > v\}|}{|V|} \tag{42}$$

Test instances were generated based on the principles adopted in Ref. [40]. Services with high reliability or fast processing speed were given high unit costs, and vice versa. As shown in Table 4, 21 instances were generated to fully evaluate the algorithm, where the number of subtasks came in three sizes: 15, 30, and 45. In each size, the proportion of subtasks using complicated services—that is, composite services and service chains—increased from 20% to 80%, with a step size of 10%. The available periods of resource services (including the component resource services in complicated services) were generated randomly. Each subtask has 50 candidate services in this paper, and up to three services can be selected.

Table 4
The proportion of different services in test instances.

Test instance	Number of subtasks	Test instance	Number of subtasks	Test instance	Number of subtasks	Proportion		
						Service chain	Composite service	Resource service
1	15	8	30	15	45	10%	10%	80%
2		9		16		15%	15%	70%
3		10		17		20%	20%	60%
4		11		18		25%	25%	50%
5		12		19		30%	30%	40%
6		13		20		35%	35%	30%
7		14		21		40%	40%	20%

5.2. Sensitivity analysis of the coordination coefficient η

Deployed to update the selection probabilities of local search operators, the parameter η in the competition mechanism is the sole parameter to be tuned in CMOMA, and impacts the performance of the algorithm. Theoretically, a big η value causes the algorithm to concentrate on the effect on the oriented objective when updating the selection probabilities. In contrast, a small η value causes the algorithm to notice the effects on the other objectives.

To study the impact of η on CMOMA and select a suitable value of η , a group of experiments was conducted. First, four values, 0.4, 0.6, 0.8, and 1.0, were tested to roughly determine a reasonable scope of η . At each value, CMOMA was independently executed 20 times on the medium-sized instance 11; the running time for each execution was 15 s. After preliminary experiments, the population size was set to 200. Fig. 5 shows the means of the IGD with the standard deviations. The true Pareto-optimal front consists of the non-dominated solutions obtained by running CMOMA at all η values. As shown in Fig. 5, $\eta > 0.8$ is a desirable scope for CMOMA, where the algorithm maintains a preferable performance. In other words, when updating the selection probabilities, considering more effects on the oriented objective benefits the performance of the algorithm. To refine η , further tests were performed in [0.8, 1.0] with a step size of 0.02. Fig. 6 depicts the means of HV with the 95% confidence intervals. Although it is not apparent, the η in [0.9, 1.0] has superiorities over that in [0.8, 0.9]. However, with an increase in η , the performance of the algorithm deteriorates. In particular, at the point $\eta = 1$, where no effect of other objectives is included, noticeable performance degradation can be observed. This phenomenon suggests that the effects of both the oriented objective and other objectives are essential, and a balance between the two aspects is required. Among the tested values, when $\eta = 0.9$, CMOMA reaches a higher mean of HV with a narrow confidence interval. Therefore, η is set to 0.9 for the trade-off. In addition, from a global perspective, the trajectory of HV shows slight fluctuations but generally remains stable when $\eta > 0.8$, illustrating that η is insensitive if it is within an appropriate scope, which supports the robustness of CMOMA.

5.3. Performance verification of the competition mechanism

To verify the effectiveness of the competition mechanism, a variant of CMOMA, fixed-probability multi-objective memetic algorithm (FMOMA), was developed for comparison; this variant removes the competition mechanism and adopts fixed equal probabilities to select local search operators. Both algorithms were repeated 20 times on each instance with the same running time for each repetition (10, 15, and 20 s for small, medium-sized, and large instances, respectively). Table 5 presents the means of the GD, IGD, and HV. To calculate the IGD, the results of three well-known multi-objective algorithms—namely, NSGA-II [48], strength Pareto

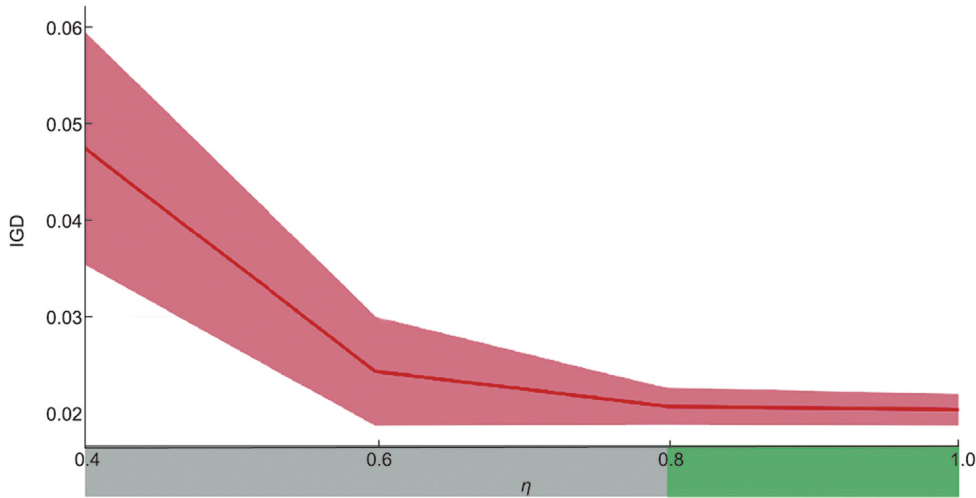


Fig. 5. Means of the IGD with standard deviations at different η values on test instance 11.

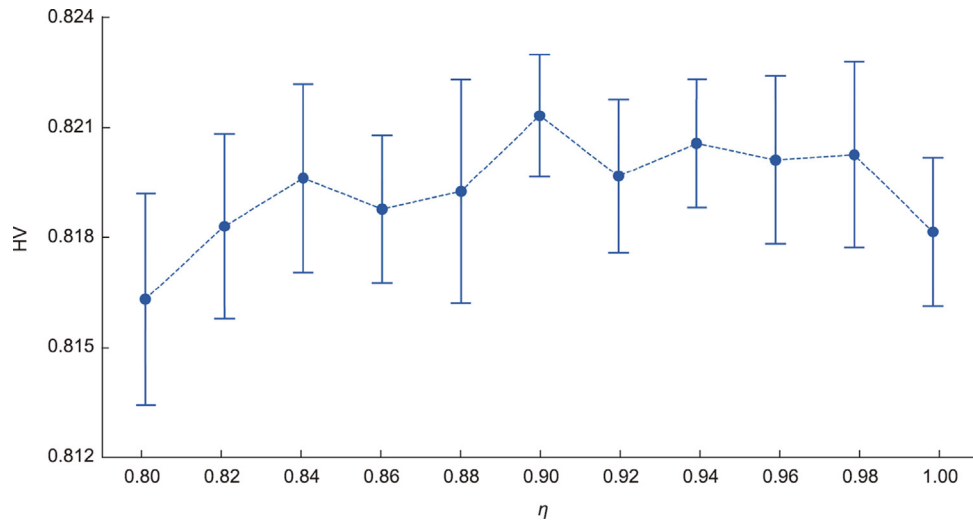


Fig. 6. Means of HV with 95% confidence intervals at different η values on test instance 11.

Table 5
Means and two-sample *t*-test results of GD, IGD, and HV for CMOMA and FMOMA on all test instances.

Test instance	GD		IGD		HV	
	CMOMA	FMOMA	CMOMA	FMOMA	CMOMA	FMOMA
1	1.1402×10^{-3}	$1.3180 \times 10^{-3}+$	2.3340×10^{-2}	$2.3355 \times 10^{-2}=\text{}$	7.9977×10^{-1}	$7.9802 \times 10^{-1}=\text{}$
2	8.9269×10^{-4}	$1.0554 \times 10^{-3}+$	2.1670×10^{-2}	$2.2390 \times 10^{-2}+$	8.2696×10^{-1}	$8.2328 \times 10^{-1}+$
3	9.2624×10^{-4}	$1.1366 \times 10^{-3}+$	2.0485×10^{-2}	$2.1510 \times 10^{-2}+$	8.1010×10^{-1}	$8.0836 \times 10^{-1}=\text{}$
4	1.0683×10^{-3}	$1.3717 \times 10^{-3}+$	1.9490×10^{-2}	$2.1390 \times 10^{-2}+$	8.5031×10^{-1}	$8.4925 \times 10^{-1}=\text{}$
5	9.0086×10^{-4}	$1.2245 \times 10^{-3}+$	2.0115×10^{-2}	$2.1690 \times 10^{-2}+$	8.5315×10^{-1}	$8.5117 \times 10^{-1}+$
6	8.8471×10^{-4}	$1.1432 \times 10^{-3}+$	1.7660×10^{-2}	$1.8415 \times 10^{-2}+$	8.5614×10^{-1}	$8.5769 \times 10^{-1}=\text{}$
7	1.0415×10^{-3}	$1.4650 \times 10^{-3}+$	1.7275×10^{-2}	$1.8140 \times 10^{-2}+$	8.8252×10^{-1}	$8.8138 \times 10^{-1}=\text{}$
8	7.9311×10^{-4}	$1.0661 \times 10^{-3}+$	1.9570×10^{-2}	$2.0410 \times 10^{-2}+$	7.7647×10^{-1}	$7.7718 \times 10^{-1}=\text{}$
9	8.7197×10^{-4}	$1.0704 \times 10^{-3}+$	1.9740×10^{-2}	$1.9670 \times 10^{-2}=\text{}$	8.0296×10^{-1}	$7.9883 \times 10^{-1}+$
10	7.6937×10^{-4}	$1.0079 \times 10^{-3}+$	1.9885×10^{-2}	$2.0125 \times 10^{-2}=\text{}$	8.1026×10^{-1}	$8.0704 \times 10^{-1}+$
11	7.8789×10^{-4}	$1.0368 \times 10^{-3}+$	2.0765×10^{-2}	$2.1170 \times 10^{-2}=\text{}$	8.2136×10^{-1}	$8.1340 \times 10^{-1}+$
12	7.7134×10^{-4}	$1.0219 \times 10^{-3}+$	1.7870×10^{-2}	$1.9090 \times 10^{-2}+$	7.9085×10^{-1}	$7.8832 \times 10^{-1}=\text{}$
13	8.7732×10^{-4}	$1.0757 \times 10^{-3}+$	1.8085×10^{-2}	$1.9780 \times 10^{-2}+$	8.4326×10^{-1}	$8.3691 \times 10^{-1}+$
14	7.9210×10^{-4}	$1.0462 \times 10^{-3}+$	1.8105×10^{-2}	$1.9375 \times 10^{-2}+$	8.3489×10^{-1}	$8.2903 \times 10^{-1}+$
15	7.4785×10^{-4}	$1.0317 \times 10^{-3}+$	1.9610×10^{-2}	$2.0175 \times 10^{-2}=\text{}$	7.6262×10^{-1}	$7.5633 \times 10^{-1}+$
16	7.0934×10^{-4}	$1.0484 \times 10^{-3}+$	1.8290×10^{-2}	$2.0220 \times 10^{-2}+$	7.9199×10^{-1}	$7.8373 \times 10^{-1}+$
17	6.9929×10^{-4}	$1.0574 \times 10^{-3}+$	1.9605×10^{-2}	$2.0405 \times 10^{-2}=\text{}$	7.9177×10^{-1}	$7.8538 \times 10^{-1}+$
18	7.0551×10^{-4}	$1.0221 \times 10^{-3}+$	1.7440×10^{-2}	$1.8970 \times 10^{-2}+$	8.0302×10^{-1}	$7.9714 \times 10^{-1}+$
19	8.1797×10^{-4}	$1.0681 \times 10^{-3}+$	1.8055×10^{-2}	$2.0265 \times 10^{-2}+$	8.1730×10^{-1}	$8.0498 \times 10^{-1}+$
20	8.2308×10^{-4}	$1.1589 \times 10^{-3}+$	1.8810×10^{-2}	$2.0905 \times 10^{-2}+$	8.0723×10^{-1}	$7.9788 \times 10^{-1}+$
21	7.6813×10^{-4}	$1.1156 \times 10^{-3}+$	1.8935×10^{-2}	$2.1415 \times 10^{-2}+$	8.0504×10^{-1}	$7.9633 \times 10^{-1}+$

The symbols “+” and “=” are respectively used below to indicate that the results of CMOMA are statistically superior or similar to those of FMOMA.

evolutionary algorithm-2 (SPEA-2) [49], and multi-objective Grey Wolf optimizer (MOGWO) [50], which will be discussed in the following section in detail—were deployed to construct the true Pareto-optimal front. An independent two-sample *t*-test with a 0.05 significance level was performed on each instance. The symbols “+” and “=” are respectively used below to indicate that the results of CMOMA are statistically superior or similar to those of FMOMA. It is clear that CMOMA obtains the best GD values on all instances. This advantage is also confirmed from a statistical perspective, demonstrating the preferable convergence of CMOMA. For the IGD, CMOMA achieves the best values on 20 instances, among which the results on 15 instances are significantly superior to those of FMOMA. In terms of HV, CMOMA always yields better results than FMOMA, despite their similar values on seven instances in the statistical sense. To sum up, CMOMA outstrips FMOMA in terms of comprehensive metrics.

To assess the quality, Table 6 compares the means of SC. $C(\text{CMOMA}, \text{FMOMA})$ are manifestly bigger than $C(\text{FMOMA}, \text{CMOMA})$ on all instances, suggesting the advantage of CMOMA in searching for high-quality solutions. To obtain more intuitive insights, Fig. 7 depicts the SC results with the linear regression trend lines. The superiority of CMOMA becomes apparent as the instance size increases. In particular, on large-sized instances, approximately half of the solutions of FMOMA are dominated by those of CMOMA. On instances with the same scale, a similar conclusion is observed. As the proportion of complicated services increases, the advantage of CMOMA is enhanced. This can be primarily attributed to the competition mechanism enabling the algorithm to find appropriate selection probabilities for local search operators. Combining all the above analyses confirms that CMOMA

surpasses FMOMA and that the designed competition mechanism contributes to the performance.

5.4. Comparisons with other algorithms

To further evaluate the performance of CMOMA, comparisons were made with both classical and emerging multi-objective algorithms: namely, NSGA-II, SPEA-2, and MOGWO. The experiment settings remained the same as those described in Section 5.3. Moreover, the Taguchi method was executed to tune the parameters of the comparison algorithms for optimal performance. For NSGA-II, the crossover rate is 1, and the mutation rate is 0.02; for SPEA-2, the crossover rate is 1, and the mutation rate is 0.03. MOGWO requires no predefined parameter.

Tables 7–10 present the comparison results of the GD, IGD, HV, and SC. It can be seen that CMOMA achieves the best results in all metrics, showing impressive performance. Regarding the GD, CMOMA significantly outperforms NSGA-II on 20 out of 21 instances and significantly outperforms SPEA-2 and MOGWO on all instances. In terms of the IGD, CMOMA overwhelms the comparison algorithms. NSGA-II and SPEA-2 show a comparable performance with each other, outstripping MOGWO. However, their results are far inferior to those of CMOMA, particularly on large instances. For HV, the results of CMOMA on all instances remain at around 0.8, which means that the non-dominated solutions of CMOMA consistently cover a large volume of the objective space. Conversely, the performances of NSGA-II, SPEA-2, and MOGWO are inferior, and significant performance degradations can be seen on medium-sized and large instances. As for the quality comparison, CMOMA has tremendous advantages. The solutions

Table 6
A comparison of the means of SC between CMOMA and FMOMA on all test instances.

Test instance	$C(\text{CMOMA}, \text{FMOMA})$	$C(\text{FMOMA}, \text{CMOMA})$	Test instance	$C(\text{CMOMA}, \text{FMOMA})$	$C(\text{FMOMA}, \text{CMOMA})$
1	27.75%	17.95%	12	37.35%	11.25%
2	28.55%	14.43%	13	42.05%	11.63%
3	31.23%	13.70%	14	41.00%	12.83%
4	31.38%	16.03%	15	36.05%	9.75%
5	34.40%	13.25%	16	43.93%	8.58%
6	30.20%	14.65%	17	44.25%	6.35%
7	28.88%	16.58%	18	44.05%	8.65%
8	32.78%	10.33%	19	41.53%	10.78%
9	33.00%	11.00%	20	50.03%	8.10%
10	33.38%	11.70%	21	53.33%	8.18%
11	41.13%	9.33%			

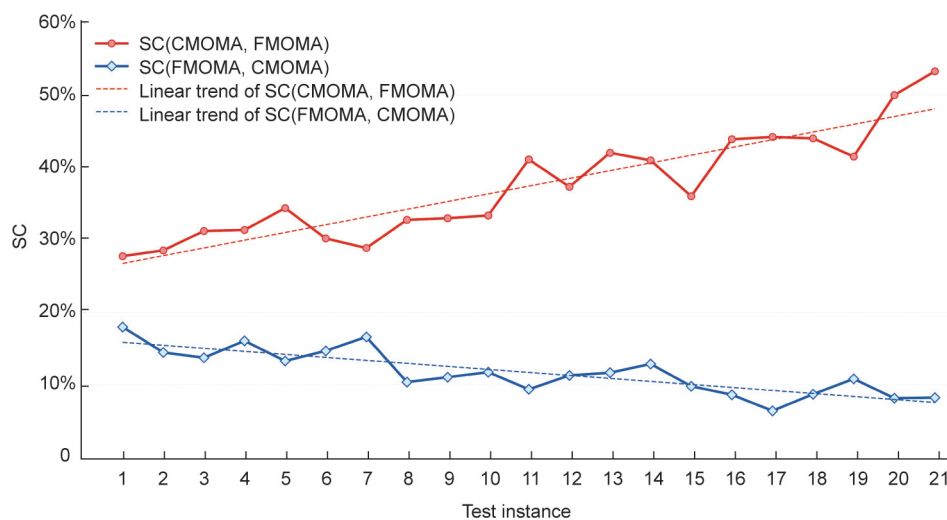


Fig. 7. SC results for CMOMA and FMOMA on all test instances.

Table 7
Means and two-sample *t*-test results of GD for CMOMA, NSGA-II, SPEA-2, and MOGWO on all test instances.

Test instance	CMOMA	NSGA-II	SPEA-2	MOGWO
1	1.1402×10^{-3}	$1.1716 \times 10^{-3} =$	$1.8564 \times 10^{-3} +$	$2.5177 \times 10^{-3} +$
2	8.9269×10^{-4}	$1.1832 \times 10^{-3} +$	$1.5757 \times 10^{-3} +$	$2.3667 \times 10^{-3} +$
3	9.2624×10^{-4}	$1.2792 \times 10^{-3} +$	$1.9651 \times 10^{-3} +$	$2.6771 \times 10^{-3} +$
4	1.0683×10^{-3}	$1.3604 \times 10^{-3} +$	$1.9080 \times 10^{-3} +$	$3.1077 \times 10^{-3} +$
5	9.0086×10^{-4}	$1.3917 \times 10^{-3} +$	$2.1230 \times 10^{-3} +$	$3.7516 \times 10^{-3} +$
6	8.8471×10^{-4}	$1.2978 \times 10^{-3} +$	$1.6351 \times 10^{-3} +$	$2.7415 \times 10^{-3} +$
7	1.0415×10^{-3}	$1.3033 \times 10^{-3} +$	$1.8277 \times 10^{-3} +$	$4.1431 \times 10^{-3} +$
8	7.9311×10^{-4}	$8.9139 \times 10^{-4} +$	$1.0697 \times 10^{-3} +$	$1.6005 \times 10^{-3} +$
9	8.7197×10^{-4}	$1.1159 \times 10^{-3} +$	$1.3992 \times 10^{-3} +$	$2.2571 \times 10^{-3} +$
10	7.6937×10^{-4}	$9.7697 \times 10^{-4} +$	$1.1289 \times 10^{-3} +$	$2.0441 \times 10^{-3} +$
11	7.8789×10^{-4}	$1.0573 \times 10^{-3} +$	$1.2310 \times 10^{-3} +$	$2.1723 \times 10^{-3} +$
12	7.7134×10^{-4}	$9.5573 \times 10^{-4} +$	$1.2966 \times 10^{-3} +$	$2.2548 \times 10^{-3} +$
13	8.7732×10^{-4}	$1.0020 \times 10^{-3} +$	$1.5837 \times 10^{-3} +$	$3.1690 \times 10^{-3} +$
14	7.9210×10^{-4}	$1.0903 \times 10^{-3} +$	$1.2610 \times 10^{-3} +$	$2.4987 \times 10^{-3} +$
15	7.4785×10^{-4}	$1.0657 \times 10^{-3} +$	$1.2482 \times 10^{-3} +$	$1.9489 \times 10^{-3} +$
16	7.0934×10^{-4}	$1.0747 \times 10^{-3} +$	$1.1446 \times 10^{-3} +$	$1.8948 \times 10^{-3} +$
17	6.9929×10^{-4}	$9.4979 \times 10^{-4} +$	$1.1071 \times 10^{-3} +$	$2.2707 \times 10^{-3} +$
18	7.0551×10^{-4}	$9.8959 \times 10^{-4} +$	$1.2360 \times 10^{-3} +$	$2.4601 \times 10^{-3} +$
19	8.1797×10^{-4}	$1.1376 \times 10^{-3} +$	$1.2641 \times 10^{-3} +$	$2.2323 \times 10^{-3} +$
20	8.2308×10^{-4}	$1.1959 \times 10^{-3} +$	$1.4157 \times 10^{-3} +$	$2.7575 \times 10^{-3} +$
21	7.6813×10^{-4}	$1.1232 \times 10^{-3} +$	$1.1994 \times 10^{-3} +$	$2.6607 \times 10^{-3} +$

Table 8
Means and two-sample *t*-test results of IGD for CMOMA, NSGA-II, SPEA-2, and MOGWO on all test instances.

Test instance	CMOMA	NSGA-II	SPEA-2	MOGWO
1	2.3340×10^{-2}	$7.2065 \times 10^{-2} +$	$1.0358 \times 10^{-1} +$	$1.5891 \times 10^{-1} +$
2	2.1670×10^{-2}	$6.1960 \times 10^{-2} +$	$9.6220 \times 10^{-2} +$	$1.4701 \times 10^{-1} +$
3	2.0485×10^{-2}	$5.6040 \times 10^{-2} +$	$8.9475 \times 10^{-2} +$	$1.4169 \times 10^{-1} +$
4	1.9490×10^{-2}	$6.0020 \times 10^{-2} +$	$8.8250 \times 10^{-2} +$	$1.3948 \times 10^{-1} +$
5	2.0115×10^{-2}	$7.9250 \times 10^{-2} +$	$1.0278 \times 10^{-1} +$	$1.5871 \times 10^{-1} +$
6	1.7660×10^{-2}	$5.7635 \times 10^{-2} +$	$9.4770 \times 10^{-2} +$	$1.5045 \times 10^{-1} +$
7	1.7275×10^{-2}	$7.0960 \times 10^{-2} +$	$8.8365 \times 10^{-2} +$	$1.5922 \times 10^{-1} +$
8	1.9570×10^{-2}	$1.0879 \times 10^{-1} +$	$1.2709 \times 10^{-1} +$	$1.8022 \times 10^{-1} +$
9	1.9740×10^{-2}	$1.3424 \times 10^{-1} +$	$1.4818 \times 10^{-1} +$	$2.0281 \times 10^{-1} +$
10	1.9885×10^{-2}	$1.4195 \times 10^{-1} +$	$1.5176 \times 10^{-1} +$	$2.0302 \times 10^{-1} +$
11	2.0765×10^{-2}	$1.3806 \times 10^{-1} +$	$1.4424 \times 10^{-1} +$	$1.9940 \times 10^{-1} +$
12	1.7870×10^{-2}	$1.5765 \times 10^{-1} +$	$1.5572 \times 10^{-1} +$	$2.1148 \times 10^{-1} +$
13	1.8085×10^{-2}	$1.4425 \times 10^{-1} +$	$1.4284 \times 10^{-1} +$	$1.9310 \times 10^{-1} +$
14	1.8105×10^{-2}	$1.5876 \times 10^{-1} +$	$1.5185 \times 10^{-1} +$	$2.0712 \times 10^{-1} +$
15	1.9610×10^{-2}	$1.6540 \times 10^{-1} +$	$1.6076 \times 10^{-1} +$	$2.1196 \times 10^{-1} +$
16	1.8290×10^{-2}	$1.8027 \times 10^{-1} +$	$1.7266 \times 10^{-1} +$	$2.2438 \times 10^{-1} +$
17	1.9605×10^{-2}	$1.8536 \times 10^{-1} +$	$1.6972 \times 10^{-1} +$	$2.2557 \times 10^{-1} +$
18	1.7440×10^{-2}	$1.7498 \times 10^{-1} +$	$1.7019 \times 10^{-1} +$	$2.1648 \times 10^{-1} +$
19	1.8055×10^{-2}	$1.9341 \times 10^{-1} +$	$1.8550 \times 10^{-1} +$	$2.3061 \times 10^{-1} +$
20	1.8810×10^{-2}	$1.7300 \times 10^{-1} +$	$1.6607 \times 10^{-1} +$	$2.0930 \times 10^{-1} +$
21	1.8935×10^{-2}	$1.8195 \times 10^{-1} +$	$1.7194 \times 10^{-1} +$	$2.1275 \times 10^{-1} +$

Table 9
Means and two-sample *t*-test results of HV for CMOMA, NSGA-II, SPEA-2, and MOGWO on all test instances.

Test instance	CMOMA	NSGA-II	SPEA-2	MOGWO
1	7.9977×10^{-1}	$7.2323 \times 10^{-1} +$	$6.4637 \times 10^{-1} +$	$5.6612 \times 10^{-1} +$
2	8.2696×10^{-1}	$7.3936 \times 10^{-1} +$	$6.5293 \times 10^{-1} +$	$5.8092 \times 10^{-1} +$
3	8.1010×10^{-1}	$7.2721 \times 10^{-1} +$	$6.4423 \times 10^{-1} +$	$5.7386 \times 10^{-1} +$
4	8.5031×10^{-1}	$7.5772 \times 10^{-1} +$	$6.8929 \times 10^{-1} +$	$6.0273 \times 10^{-1} +$
5	8.5315×10^{-1}	$7.6331 \times 10^{-1} +$	$7.0096 \times 10^{-1} +$	$6.1010 \times 10^{-1} +$
6	8.5614×10^{-1}	$7.6245 \times 10^{-1} +$	$6.7799 \times 10^{-1} +$	$5.8590 \times 10^{-1} +$
7	8.8252×10^{-1}	$7.5996 \times 10^{-1} +$	$7.0769 \times 10^{-1} +$	$5.9014 \times 10^{-1} +$
8	7.7647×10^{-1}	$5.8710 \times 10^{-1} +$	$5.4731 \times 10^{-1} +$	$4.7669 \times 10^{-1} +$
9	8.0296×10^{-1}	$6.1238 \times 10^{-1} +$	$5.6919 \times 10^{-1} +$	$5.0286 \times 10^{-1} +$
10	8.1026×10^{-1}	$6.0328 \times 10^{-1} +$	$5.6409 \times 10^{-1} +$	$5.0319 \times 10^{-1} +$
11	8.2136×10^{-1}	$6.0780 \times 10^{-1} +$	$5.7310 \times 10^{-1} +$	$5.0382 \times 10^{-1} +$
12	7.9085×10^{-1}	$5.7480 \times 10^{-1} +$	$5.4932 \times 10^{-1} +$	$4.8258 \times 10^{-1} +$
13	8.4326×10^{-1}	$5.8471 \times 10^{-1} +$	$5.6201 \times 10^{-1} +$	$4.9909 \times 10^{-1} +$
14	8.3489×10^{-1}	$5.9109 \times 10^{-1} +$	$5.8402 \times 10^{-1} +$	$5.0734 \times 10^{-1} +$
15	7.6262×10^{-1}	$5.2343 \times 10^{-1} +$	$5.1068 \times 10^{-1} +$	$4.5009 \times 10^{-1} +$
16	7.9199×10^{-1}	$5.2316 \times 10^{-1} +$	$5.1511 \times 10^{-1} +$	$4.6096 \times 10^{-1} +$
17	7.9177×10^{-1}	$5.3436 \times 10^{-1} +$	$5.2781 \times 10^{-1} +$	$4.6419 \times 10^{-1} +$
18	8.0302×10^{-1}	$5.2764 \times 10^{-1} +$	$5.1421 \times 10^{-1} +$	$4.6993 \times 10^{-1} +$
19	8.1730×10^{-1}	$5.2730 \times 10^{-1} +$	$5.2423 \times 10^{-1} +$	$4.7525 \times 10^{-1} +$
20	8.0723×10^{-1}	$5.1571 \times 10^{-1} +$	$5.1236 \times 10^{-1} +$	$4.5845 \times 10^{-1} +$
21	8.0504×10^{-1}	$5.0620 \times 10^{-1} +$	$5.0290 \times 10^{-1} +$	$4.5930 \times 10^{-1} +$

Table 10

A comparison of the means of SC between CMOMA and NSGA-II, SPEA-2, and MOGWO on all test instances.

Test instance	C(CMOMA, NSGA-II)	C(NSGA-II, CMOMA)	C(CMOMA, SPEA-2)	C(SPEA-2, CMOMA)	C(CMOMA, MOGWO)	C(MOGWO, CMOMA)
1	20.93%	19.35%	55.43%	2.70%	89.64%	0.13%
2	32.73%	11.60%	54.65%	3.78%	92.09%	0.13%
3	39.73%	11.03%	75.53%	1.18%	97.25%	0.05%
4	39.53%	9.20%	63.05%	2.48%	95.83%	0.15%
5	45.25%	8.85%	66.08%	2.15%	96.74%	0.05%
6	55.08%	8.35%	65.93%	3.15%	98.28%	0.03%
7	47.53%	8.28%	66.45%	3.00%	99.51%	0
8	38.93%	6.25%	57.23%	1.35%	97.41%	0
9	56.88%	3.08%	77.70%	0.23%	99.86%	0
10	50.55%	4.78%	68.30%	1.05%	99.14%	0
11	61.75%	1.95%	84.78%	0.40%	100.00%	0
12	55.30%	3.68%	83.31%	0.65%	99.77%	0
13	49.60%	3.03%	79.80%	0.45%	100.00%	0
14	56.03%	2.70%	81.28%	0.45%	100.00%	0
15	57.63%	1.55%	77.43%	0.43%	99.37%	0
16	72.15%	0.73%	86.78%	0.18%	99.93%	0
17	70.30%	0.80%	89.95%	0.05%	100.00%	0
18	61.80%	1.08%	92.90%	0.05%	99.87%	0
19	63.74%	0.98%	83.06%	0.35%	100.00%	0
20	70.45%	0.58%	90.90%	0.05%	100.00%	0
21	82.11%	0.45%	98.20%	0.00%	100.00%	0

of CMOMA dominate most of the solutions of SPEA-2 and MOGWO. Except for the first instance, the solutions of CMOMA are much better than those of NSGA-II, particularly on large instances.

The above analyses reveal a phenomenon in which the superiority of CMOMA becomes more apparent as the instance size increases. To further explore this phenomenon, non-dominated solutions obtained by randomly running the algorithms on three instances of each size are shown in Fig. 8. On the small instance, NSGA-II reaches a limited Pareto front and presents a relatively good distribution, surpassing SPEA-2. However, on the medium-sized and large instances, the performance of NSGA-II degrades palpably. The solutions of NSGA-II, SPEA-2, and MOGWO are close and concentrated in a small area, failing to explore the objective space adequately. In contrast, the solutions of CMOMA on all instances occupy large spaces and cover most of the solutions of the comparison algorithms. This advantage mainly stems from the local search operators and the competition mechanism, which allow the algorithm to search broadly and efficiently.

To recap, this paper illustrates the efficacy of the proposed DMSCO methodology by preferentially investigating a sequential pattern in which the subtasks of a task are fulfilled in sequence. This is a fundamental pattern that can be effortlessly extended to other patterns. For example, to accommodate a pattern in which parallel subtasks exist, all research can be adopted with just two minor modifications. First, it is necessary to convert Eq. (20) $F = F_l$ to $F = \max_{CS_i \in CS} (F_i)$ and Eq. (21) $B_i \geq F_{i-1} \geq 0$ to $B_i \geq F_l \geq 0$, where l is the index of subtasks and subtask ST_l should be started after ST_l (as more than one precursor ST_l could exist with a parallel relationship before ST_l). Second, in the encoding scheme, it is necessary to ensure that a subtask is placed after all its precursor subtasks. In this way, this paper shows how DMSCO improves service collaboration solutions while being capable of providing references for other patterns and retaining extensibility.

6. Conclusions

With the application of Industrial Internet platforms in the manufacturing industry, high-quality manufacturing service collaboration solutions are desirable. This paper proposes a DMSCO methodology to optimize service selection and amount distribution in order to improve the solution quality. In this work, manu-

facturing services are modeled at three granularities, and their components and structures are discussed to ensure the feasibility of collaboration solutions. Unlike previous works, the availability of a manufacturing service is formulated as a series of accessible periods of the physical resource corresponding to the service. In particular, the availability of a complicated service is affected by the component services and their relationships. The DMSCO model is then formulated to simultaneously optimize service selection and amount distribution. A CMOMA embedding a competition mechanism is developed for searching for solutions to the model. Extensive experiments are conducted, and the results suggest the superiority of the proposed algorithm. To be specific, CMOMA outperforms FMOMA that is without the competition mechanism in all considered metrics, thereby verifying the effectiveness of the designed mechanism. Compared with NSGA-II, SPEA-2, and MOGWO, CMOMA also exhibits excellent performance. Moreover, as the size of the instances increases, the superiority of CMOMA over the comparison algorithms becomes evident.

This paper aims to introduce a novel DMSCO for Industrial Internet platforms, thereby providing fresh insights to stakeholders. Motivated by studies on optimal service selection, the developed DMSCO is comprehensible for relevant practitioners and easily compatible with existing processes. In parallel, the results of the experiments in this study reveal the proposed method's potential for practical applications in scenarios with complex production processes and large production amounts, such as automobile, equipment, and household appliance manufacturing. A link in the production process can be assigned to multiple qualified manufacturing services, thus enhancing the overall service performance. However, it should be noted that the manufacturing service modeling in this paper focuses on physical processing resources, while Industrial Internet platforms also contain other types of resources, such as software and logistics resources. For future work, the proposed method will be validated and evaluated through real-world scenarios and experiments. Research on enhancing the expressive ability of manufacturing service modeling and the corresponding service collaboration model is recommended. In addition, this paper mainly focuses on collaboration solutions that satisfy customer requirements. Since the interests of service providers are also crucial, it is worth studying how to simultaneously improve the satisfaction of both sides.

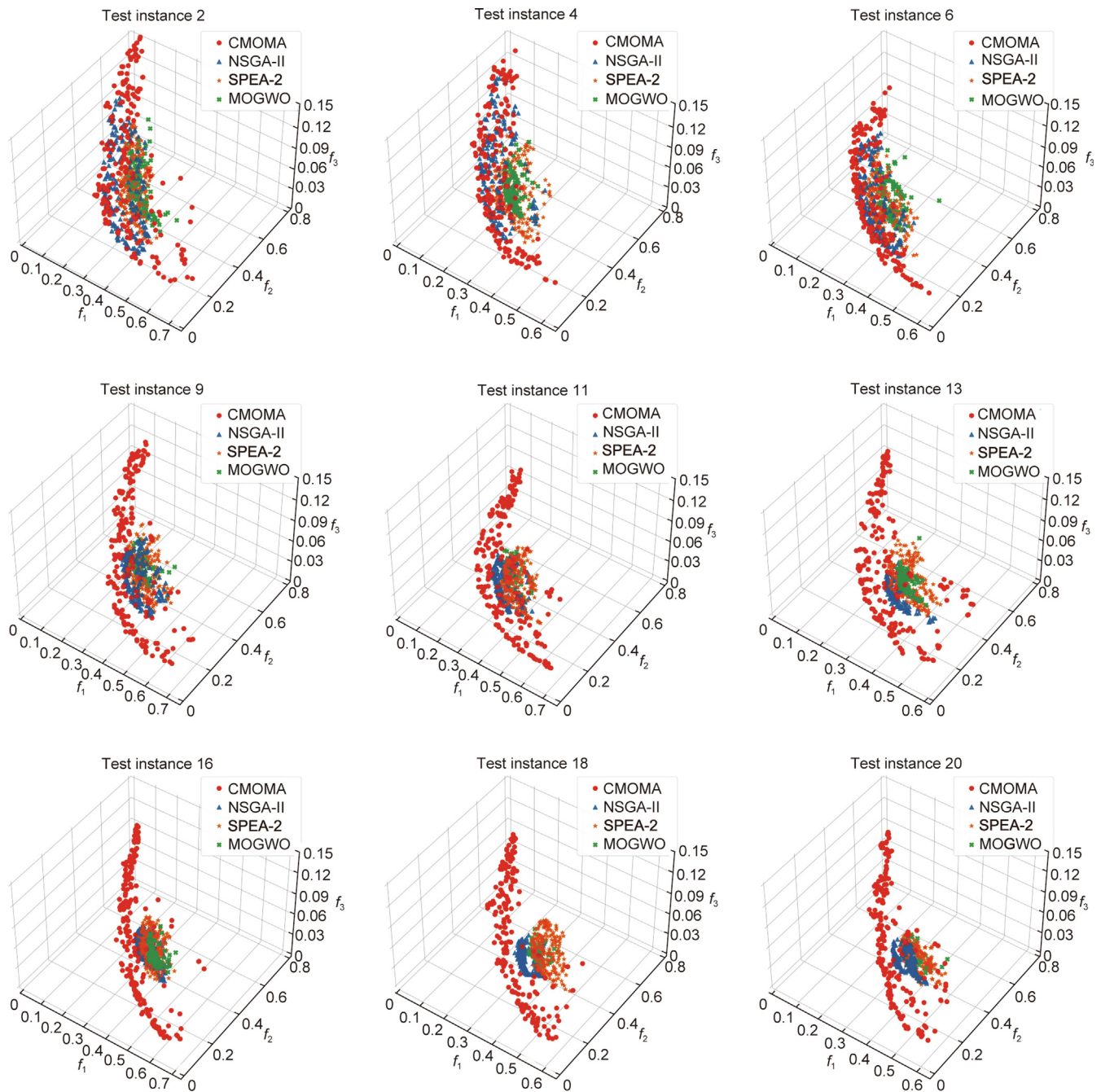


Fig. 8. Non-dominated solutions obtained by a random running of CMOMA, NSGA-II, SPEA-2, and MOGWO on small test instances (2, 4, and 6), medium-sized test instances (9, 11, and 13), and large test instances (16, 18, and 20).

Acknowledgments

This work is funded by the National Natural Science Foundation of China (51905396) and the China Scholarship Council.

Compliance with ethics guidelines

Shibao Pang, Shunsheng Guo, Xi Vicent Wang, Lei Wang, and Lihui Wang declare that they have no conflict of interest or financial conflicts to disclose.

References

- [1] Zhou J, Li P, Zhou Y, Wang B, Zang J, Meng L. Toward new-generation intelligent manufacturing. *Engineering* 2018;4(1):11–20.
- [2] Zhong RY, Xu X, Klotz E, Newman ST. Intelligent manufacturing in the context of Industry 4.0: a review. *Engineering* 2017;3(5):616–30.
- [3] Tao F, Qi Q, Wang L, Nee AYC. Digital twins and cyber-physical systems toward smart manufacturing and Industry 4.0: correlation and comparison. *Engineering* 2019;5(4):653–61.
- [4] Wang B. The future of manufacturing: a new perspective. *Engineering* 2018;4(5):722–8.
- [5] Cheng Y, Gao Y, Wang L, Tao F, Wang QG. Graph-based operational robustness analysis of Things platform for manufacturing service collaboration. *Int J Prod Res* 2022;2022:1–28.
- [6] Wang L, Gao T, Zhou B, Tang H, Xiang F. Manufacturing service recommendation method toward Industrial Internet platform considering the cooperative relationship among enterprises. *Expert Syst Appl* 2022;192:116391.
- [7] Song Y, Lin J, Tang M, Dong S. An Internet of Energy Things based on wireless LPWAN. *Engineering* 2017;3(4):460–6.
- [8] Lin H, Garg S, Hu J, Wang X, Piran MJ, Hossain MS. Data fusion and transfer learning empowered granular trust evaluation for Internet of Things. *Inf Fusion* 2022;78:149–57.

- [9] Zhou H, Yang C, Sun Y. Intelligent ironmaking optimization service on a cloud computing platform by digital twin. *Engineering* 2021;7(9):1274–81.
- [10] Wang XV, Wang L, Mohammed A, Givehchi M. Ubiquitous manufacturing system based on cloud: a robotics application. *Robot Comput Integr Manuf* 2017;45:116–25.
- [11] Cheng Y, Xie Y, Wang D, Tao F, Ji P. Manufacturing services scheduling with supply-demand dual dynamic uncertainties toward Industrial Internet platforms. *IEEE Trans Industr Inform* 2021;17(5):2997–3010.
- [12] Zhang X, Ming X. A comprehensive industrial practice for Industrial Internet Platform (IIP): general model, reference architecture, and industrial verification. *Comput Ind Eng* 2021;158:107426.
- [13] Li P, Cheng Y, Tao F. Failures detection and cascading analysis of manufacturing services collaboration toward Industrial Internet platforms. *J Manuf Syst* 2020;57:169–81.
- [14] Wang Y, Zhang Y, Tao F, Chen T, Cheng Y, Yang S. Logistics-aware manufacturing service collaboration optimisation towards Industrial Internet platform. *Int J Prod Res* 2019;57(12):4007–26.
- [15] Li P, Cheng Y, Song W, Tao F. Manufacturing services collaboration: connotation, framework, key technologies, and research issues. *Int J Adv Manuf Technol* 2020;110(9–10):2573–89.
- [16] Bouzary H, Chen FF. A classification-based approach for integrated service matching and composition in cloud manufacturing. *Robot Comput Integr Manuf* 2020;66:101989.
- [17] Yang B, Wang S, Li S, Jin T. A robust service composition and optimal selection method for cloud manufacturing. *Int J Prod Res* 2022;60(4):1134–52.
- [18] Liu Y, Wang L, Wang XV, Xu X, Zhang L. Scheduling in cloud manufacturing: state-of-the-art and research challenges. *Int J Prod Res* 2019;57(15–16):4854–79.
- [19] Shi SY, Mo R, Yang HC, Chang ZY, Chen ZF. An implementation of modelling resource in a manufacturing grid for resource sharing. *Int J Comput Integr Manuf* 2007;20(2–3):169–77.
- [20] Vichare P, Nassehi A, Kumar S, Newman ST. A unified manufacturing resource model for representing CNC machining systems. *Robot Comput Integr Manuf* 2009;25(6):999–1007.
- [21] Ameri F, McArthur C. Semantic rule modelling for intelligent supplier discovery. *Int J Comput Integr Manuf* 2014;27(6):570–90.
- [22] Wang XV, Wang L. A cloud-based production system for information and service integration: an Internet of Things case study on waste electronics. *Enterprise Inf Syst* 2017;11(7):952–68.
- [23] Li H, Chan KCC, Liang M, Luo X. Composition of resource-service chain for cloud manufacturing. *IEEE Trans Ind Inform* 2016;12(1):211–29.
- [24] Wu L. Resource virtualization model in cloud manufacturing. *Adv Mat Res* 2010;143–144:1250–3.
- [25] Liu N, Li X, Shen W. Multi-granularity resource virtualization and sharing strategies in cloud manufacturing. *J Netw Comput Appl* 2014;46:72–82.
- [26] Yu C, Zhang W, Xu X, Ji Y, Yu S. Data mining based multi-level aggregate service planning for cloud manufacturing. *J Intell Manuf* 2018;29(6):1351–61.
- [27] Zhang Z, Zhang Y, Lu J, Xu X, Gao F, Xiao G. CMfgIA: a cloud manufacturing application mode for industry alliance. *Int J Adv Manuf Technol* 2018;98(9–12):2967–85.
- [28] Li H, Weng S, Tong J, He T, Chen W, Sun M, et al. Composition of resource-service chain based on evolutionary algorithm in distributed cloud manufacturing systems. *IEEE Access* 2020;8:19911–20.
- [29] Lu Y, Xu X. A semantic web-based framework for service composition in a cloud manufacturing environment. *J Manuf Syst* 2017;42:69–81.
- [30] Ren M, Ren L, Jain H. Manufacturing service composition model based on synergy effect: a social network analysis approach. *Appl Soft Comput* 2018;70:288–300.
- [31] Zhou L, Zhang L, Horn BKP. Collaborative optimization for logistics and processing services in cloud manufacturing. *Robot Comput Integr Manuf* 2021;68:102094.
- [32] Wu Y, Jia G, Cheng Y. Cloud manufacturing service composition and optimal selection with sustainability considerations: a multi-objective integer bi-level multi-follower programming approach. *Int J Prod Res* 2020;58(19):6024–42.
- [33] Wang Y, Wang S, Gao S, Guo X, Yang B. Adaptive multi-objective service composition reconfiguration approach considering dynamic practical constraints in cloud manufacturing. *Knowl Base Syst* 2021;234:107607.
- [34] Tao F, Lai Li Y, Xu L, Zhang L. FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. *IEEE Trans Industr Inform* 2013;9(4):2023–33.
- [35] Wang L, Liu Z, Liu A, Tao F. Artificial intelligence in product lifecycle management. *Int J Adv Manuf Technol* 2021;114(3–4):771–96.
- [36] Bouzary H, Chen FF. A hybrid grey wolf optimizer algorithm with evolutionary operators for optimal QoS-aware service composition and optimal selection in cloud manufacturing. *Int J Adv Manuf Technol* 2019;101(9–12):2771–84.
- [37] Akbaripour H, Houshmand M. Service composition and optimal selection in cloud manufacturing: landscape analysis and optimization by a hybrid imperialist competitive and local search algorithm. *Neural Comput Appl* 2020;32(15):10873–94.
- [38] Zhang S, Xu Y, Zhang W. Multitask-oriented manufacturing service composition in an uncertain environment using a hyper-heuristic algorithm. *J Manuf Syst* 2021;60:138–51.
- [39] Zhou J, Yao X. A hybrid approach combining modified artificial bee colony and cuckoo search algorithms for multi-objective cloud manufacturing service composition. *Int J Prod Res* 2017;55(16):4765–84.
- [40] Zhang Y, Tao F, Liu Y, Zhang P, Cheng Y, Zuo Y. Long/short-term utility aware optimal selection of manufacturing service composition toward Industrial Internet platforms. *IEEE Trans Industr Inform* 2019;15(6):3712–22.
- [41] Xue X, Wang S, Lu B. Manufacturing service composition method based on networked collaboration mode. *J Netw Comput Appl* 2016;59:28–38.
- [42] Zhang S, Xu S, Huang X, Zhang W, Chen M. Networked correlation-aware manufacturing service supply chain optimization using an extended artificial bee colony algorithm. *Appl Soft Comput* 2019;76:121–39.
- [43] Zhu LN, Li PH, Zhou XL. IHDETBO: a novel optimization method of multi-batch subtasks parallel-hybrid execution cloud service composition for cloud manufacturing. *Complexity* 2019;2019:7438710.
- [44] Ding T, Yan G, Lei Y, Xu X. A niching behaviour-based algorithm for multi-level manufacturing service composition optimal-selection. *J Ambient Intell Humaniz Comput* 2020;11(3):1177–89.
- [45] Zhang Y, Xi D, Li R, Sun S. Task-driven manufacturing cloud service proactive discovery and optimal configuration method. *Int J Adv Manuf Technol* 2016;84(1–4):29–45.
- [46] Ma L, Li J, Lin Q, Gong M, Coello Coello CA, Ming Z. Cost-aware robust control of signed networks by using a memetic algorithm. *IEEE Trans Cybern* 2020;50(10):4430–43.
- [47] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw* 2014;69:46–61.
- [48] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 2002;6(2):182–97.
- [49] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Report. Zürich: Eidgenössische Technische Hochschule (ETH) Zürich; 2001 May. Report No.: TIK-Report 103.
- [50] Mirjalili S, Saremi S, Mirjalili SM, Coelho LDS. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Syst Appl* 2016;47:106–19.