PAPER *Special Section on Computational Intelligence and Big Data for Scientific and Technological Resources and Services*

Optimization and Combination of Scientific and Technological Resource Services Based on Multi-Community Collaborative Search

Yida HONG[†], ***Student Member***, **Yanlei YIN**[†a)], **Cheng GUO**[††], ***Nonmembers***, ***and*** **Xiaobao LIU**[†], ***Member***

SUMMARY Many scientific and technological resources (STR) cannot meet the needs of real demand-based industrial services. To address this issue, the characteristics of scientific and technological resource services (STRS) are analyzed, and a method of the optimal combination of demand-based STR based on multi-community collaborative search is then put forward. An optimal combined evaluative system that includes various indexes, namely response time, innovation, composability, and correlation, is developed for multi-services of STR, and a hybrid optimal combined model for STR is constructed. An evaluative algorithm of multi-community collaborative search is used to study the interactions between general communities and model communities, thereby improving the adaptive ability of the algorithm to random dynamic resource services. The average convergence value $C_{MCCSA} = 0.00274$ is obtained by the convergence measurement function, which exceeds other comparison algorithms. The findings of this study indicate that the proposed methods can preferably reach the maximum efficiency of demand-based STR, and new ideas and methods for implementing demand-based real industrial services for STR are provided.

key words: *scientific and technological services, evaluative system, hybrid optimal combined model, multi-community collaborative search, maximum efficiency of combination*

1. Introduction

The General Office of the State Council on Accelerating Development of Scientific and Technological Services clarified an emphasis on the development of STRS including fundamental research, technological transmission, detection certification, entrepreneurship promotion, intellectual property, technology consulting, scientific finance, and technology popularization [1]. However, because of the various types and distributions of STR and service systems, the compositions and interactions in STRS systems and in real economic industries are complicated. Thus, optimal combined STRS must be searched, analyzed, and matched from huge technological resource libraries that are distributed in different locations and industries. The combination of STRS with business processes would form a kind of STRS system [2]. Therefore, the real-time dynamics of real industrial services and the random execution times for STR are more complicated than expected. As the uncertainty arising from STRS makes them difficult to be solved by traditional optimal combination theory and other methods, it is necessary to deliberate the interactions among the response time of services, correlations, and innovations when integrating STRS.

Today, scholars worldwide find references for STRS from libraries [3] and intelligence agencies [4], and aim to expand these areas from the perspectives of literature novelty, literature retrieval, and knowledge services. However, most studies on the combination and optimization of resources focus on cloud computing resources [5], [6], internet resources [7], [8], and cloud manufacturing resources [9], [10]. Furthermore, cloud computing resource services are mainly associated with optimal combined algorithms, such as the particle swarm optimization algorithm [11], [12], fireworks algorithm [13], bee colony algorithm [14], and simulated annealing algorithm [15]. Such methods inadequately consider factors that may influence practical resource services, leading to low efficiency; thus, these methods cannot be used in the demand-based combination of heterogeneous STR. Tao et al. reviewed the methods for the combination of STRS, and pointed out the key issues in cloud manufacturing services. They then put forward relative modelling and evaluative methods [16]. Strunk et al. optimized relative services via semantic matching and genetic algorithms [17]. Ning et al. [18] solved the issues in flow direction and resource combination by a framework of cloud manufacturing. Zeng et al. [19] put forward a service combination based on quality optimization. Tao et al. [20] selected each service resource from cloud manufacturing service platforms to choose resource assemblies that satisfy the functionality of subtasks, and then combined these into resource services according to certain rules, thus allowing for the cooperative access to multi-service tasks. Although many scholars worldwide have conducted research on the combination and optimization of resources, most existing studies have not been technologically oriented and therefore cannot meet the needs for STRS in real industries, which include the sharing and utilization of STR under demand-based environments of distributed STRS.

The combination and optimization of STRS in the present study is an assessment of the performances and demands of services faced by real industries. An integrated mode based on performance assessment is used to provide STR and services during developmental products, thereby creating a reflective transformation from the qualitative demands of STRS to a quantitative combination of resource services. First, a combined framework based on performance assessment for the combination and optimization of resource services is constructed. Then, an optimal combined

Manuscript received November 11, 2020.
Manuscript revised January 30, 2021.
Manuscript publicized May 6, 2021.

[†] The authors are with the Kunming University of Science and Technology, Yunnan, China.
[††] The author is with the Yunnan Electrical Power Research Institute, Kunming, China.
a) E-mail: yinyanlei1990@163.com

DOI: 10.1587/transinf.2020BDP0023

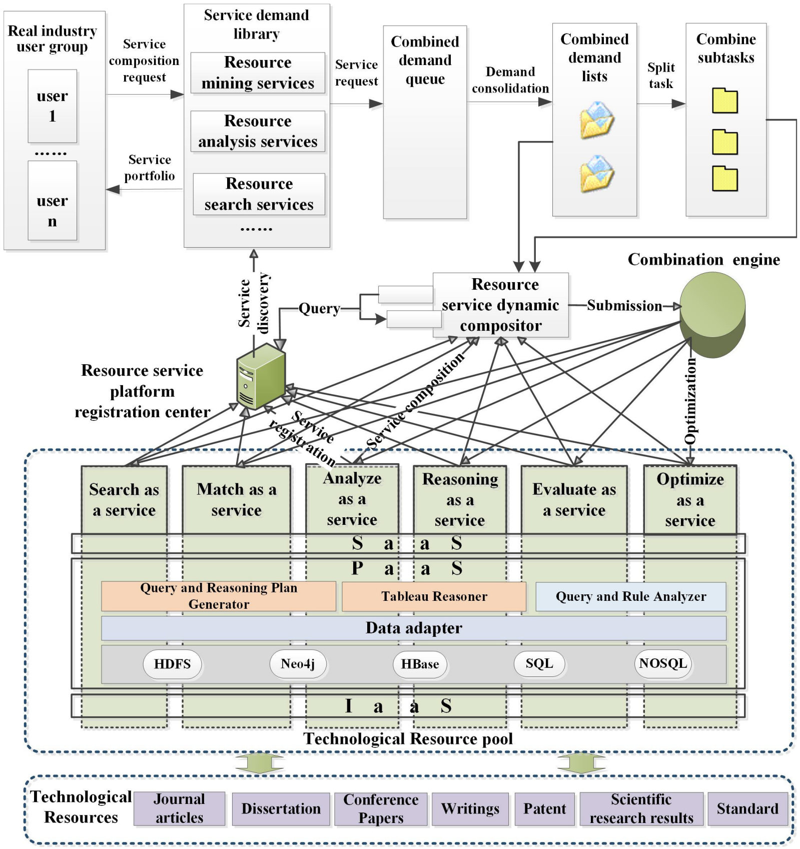


Fig. 1 Combination frameworks for the demands of real industries in STRS.

evaluative system with multi-service tasks is developed, and includes serval indexes including response time, composability, correlation, and innovation. At last, a hybrid optimal combined model of STR is designed. The response ability and performances of resource services are improved by using the strategies of a multi-community bidirectional drive collaborative search algorithm and an asynchronous parallel method. This study provides new methods and techniques for solving the issues of the combination and optimization of multi-service tasks in STRS.

2. Combination Frameworks for the Demands of Real Industries in STRS

In the mode of cloud services, users from real industries submit service tasks to platforms associated with their demands. The platforms analyze the service tasks and then encapsulate different STRS into minimum service units according to the real-time information of STRS and the service abilities of platforms. These units allow for the transfer and combination of STRS for the platforms. The combination frameworks in the present study, which are illustrated in Fig. 1, include the demand side for resources, cloud platforms, and STR. During combination processes, real industrial users will input parameters, and the platforms will automatically produce combined tasks for STRS. The task information will be sent to a resource service composition executor (RSCE), which inquires whether users have registered corresponding combined services. If the inquiry is successful, the combined tasks will be transferred to users; if not, the combined tasks will be divided into subtasks and then sent to the RSCE for executive sequences, and the optimal solutions of STR will be calculated by an intelligent optimization algorithm and returned to the users.

In these frameworks, the resource services can be categorized into a single task, a single task with combinations, or multi-tasks with combinations according to the granularity of users' demands. The single task can be further divided into several single-task units, which is achieved by modelling users' demands under QOS restrictions. The present study focuses on the modelling and optimization of multi-task combinations for STR.

3. Optimal Combined Models of STRS

3.1 The Evaluative Indexes for the Combination and Optimization of STRS

STRS are demand-based modes for the distributed aggregation and sharing of STR in the context of the deep integration of the service industry and real industries. STRS are driven by the demands of services form multi-task collaborative networks, and are unitedly run and controlled by cloud platforms. Therefore, the characteristics of STRS are as follows.

(1) Rapid responses to services. STRS will search, analyze, match, and optimize different dynamic schemes according to different requests from real industrial users. The system will then choose the optimal scheme and return the requests from users via optimal managerial techniques and qualitative evaluation.

(2) Flexible combination of services. STRS are distributed aggregative modes and demand-based sharing modes for the demands of STR. Because of the complicated interactions and compositions among STRS systems, and those between STRS systems and real economic industries, STRS will transfer several processes of resource services. Furthermore, the large scale of intercross, aggregation, and dynamic evolution of multi-language correlations will also be introduced in STRS. Therefore, demand-driven STRS activities have strong flexibility.

(3) Dynamics of service correlations. Real industrial users submit service tasks for their demands to platforms in the STRS environment, and cloud platforms will analyze the service tasks in time. When several tasks are executed interactively, correlative interactions occur between different tasks. To finish one specific task, more related resource services should be considered in multi-mode combinations (series and parallel connections, selection, circulation, etc.).

(4) Innovation of services. The providers, consumers, and operators of STRS are the main components of intelligent collaboration processes. Thus, determining how to satisfy their own demands and how to earn maximum profits from services are the main objectives for all collaborators. The innovation of STRS will have strong effects on the efficiency of resource services.

In summary, the optimization and combination of STRS must not only consider evaluative indexes such as service time, but also other factors such as innovation (In), composability (Cp), and correlation (Ca). Therefore, a combined evaluative system was built as follows.

(1) Service Time. The service time is the time length from receiving the requests to the output of the results, and includes executable service time (T_{pro}) and delay time of service (T_{del}):

$$T = T_{pro} + T_{del} \tag{1}$$

where T_{pro} is the executable service time and T_{del} is the time gap between sending and receiving service requests.

(2) Innovation. Innovation means the originality of services and the novelty of combination. The more innovative a service is, the more value the service has. The function is:

$$In = \frac{1}{\sum\limits_{i=1}^{n} sim(RS_i, RS_j)} \tag{2}$$

$$sim(RS_i, RS_j) = \frac{\omega(a_1 + a_2)}{\sqrt{\sum (RS, RS_j)^2} \cdot \max(|a_1 + a_2|)} \tag{3}$$

where $\omega > 0$, a_1 and a_2 respectively represent the levels of semantic classification trees of RS_i and RS_j getting semantic rules from user feedback, $sim\left(RS_i, RS_j\right)$ represents the degree of similarity of RS_i and RS_j, and $\sqrt{\sum\left(RS_i, RS_j\right)^2}$ indicates the distance between RS_i and RS_j.

(3) Composability. Composability is the probability of STRS being composited during execution, and determines whether the services are transferred as a single function or as a service resource from combined services. The value of composability is the ratio of the specific executing time to total executing time. The function is:

$$Cp = \frac{F_{Cp}}{f_c} \tag{4}$$

where f_c is the time of the specific service being executed, and F_{Cp} indicates the total executing time.

(4) Correlation. Correlation is the degree to which two logic technological resource are correlated. To ensure that two services can coordinate in combined forms, the outputs of former services should match the inputs of later services.

$Ex_i = \{Ex_{i1}, Ex_{i2}, \ldots Ex_{ik}, \ldots, Ex_{im}\}$ is the set of output values of RS_i, and $En_j = \left\{En_{j1}, En_{j2}, \ldots En_{jl}, \ldots, En_{jm}\right\}$ is the set of input values of j. The semantic similarity of Ex_{ik} and En_{jl} is:

$$sim\left(Ex_{ik}, En_{jl}\right) = \frac{\omega(a_1 + a_2)}{\sqrt{\sum\left(Ex_{ik}, En_{jl}\right)^2} \cdot \max(|a_1 + a_2|)} \tag{5}$$

where $\omega > 0$, a_1 and a_2 are the respective levels of semantic classification trees of input property Ex_{ik} and output property En_{jl} obtaining semantic rules from property sets Ex_i and En_j, and $\sqrt{\sum\left(Ex_{ik}, En_{jl}\right)^2}$ indicates the distance between output property Ex_{ik} and input property En_{jl}. Thus, the correlation of service i and service j is:

$$Ca(RS_i, RS_j) = \begin{cases} 1, Ex_i = En_j \\ m\left\{\sum\limits_{k=1}^{m} \max\left[sim\left(Ex_{ik}, En_{jl}\right)\right]^{-1}\right\}^{-1}, Ex_i \subset En_j \\ u\left\{\sum\limits_{k=1}^{m} \max\left[sim\left(Ex_{ik}, En_{jl}\right)\right]^{-\alpha}\right\}^{-1}, Ex_i \cap En_j \neq 0 \\ 0, Ex_i \cap En_j = 0 \end{cases} \tag{6}$$

where $\alpha = \begin{cases} 1, \max\left\{sim\left(Ex_{ik}, En_{jl}\right)\right\} \geq \theta \\ 0, \max\left\{sim\left(Ex_{ik}, En_{jl}\right)\right\} < \theta \end{cases}$, $\theta \in (0, 1)$ is the threshold of definition, and u is the numbers of $\alpha = 1$.

3.2 Hybrid Optimal Combined Model of STRS

The real industrial demand of STR is $RSC = \{RS_1, RS_2, \ldots, RS_m\}$. To satisfy users' needs and maintain good feedback,

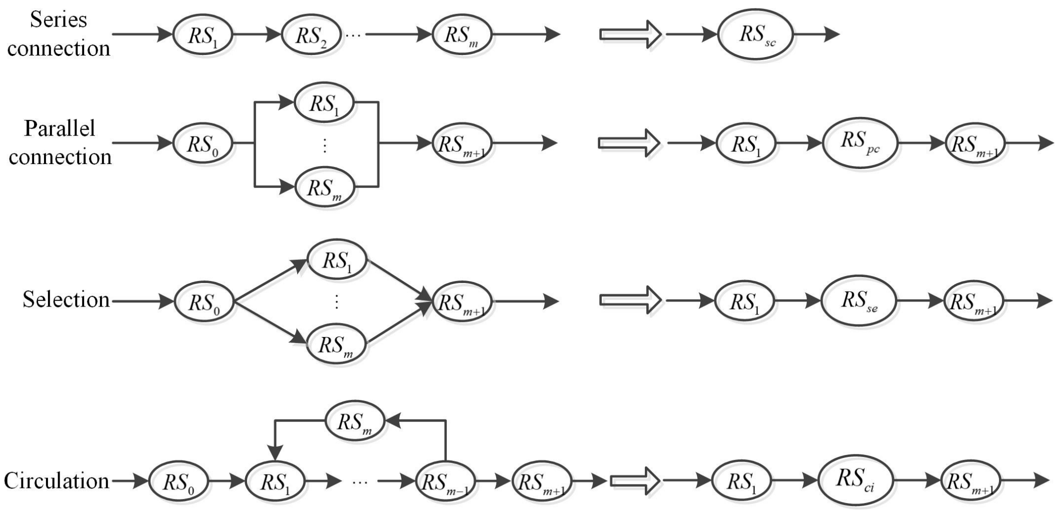


Fig. 2 The four types of combinations in STRS.

Table 1 The model for calculating different indexes under different combinations .

	T	***In***	***Cp***	***Cp***
Series connection	$\sum_{i=1}^{m} T\left(RS_i^j\right)$	$\sum_{i=1}^{m} \frac{In\left(RS_i^j\right)}{m}$	$\prod_{i=1}^{m} Cp\left(RS_i^j\right)$	$\frac{\sum_{i=0}^{m} \frac{\sum_{t=t+1}^{m+1} Ca\left(RS_i, RS_t\right)}{m-i}}{m}$
Parallel connection	$\max T\left(RS_i^j\right)$	$\sum_{i=1}^{m} \frac{In\left(RS_i^j\right)}{m}$	$\min Cp\left(RS_i^j\right)$	$\min_{i=1}^{m} \left\{ \frac{Ca(RS_0,RS_i)+Ca(RS_0,RS_{m+1})}{G(RS_0,RS_i)+G(RS_0,RS_{m+1})} \right\}$
Selection	$\sum_{i=1}^{m} \left(T\left(RS_i^j\right) \times p^j\right)$	$\sum_{i=1}^{m} \left(In\left(RS_i^j\right) \times p^j\right)$	$\sum_{i=1}^{m} \left(Cp\left(RS_i^j\right) \times p^j\right)$	$\sum_{i=1}^{m} p_i \times \left\{ \begin{array}{c} \frac{Ca(RS_0,RS_i)+Ca(RS_0,RS_{m+1})}{G(RS_0,RS_i)+G(RS_0,RS_{m+1})} \\ +Ca\left(RS_i, RS_{m+1}\right) \end{array} \right\}$
Circulation	$r \times \sum_{i=1}^{m} T\left(RS_i^j\right)$	$\sum_{i=1}^{m} \frac{In\left(RS_i^j\right)}{m}$	$\min Cp\left(RS_i^j\right)$	$\frac{\sum_{i=0}^{m} \frac{\sum_{t=t+1}^{m+1} Ca\left(RS_i, RS_t\right)}{m-i}}{m}$

an optimal model was developed by the evaluative system discussed previously.

$$
\begin{aligned}
Q(RSC) = \{&T(RSC), In(RSC),\\
&Cp(RSC), Ca(RSC)\}
\end{aligned}
\tag{7}
$$

In this paper, the main principles are borrowed from web services composition, and the combination of STRS is divided into series connection, parallel connection, selection, and circulation. In real services, however, one technological resource service can have several combinations. Therefore, the hybrid combinations can be transferred into combinations of several series and parallel connections.

The mathematic model for calculating different combinations to arbitrate STRS RS_i^j is presented in Table 1.

In this table, m indicates that the service combination is combined by m numbers of resource services RS, r is the number of cycles of the resource service candidate set RS_i corresponding to the i-th resource service, p^j indicates the probability of RS_i^j being selected in the candidate set RS_i of resource services, and $\sum_{j=1}^{n} p^j = 1$, where n is the number of candidate resources in RS_i^j. Thus, the formula of $T_{ask} = \{t_1, t_1, \ldots, t_k, \ldots, t_M\}$ for STRS is:

$$
\left\{
\begin{array}{l}
T(RSC) = \sum_{i=1}^{M} T\left(RS_i^j\right) \\
In(RSC) = \sum_{i=1}^{M} In\left(RS_i^j\right)\Big/ M \\
Cp(RSC) = \prod_{i=1}^{M} Cp\left(RS_i^j\right) \\
Ca(RSC) = \sum_{i=0}^{M} \left(\sum_{t=t+1}^{M+1} Ca\left(RS_i, RS_t\right) \Big/ M - i \right) \Big/ M
\end{array}
\right.
\tag{8}
$$

The dimensions of the evaluative indexes from the combinations of STRS are different and cannot be directly used for calculation; thus, the dimensions should be simply normalized. For those “the more the better” beneficial indexes In, Cp, and Ca,

$$
y_i = \left\{
\begin{array}{l}
\frac{u_i(In,Cp,Ca)-\min u_i(In,Cp,Ca)}{\max u_i(In,Cp,Ca)-\min u_i(In,Cp,Ca)},\ \max u_i \neq \min u_i \\
1,\ \max u_i = \min u_i
\end{array}
\right.
\tag{9}
$$

and for the “the smaller the better” index T,

$$
y_i = \left\{
\begin{array}{l}
\frac{\max v_i(T)-v_i(T)}{\max v_i(T)-\min u_i(T)},\ \max u_i \neq \min u_i \\
1,\ \max u_i = \min u_i
\end{array}
\right.
\tag{10}
$$

where u_i indicates the service evaluative indexes In,

Cp, and Ca, v_i is the service evaluative index T, and y_i is the normalized service value.

Therefore, the combinations that consist of m resource services have $\prod_{i=1}^{m} n$ combinations, as each service has n candidate resources.The evaluative index system of combined services, which is composed of the response time, innovation, composability, and correlations, can construct an optimal combined plan for STRS. Because users may have different requests for the same evaluative index under different service circumstances, the optimization of multi-tasks can be transformed into the optimization of a single task, and weighting indexes can then be used to reflect the user's request under specific circumstances. The optimization and combination of the targeted formula are as follows:

$$\max Q(RSC) = \omega_T \frac{1}{y^T(RSC)} + \omega_{In} y^{In}(RSC) + \omega_{Cp} y^{Cp}(RSC) + \omega_{Ca} y^{Ca}(RSC) \tag{11}$$

$$s.t. \begin{cases} \omega_T + \omega_{In} + \omega_{Cp} + \omega_{Ca} = 1; \\ u(RSC) \geq u(T_{ask}), u = In, Cp, Ca; \\ v(RSC) \leq v(T_{ask}), v = T. \end{cases} \tag{12}$$

where ω_T, ω_{In}, ω_{Cp}, and ω_{Ca} are weighting indexes, $\omega \in (0, 1)$, and T_{ask} is the request for services.

3.3 Optimal Combined Algorithms for STRS Based on Multi-Community Collaborative Search

The particle swarm optimization algorithm is an algorithm with which to find optimal answers via cooperation and iterative operation among particles. Many studies have suggested that this algorithm can be used in the optimization of multi-tasks. However, basic particle swarm algorithms, which core on optimal particles, are the optimal modes for a single community. Thus, they cannot be used to solve problems with dynamic data. To expand this mode into multi-community levels, the multi-community collaborative search algorithm (MCCSA), which is highly adaptive to tasks, was created. In this study, MCCSA was used to solve the problems of combination in STRS. Considering the differences between real problems in combination and the function of general multi-tasks in STRS, the operational algorithms are defined and the discrete iterations of the speed and positions of particles are completed, thereby achieving service combinations from particle searching systems.

(1) Co-evolution algorithms Rule 1: Iteration in communities. During the co-evolution of multi-communities, particles in a single community are iterated according to the speed and position of basic particle swarms, thereby optimizing the values in communities:

$$\begin{cases} v_{id}^{t+1} = \omega \cdot v_{id}^{t} + c_1 \cdot r_1 \cdot (P_{id}^{t} - x_{id}^{t}) + \\ c_2 \cdot r_2 \cdot (P_{gd}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \\ \quad i = 1, 2, \ldots, m \quad d = 1, 2, \ldots, D \end{cases} \tag{13}$$

where t is the time of the iterations during the particle searching processes, ω is the inertia weight, $c_1 = c_2 = 2$ indicates the accelerating constants, and r_1 and r_2 are random functions in the interval of $[0, 1]$.

Rule 2: Two-way co-evolution rules among communities

Rule 2.1: General communities are denoted as g_{best}, model communities are denoted as G_{best}, model learning factors are denoted as $P_{\mathrm{nd}} = \frac{\sum_{i=1}^{n} G_{besti}}{n}$, CC represents the particles in general communities, MC indicates the particles in model communities, $\forall(r_1 : \langle CC_i, MC_j \rangle) \in R$, $g_{besti} = \max\{g_{best1}, g_{best2}, \ldots, g_{bestm}\}$, $G_{bestj} = \min\{G_{best1}, G_{best2}, \ldots, G_{bestn}\}$, and $g_{besti} \geq G_{bestj}$. If particles CC_i go into CC_i, the last community MC_i is eliminated. Thus, the new iterative algorithm is:

$$\begin{cases} v_{id}^{t+1} = \omega \cdot v_{id}^{t} + c_1 \cdot r_1 \cdot \left(P_{id}^{t} - x_{\mathrm{id}}^{\mathrm{t}}\right) + \\ c_2 \cdot r_2 \left(P_{gd}^{t} - x_{id}^{t}\right) + c_3 (P_{nd}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \\ \quad i = 1, 2, \cdots, m \quad d = 1, 2, \cdots, D \end{cases} \tag{14}$$

where c_3 is the random function and satisfies the algorithm convergence $c_1 r_1 + c_2 r_2 + c_3 \in [0, 4]$.

Rule 2.2: $\forall(r_2 : \langle MC_i, MC_j \rangle) \in R$, where the collaborative intensity in a model community is $s_{MC_{\mathrm{i}}}$. To arbitrate $s_{MC_{\mathrm{j}}}$, $s_{MC_{\mathrm{i}}} \geq s_{MC_{\mathrm{j}}}$. The optimal value in a model community is $PG = G_{besti}$.

Rule 2.3: $\forall(r_3 : \langle CC_i, CC_j \rangle) \in R$, where the collaborative intensity in a general community is $s_{CC_{\mathrm{i}}}$. To arbitrate $s_{CC_{\mathrm{j}}}$, $s_{CC_{\mathrm{i}}} \geq s_{CC_{\mathrm{j}}}$. The optimal value in a general community is $P\mathrm{g} = g_{besti}$.

(2) Strategies for discretions and iterations. To solve the issues of the combination of multi-services in discretions, the operators in the algorithms were re-defined. A matrix X has n rows and n columns; $n \times n$ is the vector matrix of particle positions. $X_i =< x_{i1}, x_{i2}, \ldots, x_{in} >$ indicates the position of the i-th particle with specific combined services. x_{ij} $(j = 1, 2, \ldots, n)$ is a positive integer, indicating the numbers of candidate services in T_j selections. $V : n \times n$ is then defined as the vector matrix of particle speeds. $V_i =< V_{i1}, V_{i2}, \ldots, V_{in} >$ is the speed of the i-th particle, and v_{ij} $(j = 1, 2, \ldots, n)$ is the integrative efficiency of a specific combined service. Based on the discretions and iterations of the particles speeds and positions, the algorithms (13) and (14) are updated to:

$$\begin{cases} v_{id}^{t+1} = \omega \cdot v_{id}^{t} + c_1 \cdot r_1 \cdot (P_{id}^{t} \Theta x_{id}^{t}) + \\ c_2 \cdot r_2 \cdot (P_{gd}^{t} \Theta x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} \oplus v_{id}^{t+1} \\ \quad i = 1, 2, \ldots, m \quad d = 1, 2, \ldots, D \end{cases} \tag{15}$$

$$\begin{cases} v_{id}^{t+1} = \omega \cdot v_{id}^{t} + c_1 \cdot r_1 \cdot \left(P_{id}^{t} \Theta x_{\mathrm{id}}^{\mathrm{t}}\right) + \\ c_2 \cdot r_2 \left(P_{gd}^{t} \Theta x_{id}^{t}\right) + c_3 (P_{nd}^{t} \Theta x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} \oplus v_{id}^{t+1} \\ i = 1, 2, \cdots, m \quad d = 1, 2, \cdots, D \end{cases} \tag{16}$$

where operator Θ is the difference between the efficiency of two particles in their dimensions, and operator $\oplus$ is the selection of the new position of an arbitrary particle in different dimensions; $x_{ki} \oplus v_{ki} = \left\{ j \middle| \min_{j=1,2,\ldots,m} \left| f(Q_{IJ}(RSC)) - (V_{KI} + f(Q_{ix_{ki}}(RSC))) \right| \right\}$.

(3) Algorithm steps. Based on the two-way co-evolutionary rules and discrete iterative strategies, the steps of the optimal algorithms for combined multi-services are as follows.

Step 1: Initialize the particles. Set up the number of communities, the times of iterations, accelerating indexes, and inertia weight coefficient.

Step 2: Assign the initialized particles averagely into *1* process, contributing to a community with a size of int(n/l). Then calculate the adaptive values in *1* communities according to Eq. (11).

Step 3: Put the communities into *1* processes and conduct asynchronous parallel evolutive calculation.

Step 4: Calculate the adaptive values *Fi* in different communities, and then divide the communities into model communities and general communities according to the thresholds.

Step 5: Update the speeds and positions of particles in the communities by Eq. (16) according to the mechanisms of discretion and iteration, and then save the optimal positions of the particles.

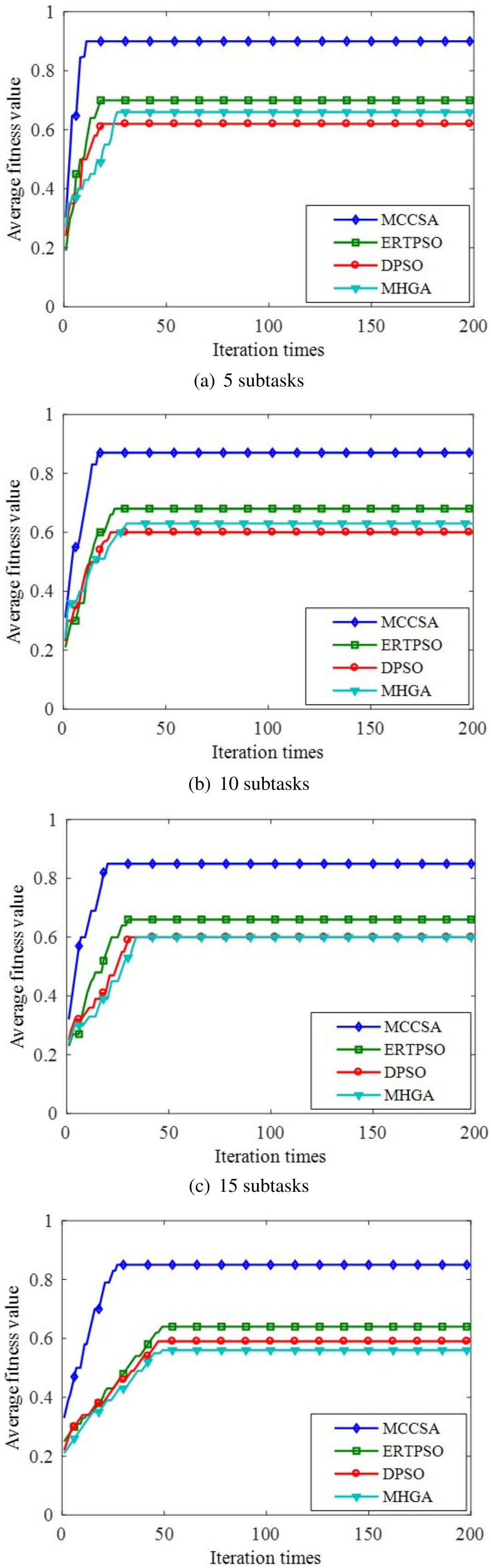
Step 6: The algorithms will finish and output optimal combinations if all the particles can satisfy the ending requirements; otherwise, return to step 5.

4. Simulation Experiments and Results

4.1 Experimental Conditions

The proposed model and algorithms of STRS were verified based on the analysis of the issues of one manufacturing enterprise. JDK 1.7 and Eclipse software were used as the integrated developmental environment, Tomacat 7.0 was used as the server, MySQL5.0 was used as the dataset, Hadoop was used as the big-data platform software, and VMware10 was used as the virtual instruments. The staple software mpiBLAS was introduced as service-transferring instructions to build up cloud platforms for STR. The simulated environments of the optimal algorithms were based on MATLAB (2016b) and a computer with a 4.00 GHz processor, 16 GB RAM, and the Windows 7 operating system. The input data for simulated experiments included the numbers of users' requests per second, which were randomly selected from 10-100. The types of STRS were randomly selected from Rs1, Rs2, ... , Rs6, and included several services such as searching, analyzing, and matching of resources. The requests of each RSC were randomly selected from 1-20. The combined evaluative indexes were the response time, innovation, composability, and correlation.

Embedded rhombus thought particle swarm optimization (ERTPSO), the modified hybrid genetic algorithm (MHGA), discrete particle swarm optimization (DPSO), and



(a) 5 subtasks

(b) 10 subtasks

(c) 15 subtasks

(d) 20 subtasks

Fig. 3 The convergence performance of different algorithms for different numbers of subtasks.

Table 2 The comparison of convergence accuracy of four algorithms .

Fun	PSize	Algorithms	Best	Worst	Mean	SD	AD
Q	50	ERPSO	8.54e-001	1.82e-001	6.03e-001	2.41e-001	1.93e-001
		DPSO	8.17e-001	2.04e-001	5.60e-001	3.26e-001	2.65e-001
		MHGA	7.64e-001	1.84e-001	5.34e-001	3.49e-001	2.86e-001
		MCCSA	9.12e-001	4.47e-001	8.17e-001	1.13e-001	9.11e-002
	100	ERPSO	9.16e-001	1.67e-001	6.58e-001	3.22e-001	2.60e-001
		DPSO	8.32e-001	1.91e-001	6.02e-001	3.33e-001	2.65e-001
		MHGA	8.71e-001	1.63e-001	5.93e-001	3.89e-001	3.18e-001
		MCCSA	9.36e-001	4.33e-001	8.56e-001	1.32e-001	1.07e-001
	400	ERPSO	9.23e-001	1.24e-001	7.22e-001	4.55e-001	3.62e-001
		DPSO	8.41e-001	1.54e-001	6.41e-001	3.57e-001	2.82e-001
		MHGA	8.73e-001	1.40e-001	6.59e-001	5.00e-001	4.13e-001
		MCCSA	1	3.26e-001	9.02e-001	1.90e-001	1.51e-001

MCCSA were used to solve the issues of the combination and optimization of STRS. The initialized parameters of the algorithms were set as follows: the number of populations was 100, the maximum evolutionary time was 200, the inertia weight was $w = 1.1$, the accelerating constant was $c_1 = c_2 = 2$, and the weighting factors of the four index properties were ω_T=0.3, ω_{In}=0.3, ω_{Cp}=0.2, and ω_{Ca}=0.2, respectively. All the simulated experiments were conducted 500 times each, and the results were averaged. All other parameters were set according to existing references.

4.2 Results and Discussion

To test the convergence performance of MCCSA, the four algorithms were used to study this problem simultaneously. During the experiments, the number of candidate services at each subtask was 120. Figure 3 presents results of algorithms on different numbers of subtasks, and shows the changes of MCCSA for different numbers of subtasks in the combination of STRS. The selection and reorganization of co-evolutionary rules enhanced the characteristics of superior searches in different communities, thereby leading to increases of adaptive ability and executive efficiency. With the increasing number of subtasks, MCCSA rapidly jump out local optimal points with a relatively fast convergence speed and consistent and effective regional optimal points.

The following convergence measurement function is used to test its convergence characteristics.

$$C = \frac{1}{E}\sum_{e=1}^{E}[f(e+1) - f(e)] \tag{17}$$

Let $E = 199$, the average convergence value of each algorithm is obtained as follows, $C_{MCCSA} = 0.00274$, $C_{ERTPSO} = 0.00214$, $C_{DPSO} = 0.00186$, $C_{MHGA} = 0.00188$. From the results of experiments, it can be concluded that the performance of MCCSA was significantly better than that of the other comparison algorithms.

A comparative analysis of the optimal functions of different population sizes by the four algorithms was conducted. All the vectors of algorithms were 30 and all the evolutive algebras were 120. The population sizes were 50, 100, and 400. MCCSA was selected for the asymmetric initialization of spaces. The results are presented Table 2, and demonstrate that MCCSA exhibited strongly adaptive abilities, a rapid convergence speed, and relatively high precision. Moreover, the zero values in the standard deviation (SD) and average deviation (AD) indicate that MCCSA had stable convergence ability. The data suggest that MCCSA can automatically adjust the searching ability of algorithms according to the cooperative rules in many populations. Thus, MCCSA can adapt to the changes in multi-service combinations. These results suggest that MCCSA presents significant advantages in terms of convergence precision, convergence stability, and convergence speed.

5. Conclusions

To address the issues of combination in technological series in the context of deeply aggregated real industries and service industries, a demand-based frame of STRS in real industries was developed, and the characteristics of STRS and their issues of combination were analyzed. In this way, an optimal evaluative system for multi-services was constructed, and a hybrid optimal combined model that includes several combinations of STRS was designed. A two-way collaborative searching algorithm for multi-populations was then used to solve this model. Finally, the results of the simulated experiments verified the effectiveness of model and algorithms, and suggest that the proposed methods can effectively solve the issues of the optimization and combination of multi-task STRS. Future studies should expand the proposed model and algorithms for combinations of STRS, thereby increasing their adaptivity for regional industrial clusters.

Acknowledgments

This paper was support by the National Key R & D Program of China (2017YFB1400301).

References

[1] Central government portal, "Several opinions on accelerating the development of science and technology service industry," http://www.gov.cn/xinwen/2014-10/28/content_ 2771609.html, accessed Aug. 25. 2019.

[2] C. Perera and A.V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," Knowledge-Based Systems, vol.109,

pp.122–136, Oct. 2016.

[3] J. Feng and N. Zhao, "A New Role of Chinese Academic Librarians—The Development of Embedded Patent Information Services at Nanjing Technology University Library," J. Academic Librarianship, vol.41, no.3, pp.292–300, May 2015.

[4] G. Dai, "Promoting innovative intelligence research in race stage," J. Intelligence, vol.38, no.8, pp.771–777, 2019. Doi: 10.3772/j.issn.1000-0135.2019.08.001.

[5] J. Guo, C. Li, Y. Chen, and Y. Luo, "On-demand resource provision based on load estimation and service expenditure in edge cloud environment," J. Network and Computer Applications, vol.38, no.8, pp.771–777, 2019.

[6] D. Liu, X. Sui, L. Li, Z. Jiang, H. Wang, Z. Zhang, and Y. Zeng, "A cloud service adaptive framework based on reliable resource allocation," Future Generation Computer Systems, vol.89, pp.455–463, Dec. 2018.

[7] X.-F. Xu, J. Hao, Y.-R. Deng, and Y. Wang, "Design optimization of resource combination for collaborative logistics network under uncertainty," Applied Soft Computing, vol.56, pp.684–691, July 2017.

[8] S. Baradaran, S.M.T. Fatemi Ghomi, M. Ranjbar, and S.S. Hashemin, "Multi-mode renewable resource-constrained allocation in PERT networks," Applied Soft Computing, vol.12, no.1, pp.82–90, Jan. 2012.

[9] J. Chen, G.Q. Huang, J.-Q. Wang, and C. Yang, "A cooperative approach to service booking and scheduling in cloud manufacturing," European J. Operational Research, vol.273, no.3, pp.861–873, March 2019.

[10] Y. Lu and X. Xu, "A semantic web-based framework for service composition in a cloud manufacturing environment," J. Manufacturing Systems, vol.42, pp.69–81, Jan. 2017.

[11] J. Liao, Y. Liu, and X. Zhu, and J. Wang, "Accurate sub-swarms particle swarm optimization algorithm for service composition," J. Systems and Software, vol.90, pp.191–203, April 2014.

[12] Y. Xie, Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A.S. Sani, D. Yuan, and Y. Yang, "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment," Future Generation Computer Systems, vol.97, pp.361–378, Aug. 2019.

[13] Y. Chen, L. Li, X. Zhao, J. Xiao, Q. Wu, and Y. Tan, "Simplified hybrid fireworks algorithm," Knowledge-Based Systems, vol.173, pp.128–139, June 2019.

[14] J. Zhou and X. Yao, "Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing," Applied Soft Computing, vol.56, pp.379–397, July 2017.

[15] P. Lv, L. Yuan, and J. Zhang, "Cloud theory-based simulated annealing algorithm and application," Engineering Applications of Artificial Intelligence, vol.22, no.4-5, pp.742–749, June 2009.

[16] F. Tao, L. Zhang, H. Gua, Y.-L. Luo, and L. Ren, "Typical characteristics of cloud manufacturing and several key issues of cloud service composition," Computer Integrate Manufacturing Systems, Fundamentals, vol.17, no.4, pp.477–486, 2011.

[17] A. Strunk, "QoS-aware service composition: a survey," IEEE. Proc. 8th IEEE European Conference Web Services, Washington, D.C., USA, pp.67–74, Dec. 2010.

[18] F. Ning, W. Zhou, F. Zhang, Q. Yin, and X. Ni, "The architecture of cloud manufacturing and its key technologies research," IEEE Computer Society, Proceeding 2011 IEEE Int. Conf. Cloud Computing and Intelligence Systems, Washington, D.C., USA, pp.259–263, Sept. 2011.

[19] L. Zeng, B. Bentallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," IEEE Trans. Softw. Eng., vol.30, no.5, pp.311–327, May 2004.

[20] F. Tao, D. Zhao, and Y. Hu, "Correlation-aware resource service composition and optimal-selection in manufacturing grid," European J. Operational Research, vol.201, no.1, pp.129–143, Feb. 2010.

Yida Hong received a bachelor's degree in engineering from Jiangxi University of Science and Technology in 2018, and has been studying for a master's degree in Kunming University of technology since September 2018, his current research interests include machine learning, knowledge engineering.

Yanlei Yin received his master's degree in engineering from Kunming University of technology in 2017. He is currently a doctorate student at Kunming University of Science and Technology, China. His research interests include evolutionary computation, deep learning technology, knowledge engineering.

Cheng Guo is currently a professor senior engineer in Yunnan Electrical Power Research Institute, Kunming. China. His current research interests include artificial intelligence, analysis and test technology.

Xiaobao Liu received his doctor's degree in mechanical and electronic engineering of University of Electronic Science and technology in 2011. From July 2011, he worked in the school of mechanical and electrical engineering of Kunming University of technology, mainly engaged in the research of artificial intelligence in manufacturing process.