Postprint: Research on Optimization of New Product Development Teams in Cloud Environments

Authors: Youling Chen, Wang Long, Zuo Lidan, Niu Yufei

Date: 2018-11-29T00:00:00+00:00

Abstract

Addressing the problem of optimal team selection for new product development in cloud environments, and fully considering teams' research and development capabilities, coordination capabilities, and service quality, a comprehensive evaluation model incorporating a knowledge similarity evaluation model, a synergy effect evaluation model, and a service quality evaluation model is established; the fitness function and search strategy of the algorithm are improved, and an enhanced artificial bee colony algorithm is proposed to solve the established model and select the optimal team combination for service demanders; finally, taking the new product development of Automated Guided Vehicles (AGV) as an example, the feasibility and effectiveness of the proposed method are verified through model solving and algorithmic comparison.

Full Text

Preamble

Research on Team Optimal Selection Based on Cloud Environment for New Product Development

Chen Youling, Wang Long†, Zuo Lidan, Niu Yufei
(The State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, China)

Abstract: For the new product development team optimal selection problems in cloud environment, this paper fully considers the team' s research and development capabilities, coordination capabilities, and quality of service, and establishes a comprehensive evaluation model that includes a knowledge similarity evaluation model, a collaborative effect evaluation model, and a quality of service evaluation model. The fitness function and search mechanism are improved, and a modified artificial bee colony algorithm is proposed to solve the established model and find an optimal team combination for the service demanding party. Finally, taking the new product development of an automatic guided vehicle (AGV) as an example, the feasibility and effectiveness of the proposed method are verified through model solution and algorithm comparison.

Key words: team optimal selection; knowledge similarity; collaborative effect; quality of service; artificial bee colony algorithm

0 Introduction

With the rapid development of cloud computing and big data technologies, the resulting cloud platforms have been widely applied. Cloud platforms utilize internet information technology to achieve resource virtualization and sharing, making full use of resources and reducing waste caused by idleness, thereby achieving a state of "concentrated use of dispersed resources and distributed service of concentrated resources" [1]. As an important component of cloud platform resources, human resources enable cross-regional capability sharing and collaboration, which forms the foundation for collaboratively completing new product development tasks in cloud environments.

Taking human resources as the object, cloud platforms select members across regions, disciplines, and industries to form temporary teams that complete design tasks through mutual collaboration and knowledge sharing [2]. However, for the design and development of complex tasks, member selection alone has become overly complicated and cannot meet task requirements for human resources. Using teams as the selection unit for team combination represents a breakthrough point.

Cloud platform operators possess massive amounts of virtualized human resource information, including numerous service teams with similar or identical services but different service quality and capabilities [3]. The service demanding party needs human resources, while the resource providing party supplies exactly that—human resources. From submitting a new product development task request to the cloud platform to obtaining task execution results, four stages must be experienced: task decomposition, subtask analysis and search matching, team combination optimization, and execution and feedback. In the task decomposition stage, the new product development task is submitted to the cloud platform, which decomposes the new product development task into N sub-development tasks based on relevant information provided by the service demanding party and the complexity of new product development, where each subtask can be completed by a single service team. In the subtask analysis and search matching stage, the cloud platform analyzes the requirements and characteristics of each subtask and searches for matching candidate team sets that satisfy the subtask. In the team combination optimization stage, it is necessary to select one team from each subtask' s candidate team set to execute the overall development task, resulting in a team optimization combination. This combination is the team combination execution path to be obtained in the team combination optimization stage. In the execution and feedback stage, after confirmation by the service demanding party, the team combination is called upon to execute the new product development task, and the entire development process is monitored and fed back by the cloud platform.

This paper focuses on effective solutions for the team combination optimization stage. The cloud platform selects one team from each candidate team set through multi-party consideration and global optimization, then sequentially calls the selected teams to construct and merge a team combination execution path.

Existing literature has addressed human resource allocation by establishing optimization indicator systems and selection methods. Chen et al. [4] constructed a member selection model in stages based on member knowledge similarity and learning ability, and conducted member assignment. Jiang et al. [5] constructed a cross-functional team member selection optimization model based on member comprehensive quality and professional skill index expectations to complete team member selection. Wang et al. [6] combined analytic hierarchy process and fuzzy preference programming to solve virtual team member optimization problems. Zhang et al. [7] considered member comprehensive capabilities and interpersonal relationships, established an interpersonal relationship model, and used a multi-objective particle swarm algorithm for solution. Liu et al. [2] built an optimization indicator system and multi-objective optimization model based on member individual capabilities and collaborative cooperation capabilities, using the SPEA2 algorithm for solution. Feng et al. [8] established a mathematical model for team partner selection based on partner coordination relationships and collaborative effects, solved by a GRASP heuristic algorithm. Feng et al. [9] established a team member optimization model based on individual and collaborative performances, using an improved non-dominated sorting genetic algorithm for solution.

The above literature primarily studied human resource combination optimization models and solution methods, achieving numerous research results that largely solved human resource combination optimization problems and promoted the selection of optimal human resource combination schemes. However, literature [4-7] selected team members by constructing member evaluation systems without considering collaborative effects among members, which may lead to suboptimal actual selection results. Literature [8,9] considered both team member information and collaborative relationships among members for virtual team formation, but individual member selection could not meet task requirements when handling complex tasks. Additionally, the above literature employed multi-objective particle swarm optimization algorithms, SPEA2 algorithms, GRASP heuristic algorithms, and non-dominated sorting genetic algorithms, but these still suffer from unstable convergence speed, poor convergence effects, and tendency to converge to local optimal solutions during actual operation.

Based on the above analysis, this paper takes teams as the selection object, analyzes the characteristics of team combination in cloud environments, fully considers team R&D capabilities, coordination capabilities, and service quality, and establishes a comprehensive evaluation model including knowledge similarity evaluation model, collaborative effect evaluation model, and service quality evaluation model. By improving the artificial bee colony algorithm to enhance convergence speed and effectiveness, this paper solves for the optimal team combination for service demanding parties.

1 Team Combination Optimization and Knowledge Transfer Process Description

1.1 Cloud Platform Team Combination and Optimization Process Description

The team selection process on the cloud platform is shown in Figure 1 [Figure 1: see original paper]. Three main parties exist on the cloud platform: service demanding party, cloud platform operator, and resource providing party. For the new product development team optimization process on the cloud platform, the service demanding party is the publisher of new product development tasks, while the resource providing party consists of various design and R&D teams.

Task ,SubtaskSTiST 12„„NiTasSTSTSTkST iSTiST 12=„„jMiiiiiCTSCTCTCTCTiSTiCTSTask 1212„„iNj

1.2 Parallel New Product Development and Team Knowledge Transfer Process Description

Through literature review, research has been conducted on the execution sequence of sub-development tasks in the new product development process. Based on the chronological order of subtask execution, design activities in the development process are divided into three basic modes: serial mode, overlapping mode, and parallel mode [10]. In serial mode, task B begins only after task A is fully completed; in overlapping mode, task B begins 提前执行任务 based on pre-released information from task A before task A is completed; in parallel mode, tasks A and B begin simultaneously with information exchange and transfer. These three modes differ in task completion cycle and cost. Serial mode has long product development cycles, few information exchanges, low development costs, and high rework possibility; overlapping mode has moderate product development cycles, moderate information exchange between related tasks, moderate development costs and rework possibility; parallel mode has short product development time, many information exchanges, high development costs, and reduced rework probability with increased communication frequency. These development modes indirectly reflect the trade-off between development time and cost. This paper studies parallel mode new product development. The new product development modes are shown in Figure 2 [Figure 2: see original paper].

New product development is essentially a process of knowledge creation, knowledge accumulation, and knowledge transfer [11], with product development teams being the providers of knowledge. The knowledge transfer process is completed through information exchange between teams. Taking two development tasks in parallel as an example, the knowledge transfer process is shown in Figure 3 [Figure 3: see original paper]. Teams a and b complete tasks A and B respectively. During the task development cycle, teams a and b conduct n knowledge transfers. Before the first knowledge transfer between teams a and b, they perform knowledge operations based on their original accumulated knowledge; after the first knowledge transfer, teams a and b perform knowledge operations based on the acquired initial knowledge; after the second knowledge transfer, due to obtaining erroneous or incomplete knowledge in the previous transfer, they need to update erroneous knowledge or complete knowledge in this transfer process, thereby performing knowledge rework [12], and teams a and b perform new knowledge operations based on updated knowledge; this continues until task completion is achieved.

The time for sub-development tasks mainly consists of knowledge operation time, knowledge rework time, and communication time. Knowledge operation time is the time consumed for knowledge operations before and after knowledge transfer, primarily determined by the task itself and unaffected by external factors, denoted by e_{iT} as the total knowledge operation time for subtask i; knowledge rework time is the time consumed for knowledge rework, denoted by $r_{ij}T$ as the knowledge rework time generated by subtask i in the k-th knowledge transfer between the teams responsible for subtasks i and j; communication time is the time consumed for knowledge transfer between two teams, denoted by $c_{ij}T$ as the communication time in the k-th knowledge transfer between the teams responsible for subtasks i and j. Additionally, team R&D capabilities, learning capabilities, communication capabilities, and collaborative capabilities are all major factors affecting the number of knowledge transfers and the time spent on operations and rework.

---

2 Model Construction

To comprehensively evaluate team optimization combinations, this paper constructs knowledge similarity evaluation model, collaborative effect evaluation model, and service quality evaluation model from three aspects: R&D capability, collaborative effect, and service quality. The advantages of the three models are effectively combined to construct a comprehensive evaluation model that provides a basis for service demanding parties to select appropriate team combinations. The model structure schematic diagram is shown in Figure 4

[Figure 4: see original paper].

2.1 Knowledge Similarity Evaluation Model

To better measure the R&D capability of development teams for new product development tasks, this paper proposes knowledge similarity (KS), which refers to the similarity degree between a development team' s knowledge structure and the knowledge requirements of the development task. Greater knowledge similarity indicates that the development team is more suitable for the development task and has stronger development capability for it, and vice versa.

This paper references relevant concepts from literature [13] to calculate knowledge similarity. The following concepts are defined: Knowledge point (KP) is the smallest data unit in new product development subtasks that cannot be further divided. The knowledge set of a development task contains multiple related knowledge points, and a knowledge structure tree is drawn based on these points. Knowledge module (KM) is a collection of the aforementioned knowledge points. The knowledge requirement module of a new product development task can be expressed as $\langle MATH_1 \rangle$, where: $\langle MATH_2 \rangle$ represents the knowledge points required for the new product development task; $\langle MATH_3 \rangle$ represents the demand degree of each knowledge point. The knowledge mastery module of each candidate team can be expressed as $\langle MATH_4 \rangle$, where $\langle MATH_5 \rangle$ represents the knowledge points mastered by the team; $\langle MATH_6 \rangle$ represents the team' s mastery degree of each knowledge point.

Knowledge similarity includes semantic similarity and distance similarity [4]. Semantic similarity is determined by the positions of related knowledge points in the knowledge structure tree, calculated by formula (1). Distance similarity is determined by the Euclidean distance between the two vector spaces $\langle MATH_7 \rangle$ and $\langle MATH_8 \rangle$, calculated by formula (2). To better describe knowledge similarity, semantic similarity and distance similarity are normalized to the interval (0,1], with results shown in formulas (3) and (4).

Where: $\langle MATH_9 \rangle$ is the relative degree between parent and child classes in the knowledge structure tree; $\langle MATH_{10} \rangle$ is the number of subclass levels belonging to both $\langle MATH_{11} \rangle$ and $\langle MATH_{12} \rangle$ under the tree; $\langle MATH_{13} \rangle$ is the number of subclass levels belonging to $\langle MATH_{14} \rangle$ under the tree; $\langle MATH_{15} \rangle$ is the number of parent class levels including both $\langle MATH_{16} \rangle$ and $\langle MATH_{17} \rangle$; $\langle MATH_{18} \rangle$ is the number of parent class levels including either $\langle MATH_{19} \rangle$ or $\langle MATH_{20} \rangle$; $\langle MATH_{21} \rangle$ represents the number of elements in $\langle MATH_{22} \rangle$. When $\langle MATH_{23} \rangle$ or $\langle MATH_{24} \rangle$, $\langle MATH_{25} \rangle$ or $\langle MATH_{26} \rangle$ is represented by 0.

From the above semantic similarity formula (3) and distance similarity formula (4), knowledge similarity can be obtained as shown in formula (5), where $\langle MATH_{27} \rangle$ is the relative weight of semantic similarity and distance similarity.

By calculating the knowledge similarity between new product development tasks and service teams, the matching degree between teams and development tasks is measured. The knowledge similarity of a team optimization combination is shown in formula (6), where $\langle MATH_{28} \rangle$ represents the knowledge similarity between candidate team $\langle MATH_{29} \rangle$ and the i-th sub-development task.

2.2 Collaborative Effect Evaluation Model

Team knowledge similarity measures R&D capability, while team collaborative effect measures interactive coordination capability. Team collaborative effect typically refers to the benefit of multiple teams coordinating and cooperating being greater than the sum of individual team benefits [14]. In parallel new product development team combinations, focusing only on each team' s ability to handle its own task often neglects the intrinsic connections between teams, resulting in developed products that fail to meet expected requirements. Therefore, this paper applies team collaborative effect to team selection in parallel new product development, pursuing the best overall collaborative effect of the team combination.

This paper analyzes factors affecting collaborative effect and, combined with the characteristics of team combinations in parallel new product development, summarizes evaluation indicators for team combination collaborative effect as shown in Table 1 .

Table 1. Team Combination Synergy Evaluation Index

Indicator	Description
Organizational Cloud Synergy (D1)	The degree of workflow and organizational structure synergy among teams within the combination
Technical Cloud Synergy (D2)	The degree of technology synergy among teams within the combination
Historical Interaction Cloud Synergy (D3)	The degree of synergy in historical interaction records among teams within the combination
Time Cloud Synergy (D4)	The degree of synergy in response time and communication time among teams within the combination

1) **Organizational Cloud Synergy (D1):** Each team in the service combination performs product development operations according to its own workflow and organizational structure. Process synchronization and organizational consistency can ensure information communication and timely problem adjustment between teams at various stages.

2) **Technical Cloud Synergy (D2):** Various professional technologies and related software support are required in the product process. Inconsistent

software usage, version differences, or non-unified professional technologies can greatly hinder information transfer and data exchange between teams.

3) **Historical Interaction Cloud Synergy (D3):** If teams have previous cooperation experience, they evaluate each cooperating team after each cooperation completion. Especially in cloud environments, the cloud platform records historical interaction data and information. If both parties have cooperation history records, collaborative effect evaluation is conducted based on historical interaction data; if no historical interaction exists, this collaborative effect is 0.

4) **Time Cloud Synergy (D4):** Team response time and whether communication time nodes are consistent determine the fit between teams and the timely update of relevant information and data and problem detection, which also affects subsequent development work.

The cloud platform stores large amounts of multi-faceted user data and possesses timely update and data analysis capabilities. Therefore, cloud platform experts can utilize big data and data mining techniques combined with historical interaction data of cloud platform users to conduct collaborative effect analysis, thereby making accurate assessments of the above collaborative effect evaluation indicator values.

Let the weight vector of each evaluation indicator in the table be $\langle MATH_{30} \rangle$. Through weighted rules, the collaborative effect value between teams in the team combination can be calculated. Where: $\langle MATH_{31} \rangle$ is the collaborative effect value between candidate team $\langle MATH_{32} \rangle$ and sub-team $\langle MATH_{33} \rangle$; $\langle MATH_{34} \rangle$ represents the i-th collaborative effect evaluation indicator. Formula (7) represents the average collaborative effect value between candidate team $\langle MATH_{35} \rangle$ and candidate teams within the combination; formula (8) represents the collaborative effect value of the team combination.

2.3 Service Quality Evaluation Model

In team combinations, the overall combination' s service quality must be considered, including estimated time, cost, and team reputation indicators for completing parallel new product development tasks. This paper defines the service quality information group as $\langle MATH_{36} \rangle$. The indicator calculation formulas for parallel new product development team combinations are shown in Table 2
.

The service quality evaluation model needs to consider the weight assignment $\langle MATH_{37} \rangle$ by cloud platform experts for each service quality indicator. Formula (9) is the service quality evaluation value of the team optimization combination, where $\langle MATH_{38} \rangle$ respectively represent the normalized results of time indicator, cost indicator, and reputation indicator for team optimization combination.

2.4 Comprehensive Evaluation Model

The evaluation model for team optimization combinations can be defined as a triple $\langle MATH_{39} \rangle$. According to Section 1.2, knowledge transfer between two teams generates knowledge rework time, knowledge communication time, and corresponding costs. For parallel new product development tasks, based on the model established in literature [11], the learning capability index of each team, the probability of erroneous knowledge release by the knowledge sender, the development difficulty of each sub-development task, and the relative dependency degree between tasks are specified. Under the principle of pursuing global benefit maximization, a knowledge transfer time mathematical model is established and the communication frequency between teams is solved. Knowledge rework time after each communication is solved based on the knowledge rework function, and the costs of knowledge communication and knowledge rework are calculated.

Development time $\langle MATH_{40} \rangle$ is the time consumed from when each team begins its task design to completion of the overall development task. The development time $\langle MATH_{41} \rangle$ for a single team includes knowledge operation time $\langle MATH_{42} \rangle$, knowledge rework time $\langle MATH_{43} \rangle$, and communication time $\langle MATH_{44} \rangle$. The time modification coefficient $\langle MATH_{45} \rangle$ represents the deviation degree between actual time and expected time, used to correct development time calculation. $\langle MATH_{46} \rangle$ refers to the time when the team completes the task; $\langle MATH_{47} \rangle$ refers to the expected task completion time; $\langle MATH_{48} \rangle$ refers to the total number of tasks submitted by the service demanding party that the team has completed.

Development cost $\langle MATH_{49} \rangle$ is the cost incurred from when each team begins its task design to completion of the overall development task. The development cost $\langle MATH_{50} \rangle$ for a single team includes knowledge operation cost $\langle MATH_{51} \rangle$, knowledge rework cost $\langle MATH_{52} \rangle$, and communication cost $\langle MATH_{53} \rangle$. Reputation $\langle MATH_{54} \rangle$ is an evaluation indicator by the cloud platform for service teams, reflecting a team' s reputation level. Higher reputation leads to stronger cooperation willingness from service demanding parties. The cloud platform provides a comprehensive indicator based on service teams' historical matching records and contract fulfillment status, with $\langle MATH_{55} \rangle$.

The selection of evaluation indicators, data collection, and quantification have been studied in depth in relevant research. Due to space limitations, this paper will not elaborate further.

Based on the above analysis, the comprehensive evaluation model is constructed. According to formula (10), the comprehensive evaluation value of the team combination is calculated, where $\langle MATH_{56} \rangle$ is the weight value obtained through analytic hierarchy process.

3 Improved Artificial Bee Colony Algorithm

Since the proposal of the Artificial Bee Colony (ABC) algorithm, relevant experts and researchers have improved upon the original algorithm, enhancing solution speed, stability, and model applicability. ABC algorithm improvements mainly focus on fitness function calculation and nectar source search mechanisms. This paper combines the characteristics of team combination optimization problems in cloud environments and the features of the established solution model to improve the fitness function calculation method and nectar source location update method of the ABC algorithm, proposing a Modified Artificial Bee Colony (MABC) algorithm.

In the standard ABC algorithm, bees have three roles: employed bees, scout bees, and onlooker bees. Employed bees determine nectar source $\langle MATH_{57} \rangle$ and visit it; onlooker bees select employed bees to follow based on probability value $\langle MATH_{58} \rangle$ according to shared nectar source information; scout bees discover new nectar sources after a source is abandoned [15]. The following sections describe the improvements to the standard ABC algorithm.

3.1 Fitness Function

The fitness function guides the population evolution direction, affecting algorithm iteration count and solution quality. Different fitness functions produce solutions of varying quality. The standard ABC algorithm uses the complex fitness value calculation function from literature [15]. This paper directly uses the objective function value of the comprehensive evaluation model to replace the fitness value. Since the three dimensions in the comprehensive evaluation model —knowledge similarity, collaborative effect value, and service quality evaluation value—all have values within [0,1], and the final comprehensive evaluation value is similar to the fitness value, it can determine nectar source quality and facilitate probability value calculation. The fitness function calculation method is shown in formula (11).

Where: $\langle MATH_{59} \rangle$ represents the fitness function; $\langle MATH_{60} \rangle$ is the weight value in the comprehensive evaluation model; $\langle MATH_{61} \rangle$ represents nectar source $\langle MATH_{62} \rangle$, a feasible solution to the problem, which is the three-dimensional metric value generated by the team combination; $\langle MATH_{63} \rangle$ is the function value obtained by the objective function on $\langle MATH_{64} \rangle$; $\langle MATH_{65} \rangle$ is the number of nectar sources, and $\langle MATH_{66} \rangle$ is the probability of onlooker bees selecting employed bees.

3.2 Search Method

The search methods of employed bees and scout bees determine the speed and accuracy of finding nectar sources. The standard ABC algorithm reflects algorithm randomness by randomly selecting another nectar source near the current one, but this limits solution precision and convergence speed. To improve algorithm performance, this paper improves the new nectar source location search method [16], as shown in formula (13).

Where: $\langle MATH_{67}\rangle$ is the new nectar source replacing $\langle MATH_{68}\rangle$; $\langle MATH_{69}\rangle$ is a random number in [-1,1]; $\langle MATH_{70}\rangle$ is a random number in [0,1]; $\langle MATH_{71}\rangle$ is the current iteration number; $\langle MATH_{72}\rangle$ is a random integer in [1,SN]; $\langle MATH_{73}\rangle$ is the j-th parameter of the current optimal nectar source $\langle MATH_{74}\rangle$; $\langle MATH_{75}\rangle$ and $\langle MATH_{76}\rangle$ are two adaptive control parameters used to balance search in different algorithm stages. In the early algorithm stage, $\langle MATH_{77}\rangle$ should be a larger value and $\langle MATH_{78}\rangle$ a smaller value; as iteration count increases, $\langle MATH_{79}\rangle$ gradually decreases while $\langle MATH_{80}\rangle$ gradually increases. During algorithm operation, through mutual cooperation between $\langle MATH_{81}\rangle$, self-regulation of the algorithm operation process is achieved, and changes in $\langle MATH_{82}\rangle$ adjust the search process to overcome the defect of the artificial bee colony algorithm easily falling into local optima [17].

The calculation of $\langle MATH_{83}\rangle$ is shown in formula (14).

Where: $\langle MATH_{84}\rangle$ is the current iteration number; $\langle MATH_{85}\rangle$ is the maximum iteration number; $\langle MATH_{86}\rangle$ and $\langle MATH_{87}\rangle$ are the minimum and maximum values.

Additionally, based on the characteristics of the team combination optimization problem, the solution space of the model has a small feasible region, making it difficult to randomly generate feasible solutions or infeasible solutions near the feasible region. When a nectar source is visited more than threshold $\langle MATH_{88}\rangle$ times, scout bees randomly search for new nectar sources. To improve this random situation, this paper improves its update method, as shown in formula (15).

Where: $\langle MATH_{89}\rangle$ is a random number in [-1,1]; $\langle MATH_{90}\rangle$ is a random integer in [1,SN]; $\langle MATH_{91}\rangle$ is the j-th parameter of the current optimal nectar source $\langle MATH_{92}\rangle$.

3.3 Algorithm Flow

Based on the above improvements, the algorithm is modified. The flow of cloud environment-based new product development team combination optimization using the improved artificial bee colony algorithm is shown in Figure 5 [Figure 5: see original paper].

---

4 Experimental Results and Analysis

To verify the feasibility and effectiveness of the MABC algorithm and comprehensive evaluation model in solving new product development team combination optimization problems in cloud environments, this paper constructs experimental examples through effective data collection and relevant data simulation. The experimental environment is MATLAB R2015a, Windows 10, 2.60 GHz, 8 GB RAM.

The experiment uses AGV trolley design as the background. The service demanding party submits new product development design task requirements for AGV trolleys to the cloud platform. Based on new product development task requirements and conditions, the cloud platform decomposes the design task into eight sub-development tasks as shown in Figure 6 [Figure 6: see original paper]. The interdependency degrees among subtasks are shown in Table 3 . Simultaneously, the cloud platform searches and matches to find relevant candidate development team sets for each sub-development task, with related information shown in Table 4 . The table includes knowledge similarity values of each team for subtasks processed by the knowledge similarity model, collaborative effect evaluation model data, and service quality evaluation model data.

To facilitate MABC algorithm solution of the model, collaborative effect evaluation model data are calculated and preprocessed. In the collaborative effect evaluation model, the weight vector is set as $\langle MATH_{93} \rangle$, and the inter-team synergy value matrix $\langle MATH_{94} \rangle$ is obtained. Additionally, the service quality evaluation model requires analysis of team communication and knowledge operation time and cost consumed after completing parallel new product development tasks. Using the model established in literature [11] and based on information from Tables 3 and 4, the communication frequency between teams after completing parallel new product development tasks is solved. According to the knowledge rework function, the knowledge rework time after each communication is solved, and the costs of knowledge communication and knowledge rework are calculated. The related information is shown in Table 6 , providing data support for team combination optimization. In the table, "3/7/12/14" respectively represent "communication time / communication cost / knowledge rework time / knowledge rework cost" . No collaborative effect evaluation is conducted among teams in the same team set, indicated by "–" .

4.2 Algorithm Solution

Based on the above initialization results, the MABC algorithm solves the team combination optimization evaluation model, with 100 repeated experiments conducted under the same parameter settings and environment. The MABC algorithm parameter settings are as follows: $\langle MATH_{95} \rangle$. Additional parameters for the new product development team selection comprehensive evaluation model are: $\langle MATH_{96} \rangle$; constraint information in the service quality evaluation model is: development time not exceeding 500, development cost not exceeding 8,000, and team combination reputation not lower than 0.88.

Based on candidate team sets for sub-development tasks, the experimental case has a total of 576 team combination methods, with different combination methods yielding different comprehensive evaluation values. The model and algorithm solve for the team combination with the highest comprehensive evaluation value. Using team combination knowledge similarity, collaborative effect value, and service quality as three-dimensional coordinates, the solution space of the case model is plotted as shown in Figure 7 [Figure 7: see original paper]. Due to relevant constraint conditions in the service quality evaluation model, infeasible solutions exist for team combinations.

According to the solved comprehensive evaluation results, the optimal team combination is $\langle MATH_{97} \rangle$. The comprehensive evaluation value of this optimal team combination is $\langle MATH_{98} \rangle$, where the knowledge similarity evaluation value is $\langle MATH_{99} \rangle$, the collaborative effect evaluation value is $\langle MATH_{100} \rangle$, and the service quality evaluation value is $\langle MATH_{101} \rangle$. Each evaluation value conforms to actual conditions. Based on these evaluation results, the improved artificial bee colony algorithm and comprehensive evaluation model are feasible for solving team combination optimization problems.

4.3 Effectiveness Validation and Algorithm Comparison

In the initial algorithm construction stage, this paper' s algorithm directly uses the objective function value of the comprehensive evaluation model to replace the fitness value. As iteration count increases, the fitness value gradually increases to the maximum fitness value. This paper analyzes the effectiveness of the MABC algorithm in handling team combination optimization problems by comparing the fitness function values before and after improvement, with experimental results shown in Figure 8 [Figure 8: see original paper].

Figure 8 reflects to some extent the impact of improvements on the artificial bee colony algorithm. Under the same iteration count, the MABC algorithm has higher fitness values; when achieving the same fitness value, the MABC algorithm consumes less time. Because the search method of the artificial bee colony algorithm is improved, convergence speed is accelerated to some extent and the algorithm avoids falling into local optima.

To further analyze the feasibility and efficiency of the MABC algorithm in handling team combination optimization problems, the MATLAB algorithm toolbox is used to introduce Ant Colony Optimization (ACO) algorithm [18] and Genetic Algorithm (GA). Under the same experimental environment, similar parameter settings are maintained. In the ACO algorithm, the information heuristic factor, expectation heuristic factor, and evaporation coefficient are set to 1, 2, and 0.8 respectively; in the GA algorithm, the crossover operator and mutation operator are set to 0.7 and 0.1 respectively. Each algorithm is run 100 times, and the data for the four algorithms handling new product development team combination optimization problems in cloud environments are obtained as shown in Table 7 . Through comparison of operation results (mean $\pm$ standard deviation), the MABC algorithm demonstrates advantages in solution performance, solution stability, and solution speed.

Table 7. Comparison of Algorithm Operation Results

Algorithm	Comprehensive Evaluation Value	Knowledge Similarity	Collaborative Effect Value	Service Quality Evaluation Value	Completion Time (s)
MABC	0.7850±	0.8210±	0.6530±	0.8809±	16.1±
ABC	0.7841±	0.8201±	0.6528±	0.8792±	20.3±
ACO	0.7844±	0.8205±	0.6529±	0.8798±	18.2±
GA	0.7838±	0.8199±	0.6527±	0.8788±	21.9±

5 Conclusion

This paper analyzes the entire process of new product development team combination optimization in cloud environments. For team combination optimization problems, a comprehensive evaluation model including knowledge similarity evaluation model, collaborative effect evaluation model, and service quality evaluation model is established. To ensure algorithm solution effectiveness and speed, the artificial bee colony algorithm is introduced and its fitness function and search method are improved. During team combination optimization model construction, this paper considers not only team R&D strength but also collaborative effects between teams and task constraint information. Additionally, through case experiments and analysis, the feasibility of the proposed optimization method and model in solving team combination optimization problems is verified; through algorithm comparison, the effectiveness of the improved algorithm in handling the proposed model is validated.

This paper measures the matching degree between teams and tasks through knowledge similarity to assess team R&D strength. Subsequent research should improve team strength evaluation indicators to more accurately measure team capabilities. Moreover, this paper adopts a pairwise information exchange mode between teams; future research could attempt multi-team information exchange to better reflect reality.

References

[1] Li Bohu, Zhang lin, Wang Shilong, et al. Cloud manufacturing: a new service-oriented networked manufacturing model [J]. Computer Integrated Manufacturing Systems, 2010, 16 (1): 1-8.

[2] Liu Jing, Yu Suihuai, Chu Jianjie, et al. Research on member optimal selection of network team [J]. Computer Integrated Manufacturing Systems, 2017, 23 (6): 1205-1215.

[3] Zhao Jinhui, Wang Xuehui. Two-sided matching model of cloud service based on QoS in cloud manufacturing environment [J]. Computer Integrated Manufacturing Systems, 2016, 22 (1): 104-112.

[4] Chen Youling, Yang Weiqi, Liu Chuanbiao, et al. Personnel assignment method of product design based on knowledge similarity and learning ability [J]. Computer Integrated Manufacturing Systems, 2017, 23 (8): 1700-1710.

[5] Jiang Yanping, Pan En, Liang Haiming. Method for member selection of cross-functional teams in new product development considering index aspiration [J]. Operations Research And Management Science, 2013, 22 (4): 204-211.

[6] Wang Juanru, Zhao Songzheng. On selection of partners using AHP and FPP in virtual team [J]. Industrial Engineering Journal, 2007 (6): 105-108.

[7] Lianying Zhang, Xiang Zhang. Multi-objective formation optimization for new product development [J]. Computers & Industrial Engineering, 2013, 64: 804-811.

[8] Feng Bo, Fan Zhiping. A partner selection method for knowledge creation team based on collaborative effect [J]. Chinese Journal of Management, 2012, 9 (2): 258-261.

[9] Feng Bo, Jiang Zhongzhong, Fan Zhiping, et al. A method for member selection of cross-functional teams using the individual and collaborative performances [J]. European Journal of Operational Research, 2010, 203: 415-424.

[10] Xu Ting. Research on simulation and optimization of overlapping design process based on Simio [D]. Jinan: Shandong University, 2016.

[11] Chen Youling, Lan Guihua, Liang Peixin. Estimation of overlapping time between upstream and downstream activities based on knowledge ohm' s law [J]. Computer Integrated Manufacturing Systems, 2016, 22 (12): 2759-2766.

[12] Ma Wenjian, Liu Wei, Li Chuanzhao. Overlapping and communication frequency between activities within concurrent product development [J]. Computer Integrated Manufacturing Systems, 2008 (4): 630-636.

[13] Lyu Yanjie, Zhao Gang, Yu Yong. Task assignment method of complex product design based on fuzzy set theory and knowledge similarity [J]. Computer Integrated Manufacturing Systems, 2015, 21 (4): 904-913.

[14] Li Jing, Zhang Yongan. An analysis of the affecting factors of collaborative effect of the logistics networks: a case study of Suning' s core logistics networks [J]. Journal of Beijing Jiaotong University: Social Sciences Edition, 2011 (4): 45-52.

[15] Ren Ziwu, Wang Zhenhua, Sun Lining. Forward kinematics solution for parallel manipulators based on improved artificial bee colony algorithm [J]. Journal of Mechanical Engineering, 2013, 49 (13): 48-55.

[16] Liang Yaosheng, Wan Zhongping, Fang Debin. An improved artificial bee colony algorithm for solving constrained optimization problems [J]. International Journal of Machine Learning and Cybernetics, 2017, 3 (8): 1-10.

[17] Li Yancang, Peng Yang. Improved artificial bee colony algorithm based on information entropy [J]. Control and Decision, 2015, 30 (6): 1121-1125.

[18] Xia Yamei, Chen Bo, Chen Junliang, et al. Optimizing services composition based on improved ant colony algorithm [J]. Chinese Journal of Computers, 2012, 35 (2): 2270-2281.