DOI: https://doi.org/10.46354/i3m.2019.emss.011

MODELING OF SERVICE CORRELATION FOR SERVICE COMPOSITION IN CLOUD MANUFACTURING

Fei Wang [a], Yuanjun Laili[a], Lin Zhang[a], Chi Xing[b],[c],[d], Liqin Guo[b],[c]

[a] School of Automation Science and Electrical Engineering, Beihang University
[b] Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center
[c] State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering
[d] Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center

[a]wangfeitime@163.com, [a]llyj0721@sina.com, [a]johnlin9999@163.com, [b],[c],[d]xingchi09@163.com, [b],[c]guo_96@163.com

ABSTRACT
Cloud manufacturing (CMfg), integrating distributed manufacturing resources as services to cloud center, aims at intelligent, green, and economic customized manufacturing. The optimal composition of services to fulfill particular manufacturing requirement is a core issue to realize efficient cloud manufacturing. Many researchers have studied the problem considering the Quality-of-Service (QoS) of independent services. However, the correlation between services is rarely considered. In this paper, the importance of service correlation is emphasized. Two kinds of service correlation, service exclusion and service collaboration, are modeled for service composition. An improved algorithm DET, which combines Differential Evolution Algorithm (DE) with a Tabu table based on service exclusive and collaborative relationships, is designed to filter composable services and find better solutions for complex tasks. Experiments have shown the effects of service correlation on the quality of composed services and demonstrated the effectiveness of the proposed method DET compared with traditional DE.

Keywords: cloud manufacturing, service correlation, service composition, differential evolution, tabu table

1. INTRODUCTION
Cloud manufacturing (CMfg), as a service-oriented intelligent manufacturing paradigm (Bo-Hu et al. 2010, LinZhang et al. 2014), means to integrate distributed manufacturing resources to produce more complex and flexible products efficiently and economically. Various manufacturing resources and capacities are connected to the CMfg center by using Internet-of-Things (IoT) techniques and are employed flexibly with virtualized cloud computing resources. It offers unified and centralized management for CMfg services (Zhou et al. 2017). Customers are able to acquire on-demand manufacturing services on the Internet (Liu et al. 2018). CMfg services are usually fine-grained and loosely-coupled with simple functions. To accomplish complex manufacturing tasks, service composition is essential to implement more complicated functions. The composed service not only satisfies the functionality requirements of customers, but also ensures good quality, economical price, and efficient process for hybrid manufacturing (Bouzary and Frank Chen 2018). Many services that have same functionality but different Quality-of-Service (QoS) are published by different providers (Cremene et al. 2016). Thus, research on service composition is carried out to find an optimal solution with the best overall QoS (Liu and Zhang 2017, Zhou and Yao 2017, Fazeli, Farjami, and Nickray 2019, Lartigau et al. 2015). Most researches only consider the situation that the services are independent with each other, while the correlation between services is often ignored. Services can be composed arbitrarily as long as they have the required functions (Feng et al. 2017). But in practice, the correlation between different services influences the QoS of services and determines whether these services can be composed in the same solution. Guo et al. investigated three kinds of correlations in services composition including composable relation, business entity relation, statistical relation (Guo et al. 2010). Tao et al. presented a QoS description mode supporting resource service correlation (Tao et al. 2010). Xu et al. considered correlation as one of the main service domain features. But correlation isn't involved in the process of service composition and wasn't analyzed adequately to improve the composition results (Xu et al. 2017).
In this paper, two kinds of service correlation, service exclusion and service collaboration are analyzed. Their models and descriptions are established. In addition, an improved Differential Evolution Algorithm, termed DET, is designed to deal with the correlation between services. DET adopts Tabu table to avoid the coexistence of excusive services in the same the composition solution. Besides, the influence of collaborative services in one solution is calculated in DET to search for optimal solution with the better overall QoS. Finally, the experiment shows the effect of service exclusion and collaboration. The effectiveness of DET is demonstrated.

This paper is organized as follows. Section 2 describes the motivating example of two kinds of service correlation. The model of service correlation and correlation-aware service composition is given in Section 3. Section 4 provides the details of DET to process correlations in service composition. Section 5 reports the effects of two kinds of correlation and the effectiveness of the proposed method by experiments. Section 6 concludes this work.

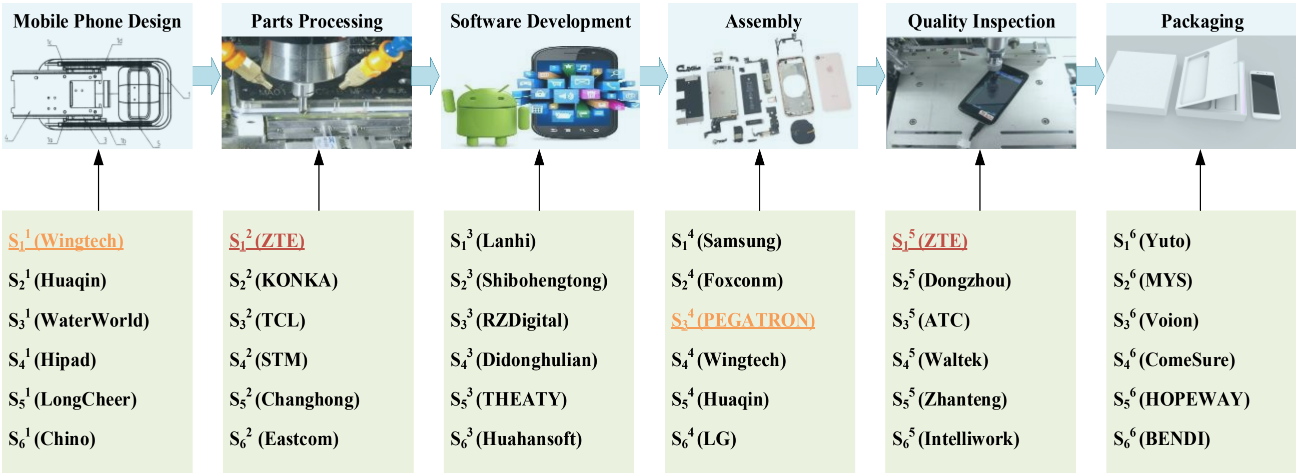

Figure 1: The Example of Mobile Phone Manufacturing

2. MOTIVATING EXAMPLE

Before introducing the main content, an example of mobile phone manufacturing is shown in Figure 1. It contains six subtasks, i.e. design, parts processing, software development, assembly, quality inspection and packaging. Each subtask is accomplished by a service. There are a set of candidate services available for handling each subtask. In Figure 1, the boxes below the corresponding subtasks show their candidate service sets. To finish the task of producing a batch of mobile phones, a group of services are to perform the subtasks. Usually, a good solution of service composition has better individual candidates with better QoS in terms of service cost, time and reliability. But the selection of services for all subtasks are not always independent.

For example, for the mobile phone manufacturing task in Figure 1, a candidate service of parts processing subtask S_1^2 and candidate service of quality inspection subtask S_1^5 are both provided by ZTE. To promote services ZTE, the costs of S_1^2 and S_1^5 are discounted by 10% if both of them are selected for a manufacturing task simultaneously.

On the contrary, if S_1^1 is selected to perform the design subtask, the service for assembly subtask can't be S_3^4. Because the craft of S_1^1 Wingtech is conflict with which of S_3^4 PEGATOR, Wingtech refuses to cooperate with PEGATOR.

These two examples demonstrate two types of correlations. In this paper, the correlation in the first situation is defined as service collaboration, which influences the QoS values of services. The second correlation is called service exclusion, that is, the service selection of one subtask determines whether some services can be selected for another subtask. Both kinds of correlation are modeled in Section 3.2 in detail.

3. A CORRELATION-AWARE SERVICE CORRELATION MODEL

In this section, the basic model of service composition is outlined firstly. Then the influences of two kinds of service correlation are analyzed. The models of the two kinds of service correlations and their effects on service composition are provided. Afterwards, a correlation-aware service composition model is introduced in the next two subsections.

3.1. Basic Service Composition Model

The aim of service composition is to obtain the optimal composition solution with the best non-functionality performance to guarantee high-efficient manufacturing (Bo and Fan 2007). The QoS attributes of a CMfg service, involving time, cost, reliability, and maintainability, measure its non-functional performance quantitatively from different aspects (Fei et al. 2009). Their definitions are as follows:

- Time: The duration of a service for executing a given task.
- Cost: The price that a customer should pay for renting a particular service in a period of time.
- Reliability: The probability that a manufacturing service can execute correctly in a certain time.
- Maintainability: The ease with which a manufacturing service can be maintained in order to recover from failure or to upgrade from current state.
- Trust: The credit of a service in dealing with a particular kind of task

Among the above QoS attributes, time and cost are called negative QoS, whose smaller values indicate better performance. Meanwhile, reliability, maintainability and trust are positive QoS, which means larger values are more desirable.

The notations used in correlation-aware service composition model are listed in Table 1.

Table 1: Notations

Notation	Meaning
T_i	The *i*th subtask of MT
M	The number of subtasks of MT
CSS_i	The candidate service set of T_i
S_j^i	The *j*th service in CSS_i
N_i	The number of services in CSS_i
CS	The composite service composed by *m* candidate services of *m* subtasks from CSS_1, CSS_2,⋯, CSS_m respectively
$T(S_j^i)$	The time value of S_j^i
$C(S_j^i)$	The cost value of S_j^i
$R(S_j^i)$	The reliability value of S_j^i
$M(S_j^i)$	The maintainability value of S_j^i
$Tr(S_j^i)$	The trust value of S_j^i
$TT(CS)$	The overall time value of *CS*
$TC(CS)$	The overall cost value of *CS*
$TR(CS)$	The overall reliability value of *CS*
$TM(CS)$	The overall maintainability value of *CS*
$TTr(CS)$	The overall trust value of *CS*
$SC(S_j^i)$	The exclusive service set of S_j^i
$QC(S_j^i)$	The collaborative service set of S_j^i

The description of CMfg service is the basis of service composition. The basic description of a CMfg service includes its functionality, the provider and the values of all QoS attributes.

$B(S_j^i)$={*Functionality*, *Provider*, *Td*, *Cd*, *Rd*, *Md*, *Trd*}

where $B(S_j^i)$ denotes the basic description of S_j^i. *Td*, *Cd*, *Rd*, *Md* and *Trd* are the default values of the QoS attributes without considering service collaboration. An example is shown as follows:

Basic description of S_1^2:
Functionality: Mobile phone parts processing
Provider: ZTE

Td: 15 Hours
Cd: 100 USD
Rd: 0.92
Md: 0.89
Trd: 0.95

Given a manufacturing task containing *n* subtasks, the CMfg platform provides *n* sets of candidate services that match the functionality requirements of *n* subtasks respectively. Service composition is to select a service from the candidate service set for each subtask to form a composite solution with optimal overall QoS. For a manufacturing task whose subtasks are executed sequentially, a composite service *CS* is composed by m services $\{S_j^1, S_j^2, \dots, S_j^m\}$. The calculation formula of the overall QoS values of *CS* are as follows:

$$TT(CS) = \sum_{i=1}^{m} T(S_j^i) \quad (1)$$

$$TC(CS) = \sum_{i=1}^{m} C(S_j^i) \quad (2)$$

$$TR(CS) = \sum_{i=1}^{m} R(S_j^i) \quad (3)$$

$$TM(CS) = \sum_{i=1}^{m} M(S_j^i) \quad (4)$$

$$TTr(CS) = \sum_{i=1}^{m} Tr(S_j^i) \quad (5)$$

The comprehensive utility of a composite service is evaluated by the weighted sum of the above overall QoS values. The QoS values are normalized to real numbers between 0 and 1.

$$\begin{aligned} f(CS) = {} & \omega_1 \bullet \frac{TT(CS) - TT_{\min}}{TT_{\max} - TT_{\min}} + \omega_2 \bullet \frac{TC(CS) - TC_{\min}}{TC_{\max} - TC_{\min}} \\ & + \omega_3 \bullet \frac{TR_{\max} - TR(CS)}{TR_{\max} - TR_{\min}} + \omega_4 \bullet \frac{TM_{\max} - TM(CS)}{TM_{\max} - TM_{\min}} \\ & + \omega_5 \bullet \frac{TTr_{\max} - TTr(CS)}{TTr_{\max} - TTr_{\min}} \end{aligned} \quad (6)$$

where *f*(CS) represents the comprehensive utility of the composite solution *CS* and $\omega_i (i = 1,2,\dots,5)$ is the weight of QoS values that satisfies $\sum_{i=1}^{5} \omega_i = 1$. The main objective of service composition is minimizing *f*(*CS*).

3.2. Service Correlation Model

In this paper, two kinds of service correlations, service exclusion and service collaboration, are modelled.

3.2.1. Service Exclusion

The exclusion set of a service S_j^i specifies services that are unable to be composed with S_j^i in the same service composition solution. The exclusion set of S_j^i is expressed as follows:

$$SC(S_1^1) = \{S_1, S_2, \cdots, S_k\} \quad (7)$$

where $SC(S_j^i)$ is the set of exclusive services of S_j^i and *k* is the number of them.

Taking the second situation of Section 2, the service exclusion description of S_1^1 is shown below:

Service exclusion description of S_1^1:
$SC(S_1^1)$={ S_5^2, S_6^2,S_3^4, S_3^5}

If S_5^2 cooperates with S_1^1 in the same composition solution, their exclusion makes this solution infeasible. Thus, considering service exclusion, candidate services of different subtasks can't compose with other services arbitrarily in case of conflict with their exclusive services.

The selected service of one subtask may constrain the service selection for other subtasks. Service exclusion becomes a constraint for service composition.

3.2.2. Service Collaboration

Service collaboration indicates how other services influence the QoS values of a certain service in the case that they are composed in the same composition solution. Correlation coefficient in the form of percentage is defined to measure the degree of service collaboration. The service collaboration of S_j^i is expressed as follows.

$$QC\left(S_j^{\,i}\right) = \left\{\left(S_1, q_1, c_1\right), \left(S_2, q_2, c_2\right), \cdots, \left(S_k, q_k, c_k\right)\right\} \quad (8)$$

where $QC(S_j^i)$ is the set of tuples that consist of quality correlated service S_i, the influenced QoS attribute q_k and its correlation coefficient c_i. The correlation coefficient c_i could be a positive number or a negative number, which depends on whether the influence is beneficial or not. If S_j^i and S_k are composed in one solution, the actual QoS value of S_j^i is calculated as follows.

$$qa_k\left(S_j^{\,i}\right) = qd_k\left(S_j^{\,i}\right) \bullet \left(1 - c_k\right) \quad (9)$$

where $qa_k(S_j^i)$ and $qd_k(S_j^i)$ are the actual and default values of the QoS attribute q_k of S_j^i.

In the first situation of Section 2, the service collaboration description of S_1^2 is shown as follows:

Service collaboration description of S_1^2:
$QC(S_1^2)=\{(S_1^5, T, 10\%)\}$

According to the service collaboration description and basic description of S_1^2, when S_1^2 and S_1^5 are composed in the same composition solution, the actual cost of S_1^2 $Ca(S_1^2)$ is calculated as follows:

$$Ca(S_1^{\,2}) = Cd(S_1^{\,2}) \bullet \left(1 - c\right) = 100 \bullet \left(1 - 10\%\right) = 90USD \quad (10)$$

The basic description, service exclusion description and service collaboration description constitute the description of one service together and form the basis of correlation-aware service composition.

4. THE IMPROVED DIFFERENTIAL EVOLUTION ALGORITHM FOR CORRELATION-AWARE SERVICE COMPOSITION

After modeling two kinds of correlation, the method based on Differential Evolution Algorithm (DE) is proposed to solve correlation-aware service composition problem. As a stochastic real-parameter optimization algorithms, DE has been used in QoS-aware service composition problem to implement global optimization (Pop et al. 2011). The proposed method focuses on how to manage service exclusion and service collaboration in the process of service composition.

Considering service correlation, exclusion conflict between services in one solution makes this solution infeasible. Thus correlation-aware service composition is a constrained optimization problem. In this paper, an improved algorithm DET is designed to ensure the feasibility of the obtained solutions. A Tabu table tailored for service exclusion to store the exclusive services, which avoids the exclusion conflict between services in one solution for correlation-aware service composition problem. The Tabu table is maintained according to the service exclusion description of services to filter the composable services.

4.1. Genotype Encoding

In DE of this paper, a genome $X_i = (S_{i1}, S_{i2}, \ldots, S_{iM})$ represents a composition solution that contains the selected services of all subtasks. It is encoded as an array of real number as is shown in Figure 2. The length of the genome array M is the number of subtasks in service composition. S_{ij} in the array refers to the selected service of the subtask T_j. S_k^j is the kth candidate service of T_i. N_i is the number of candidate services of subtask T_i.

The value of S_{ij} is a real number between 0 and 1. It represents the corresponding candidate service S_k^j according to Equation 11.

$$k = \left\lfloor S_{ij} \bullet N_i \right\rfloor \quad (11)$$

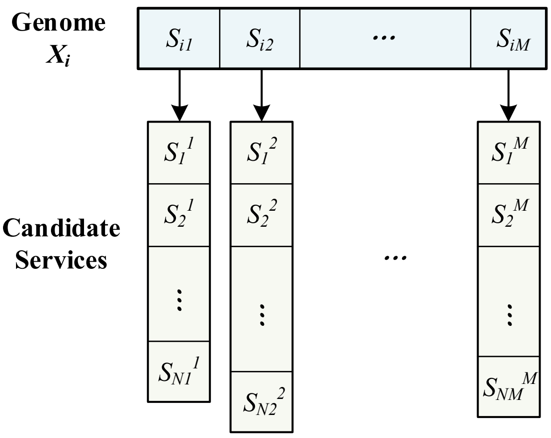

Figure 2: Encoding of Genome

4.2. Main Loop of the Proposed Method

The main loop of DE for correlation-aware service composition is presented in Algorithm 1.

Algorithm 1 Main loop of DE for correlation-aware service composition

Input: Population P, population size L, maximum number of generations $Gmax$
Output: The best solution *ind_best*

```
initialize population P by selecting n solutions from candidate service sets at random
g ← 0
while g ≤ Gmax do
    P′ = Mutation(P)
    P″ = Crossover(P′)
    Fitness Assignment (P″)
    P = P″
    g ← g+1
end while
```

The following introduces the detail of mutation, crossover and selection operators.

4.3. Mutation Operator

Firstly, we use $X_{bg} = (S_{g1}, S_{g2}, \ldots, S_{gM})$ to store the global best solution of all individuals in P during the iteration. And an array $X_{bi} = (S_{bi1}, S_{bi2}, \ldots, S_{biM})$ with the length of L to record the best solution of ith individual in P. In each generation, five individuals (i=1, 2, …, 5) are randomly chosen to conduct mutation operation. $X_w = (S_{w1}, S_{w2}, \ldots, S_{wM})$ is the newly generated genome after mutation. There are 3 kinds of mutation strategies for choice.

$$S_{wj} = S_{gj} + F \bullet \left(S_{1j} - S_{2j}\right) \quad (12)$$

$$S_{wj} = S_{b1j} + F \bullet \left(S_{b2j} - S_{b3j}\right) \quad (13)$$

$$S_{wj} = S_{b1j} + F \bullet \left(S_{b2j} - S_{b3j}\right) + F \bullet \left(S_{b4j} - S_{b5j}\right) \quad (14)$$

Where F is the scaling factor. The first mutation strategy has the priority to be chosen. The mutation of genome bits is executed in sequence. Once S_{w1} is calculated, its exclusive services are recorded in the Tabu table of this solution. Then S_{w2} is calculated according to the first strategy. To avoid exclusion conflict, two conditions need to be checked:

- S_{w2} doesn't exist in the Tabu table;
- S_{w1} isn't in the exclusive service set of S_{w2}.

If these two conditions are satisfied, S_{w2} is accepted and the services that are exclusive with S_{w2} are added into the Tabu table. Otherwise, there is exclusion conflict between S_{w1} and S_{w2}. S_{w2} needs to be recalculated according to the second mutation strategy. If it's still infeasible, the third mutation strategy is adopted. If there is still correlation conflict, S_{w2} is set to a random number whose corresponding service satisfy the above two conditions.

Once a genome bit S_{wj} is generated by a mutation strategy, it is checked whether it is included in the Tabu table and whether the previous services, that is, S_{w1}, S_{w2},a…, $S_{w(j-1)}$ are in the exclusive service set of S_{wj}. If S_{wj} doesn't satisfy these two conditions, it is recalculated by another mutation strategy, until S_{wj} doesn't have conflict with the previous services.

4.4. Crossover Operator

Crossover operator generates the new population based on the results of mutation operator. For the jth gene S_j(t+1) of the individual in (t+1) generation, randomly generate a number r between 0 and 1. If r is less that the probability p_c for X_w, the prior service is S_{w1}. Otherwise, the prior service is S_j(t). Firstly, check whether the prior service satisfies the above two conditions. If so, the prior service is selected. If not, check another alternative and repeat the above steps. If it's also unsatisfying, generate a random service S_r that meet these two conditions.

4.5. Fitness Assignment

The exclusion conflict has been avoided in the mutation and crossover operator by Tabu table. Thus, the obtained individuals are feasible. Firstly, the service collaboration between services in one solution is checked. If there is service collaboration, the actual QoS values of services in the solution are calculated according to Equation 9. Then the actual QoS values is put into Equation 6 to calculate the fitness value of the solution, which is the weighted sum of the normalized QoS values of all services in an individual.

5. EXPERIMENTS

To evaluate the effectiveness of the method proposed in this paper, we carry out two sets of experiments. The first set of experiments verifies the effect of service exclusion. The second set of experiments evaluates the influence of service collaboration and proves the feasibility and effectiveness of the proposed algorithm DET. Simulation experiments are conducted on a PC with Intel Core i7 CPU 3.6 GHz, 8 GB RAM, Windows 10, Microsoft Visual Studio V12.0.

As is shown in the motivating example of Figure 1, the manufacturing task can be decomposed to several subtasks. For every subtask, there are many candidate services that are matched by their functionality. In the two sets of experiments of this paper, the manufacturing task is set with 10 subtasks and 100 candidate services for each subtask. The population size is 20. Their QoS values are random numbers. Their ranges are listed in Table 1.

Table 1: The Ranges of QoS Attributes

QoS Attributes	Ranges of Random Values
Time/Hour	60.0-80.0
Cost/(100USD)	70.0-100.0
Reliability	80.0%-99.9%
Maintainability	80.0%-99.9%
Trust	80.0%-99.9%

The weights of five attributes in Equation 6 are 0.2. The maximum iteration generation of the algorithm is 30000.

5.1. The effect of Service Exclusion

Service exclusion makes some solutions infeasible. To reveal the effect of service exclusion, the ratio of infeasible solutions of each generation is calculated during the iteration of DE without Tabu table. In each generation, the number of infeasible solutions in the population is calculated, which shows the effect of service exclusion. In the first experiment, the number of exclusive services of each candidate service varies from 10 to 200. The exclusive services are randomly chosen from the candidate services of other subtasks. The number of infeasible solutions in every generation is observed. The average ratios of infeasible solutions in 30000 generations are shown in Figure 3. It can be seen that the ratio of infeasible solutions grows remarkably with the increase of exclusive services.

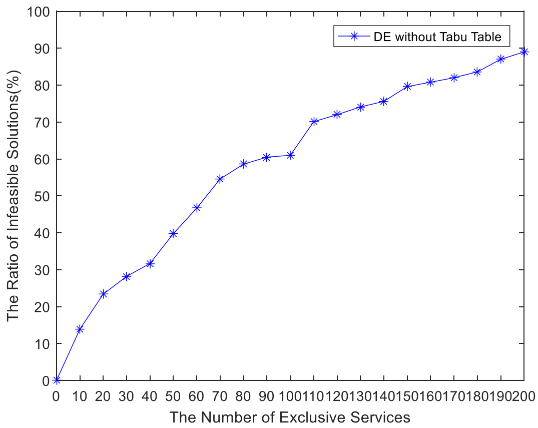

Figure 3: Ratio of Infeasible Solutions w.r.t. Number of Exclusive Services of Each Candidate Service

Figure 3 shows that when the number of exclusive services of each candidate service reach 200, 89% of the solutions on average are infeasible. The above result demonstrates that service exclusion can't be ignored in service composition because not a few solutions are infeasible for this reason. Thus, DE with Tabu table is designed and realized to avoid exclusive services in one solution. The experiment result of DE with Tabu table shows that all solutions in the population of are feasible, which proves the effectiveness of the improved DE for solving service exclusion in correlation-aware service composition.

5.2. The Effect of Service Collaboration

Service Collaboration benefits the QoS performance of collaborative services in the same solution. In the second experiment, the number of collaborative services of each candidate service varies from 0 to 90. The correlation coefficient ranges from 1% to 15%. Meanwhile, the optimal results of DET and DE are recorded and compared. The number of exclusive services of each candidate service is 10. The infeasible solutions of DE are excluded. The results are shown in Figure 4.

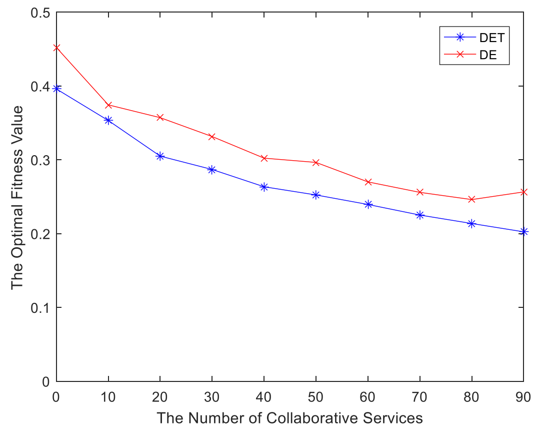

Figure 4: Optimal Fitness Value w.r.t. Number of Collaborative Services of Each Candidate Service

It can be seen from Figure 4 that the optimal fitness value obtained by DET decreases from 0.396 to 0.202 while the number of collaborative services increase to 90. For both DET and DE, the more collaborative services are, the smaller the obtained optimal fitness value is. Because more collaborative services make more solutions whose services are collaborative with each other. Meanwhile, because DET has the ability to tackle service exclusion, the proposed algorithm DET outperforms DE to search solution with optimal overall QoS considering service correlation.

6. CONCLUSION

In this study, service correlation is analyzed, and two kinds of service correlation model are built. An improved Differential Evolution Algorithm is proposed for correlation-aware service composition. The main contributions of this study are as follows.

- The model of two kinds of service correlation describes their effect on service composition, which is the basis of correlation-aware service composition.
- The proposed algorithm DET is employed to handle the above two kinds of service correlation and obtain composition solution with optimal overall QoS.

In future research, more kinds of service correlation will be investigated considering the complex relationship among services in cloud manufacturing. Besides, the method to deal with correlation-aware service composition for CMfg tasks with complex structures will be researched.

ACKNOWLEDGMENTS

The research in this paper was supported by grants 61873014 from the National Natural Science Foundation of China.

REFERENCES

Bo-Hu, L. I., Lin Zhang, Shi Long Wang, Fei Tao, Jun Wei Cao, Xiao Dan Jiang, Xiao Song, and Xu Dong Chai, 2010. Cloud manufacturing:a new service-oriented networked manufacturing model. Computer Integrated Manufacturing Systems, 16 (01), 1-7+16.

Bo, Liu, and Yushun Fan. 2007. "Research on Service-Oriented Workflow and Performance Evaluation." IEEE International Conference on Web Services.

Bouzary, Hamed, and F. Frank Chen, 2018. Service optimal selection and composition in cloud manufacturing: a comprehensive survey. The International Journal of Advanced Manufacturing Technology, 97 (1), 795-808.

Cremene, Marcel, Mihai Suciu, Denis Pallez, and D. Dumitrescu, 2016. Comparative analysis of multi-objective evolutionary algorithms for QoS-aware web service composition. Applied Soft Computing, 39 (C), 124-139.

Fazeli, Mohammad Moein, Yaghoub Farjami, and Mohsen Nickray, 2019. An ensemble optimisation approach to service composition in cloud manufacturing. International Journal of Computer Integrated Manufacturing, 32 (1), 83-91.

Fei, Tao, Yefa Hu, Dongming Zhao, Zude Zhou, Haijun Zhang, and Zhenzhen Lei, 2009. Study on manufacturing grid resource service QoS modeling and evaluation. International Journal of Advanced Manufacturing Technology, 41 (9-10), 1034-1042.

Feng, Li, Zhang Lin, Yongkui Liu, Yuanjun Laili, and Tao Fei, 2017. A clustering network-based approach to service composition in cloud manufacturing. International Journal of Computer Integrated Manufacturing, 30 (3), 1-12.

Guo, Hua, Fei Tao, Lin Zhang, Suiyi Su, and Nan Si, 2010. Correlation-aware web services composition and QoS computation model in virtual enterprise. The International Journal of Advanced Manufacturing Technology, 51 (5-8), 817-827.

Lartigau, Jorick, Xiaofei Xu, Lanshun Nie, and Dechen Zhan, 2015. Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. International Journal of Production Research, 53 (14), 4380-4404.

LinZhang, YongliangLuo, FeiTao, Hu Li Bo, LeiRen, XuesongZhang, HuaGuo, YingCheng, AnruiHu, and YongkuiLiu, 2014. Cloud manufacturing: a new manufacturing paradigm. Enterprise Information Systems, 8 (2), 167-187.

Liu, Bo, and Zili Zhang, 2017. QoS-aware service composition for cloud manufacturing based on the optimal construction of synergistic elementary service groups. The International Journal of Advanced Manufacturing Technology, 88 (9), 2757-2771.

Liu, Yongkui, Lihui Wang, Xi Vincent Wang, Xun Xu, and Lin Zhang, 2018. Scheduling in cloud manufacturing: state-of-the-art and research challenges. International Journal of Production Research, 1-26.

Pop, Florin Claudiu, Denis Pallez, Marcel Cremene, Andrea Tettamanzi, Mihai Suciu, and Mircea Florin Vaida. 2011. "QoS-based service optimization using differential evolution." Conference on Genetic & Evolutionary Computation.

Tao, Fei, Dongming Zhao, Hu Yefa, and Zude Zhou, 2010. Correlation-aware resource service composition and optimal-selection in manufacturing grid. European Journal of Operational Research, 201 (1), 129-143.

Xu, Xiaofei, Zhizhong Liu, Zhongjie Wang, Quan Z. Sheng, Jian Yu, and Xianzhi Wang, 2017. S-ABC: A paradigm of service domain-oriented artificial bee colony algorithms for service selection and composition. Future Generation Computer Systems, 68, 304-319.

Zhou, Jiajun, and Xifan Yao, 2017. A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition. The International Journal of Advanced Manufacturing Technology, 88 (9), 3371-3387.

Zhou, Longfei, Zhang Lin, Bhaba R. Sarker, Yuanjun Laili, and Ren Lei, 2017. An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing. International Journal of Computer Integrated Manufacturing, 31 (3), 1-16.