Alliance-aware Service Composition Based on Quotient Space

Yiwen Zhang[†], Guangming Cui[†], Shuiguang Deng[*] and Qiang He[‡]
[†]*School of Computer Science and Technology, Anhui University, Hefei, Anhui, China*
[*]*College of Computer Science and Technology, Zhejiang University, Hangzhou, China*
[‡]*School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia*
Email: zhangyiwen@ahu.edu.cn; cgm133123@163.com; dengsg@zju.edu.cn; qhe@swin.edu.au

***Abstract*—Along with the progress of the enterprise globalization, alliance and cooperation have become an important means for enterprises to improve their competitiveness in the market. Yet, most current methods for Service Composition Optimization (SCO) fail to address the Alliance Relation (AR) between services and assume that services are independent of each other. To address this issue, this paper presents an alliance-aware service composition method. Firstly, the fundamental properties of the AR are given based on a multi-granularity service composition model. Secondly, alliance relation granularity is coarsened into a relation granulation quotient space and the domain elements are matched reversely with service compositions, thereby reducing the complexity of query and computation of the AR. Finally, a Relation Granularity-aware Particle Swarm Optimization Algorithm (RG-PSO) is proposed based on relation granulation quotient space to solve the alliance-aware SCO prolem. Substantial experimental results show that the proposed model and algorithm are effective and efficient.**

***Keywords*-Alliance Relation (AR); Service Composition; Quotient Space; Granulation; Domination Relation (DR)**

I. Introduction

Service-oriented architecture (SOA) and its key implementation, web services, provide a promising solution for seamless integration of single-functional applications to create new large-granularity and value-added services. In recent years, the development and popularity of e-business, especially the pay-as-you-go business model promoted by cloud computing have fueled the growth of Web services [1], [2]. There are more and more functional-equivalent services available at different quality-of-service (QoS) levels. Thus, the service selection for QoS-aware service composition can be very complicated, a NP-complete problem [15] in fact, because the selected services must achieve an optimisation goal while fulfilling all quality constraints for the composite service. Many optimization approaches have been proposed based on different techniques, e.g., particle swarm optimization (PSO) [20], genetic algorithm (GA) [19], integer programming [3], dynamic programming [5] and graph algorithm [18].

With the global economic integration, alliance and cooperation have become a popular means for enterprises to improve their competitiveness in the market. The number of alliances continues to grow at an unprecedented rate. Accordingly, there are certain alliance relations (ARs) between the services, e.g., quality complementarity [16]. For instance, in 2000 alone, approximately 10200 alliances were formed worldwide [17]. Not only does the number of alliances continue to grow, but so does their significance to the allied firms. A study conducted by Accenture report that about 25% of executives said alliances account for at least 15% of their market value. Partner Alliances reports that 82% of Fortune 1000 CEOs believe alliances will be responsible for more than 26% of their companies' revenues [17]. Therefore, AR has an important influence on the service selection for quality-aware service composition in the real world. However, most current methods for QoS-aware service composition assume that the quality of the services are independent from each other, and fail to address the intrinsic link between the quality of different services. For example, a user's online shopping process is shown in Fig.1.

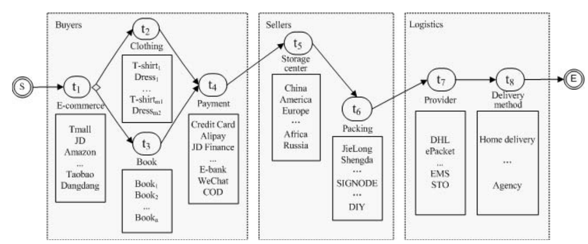

Fig. 1: The example of user's shopping

Usually, there is AR between the E-business platform service t_1 and the payment service t_4. For example, for JD business platform (http://en.jd.com), users can obtain a 5% discount on their purchases if they use JD finance service for the payment. In addition, there is an AR between the E-business service provider and the logistics service provider. For instance, if a JD American user buys a men's steel watch GS5732 and chooses the ePacket logistics services, the product delivery is free. If they choose DHL, it will cost them $33.09. The AR between two businesses may also impact the other quality dimensions of their services rather than just the prices. For example, the delivery service provided by the e-business platform itself usually promises a shorter delivery time compared to a third-party delivery service. Such ARs between different service providers have

978-1-5090-2675-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICWS.2016.51

strong influence on the composite services. With global economic integration continues to advance, AR is becoming increasingly popular and must not be ignored during the process of service composition.

The consideration of the AR between businesses further complicates the service selection problem. The solution space can be extremely large when the number of tasks and the number of candidate services for each task are large. To tackle this challenge, we introduce a novel service composition approach called ASC_QS (Alliance-aware Service Composition based on Quotient Space). The major contributions of this paper are as follows:

1) The concept of relation granulation quotient space is proposed to model AR. According to the falsity preserving principle and truth preserving principle, the completeness of the relation granulation quotient space is ensured.
2) A new alliance-aware SCO method, namely ASC_QS, is proposed based on the theory of quotient space granular computing. It analyses and demonstrates some properties and theorems of relation granulation theoretically, and verifies the effect of AR on SCO problem.
3) The relation granulation particle swarm optimization (RG-PSO) algorithm is proposed to solve the alliance-aware SCO problem. We effectively reduce the complexity of the queries and calculation in AR by the theory of relation granulation quotient space.
4) A series of experiments were conducted on two data sets to evaluate the effectiveness and efficiency of our method.

The rest of the paper is organized as follows. Section II reviews the related work. Section III motivates this research through an example scenario. Section IV describes the AR and relation granularity quotient space. Section V presents our ASC_QS approach. Section VI describes the experiments and the results. Section VII concludes this paper and discusses future directions.

II. Related Work

Service selection for service composition optimization (SCO) has been widely and intensively investigated in the past decade. Existing approaches can be divided into two categories. The first is intelligent algorithm based approaches. The PSO [20] and GA [19] have been widely used in the area of service composition. In addition, Y. Zhang [4] proposed a large-scale QoS-aware SCO method and solved SCO problem fast by improved FOA; There are also many programming techniques based approaches for SCO. M. Alrifai et al. [3] adopted a hybrid methodology by applying mixed integer programming (MIP) to seek out the best decomposition of QoS constraints into native constraints and to the simplest web service that satisfies all these constraints. V. Uc-Cetina [5] proposed a web SCO algorithm based on dynamic programming. V. Gabrel [6] presented a method to find optimal solution for transactional Web service composition using dependency graph and 0-1 linear programming. However, all the methods above are established on the assumption that each service is independent of the others, which is not realistic. Thus, they are not suitable in an environment where AR is common and have direct impact on the service selection for QoS-aware SCO.

In recent years, QoS correlation has been an active research topic. F. Wagner [7] proposed an approach that takes into account the time and input aspects which affect the QoS values of a service. However, this approach addresses the service selection problem with QoS correlations but does not incorporate QoS correlation directly into the composition process. S. Deng et al. [8] identified the QoS correlation problem and designed a method to solve this problem for the first time. However, it only considers that a service provider offers many applications of various services without taking into account the alliance between different service providers. Q. Wu [9] presented a business correlation model, and utilized an improved GA to solve the QoS-constraint SCO problem. But their approach only allows service composition optimization between two abstract tasks and the practicality is significantly limited. With the rapid increase in the number of tasks and the number of candidate services, the difficulty in the query and calculation of QoS constraint relation grows exponentially and it is hard for GA method to achieve satisfactory results.

Therefore, one major limitation to existing approaches is that the AR is not considered among services. Another major limitation is that the granularity of the service composition process, which is widely available in real-world service composition applications, has been largely overlooked in the research of Web service composition.

III. Motivating Example

AR is universal among businesses in the real world. It is a common means for profit maximization by establishing cooperation relationship between business services. As presented by the online shopping example shown in Fig.1, alliance relations are very common and various, such as the free postage offered by alliance between sellers and logistics companies. In order to demonstrate how AR is captured in the SCO model, we utilise the business process of customer purchasing clothing as an example in Fig.2, where ellipses denote tasks and circles denote services. The value of each service denotes the cost of the service. The arc denotes the AR between two services and the value on each arc denotes the cost privilege i.e., price deduction or discount.

Fig.2 shows that without considering the AR, the optimization composition scheme is $s_{1,1} \rightarrow s_{2,1} \rightarrow s_{4,1}$ and the minimal cost is 836+672+636=2144. Now consider the AR between the services. The optimization composition scheme is $s_{1,2} \rightarrow s_{2,3} \rightarrow s_{4,2}$ and the minimum cost is

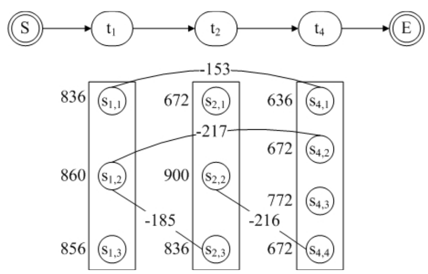

Fig. 2: The customer purchase process

860+ (836-185) + (672-217) =1966. Existing approaches for SCO do not consider the AR and hence cannot find optimal composition schemes in real-world scenarios where ARs exist between businesses, i.e., service providers.

Table I: alliance relation matrix

$s_{4,4}$	/	/	/	/	-216	/	/	/	/	*all*
$s_{4,3}$	/	/	/	/	/	/	/	/	*all*	/
$s_{4,2}$	/	-217	/	/	/	/	/	*all*	/	/
$s_{4,1}$	-153	/	/	/	/	/	*all*	/	/	/
$s_{2,3}$	/	-185	/	/	/	*all*	/	/	/	/
$s_{2,2}$	/	/	/	/	*all*	/	/	/	/	-216
$s_{2,1}$	/	/	/	*all*	/	/	/	/	/	/
$s_{1,3}$	/	/	*all*	/	/	/	/	/	/	/
$s_{1,2}$	/	*all*	/	/	/	-185	/	-217	/	/
$s_{1,1}$	*all*	/	/	/	/	/	-153	/	/	/
	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$s_{4,1}$	$s_{4,2}$	$s_{4,3}$	$s_{4,4}$

We adopt an *AR matrix* to model the AR between various services, which shows symmetrical structure. Table I presents an example *AR matrix* based on Fig.2. Thus, it can be simplified when in use to reduce time and space overhead. Multi-attribute alliance-aware SCO problem can also utilize the AR multidimensional matrix storage, and the number of domain elements does not increase in quotient space model, our approach can also handle the multi-attribute QoS alliance problem. With regard to the example business process presented in Fig.2, each composite service contains three tasks. When trying to determine the alliance between any two services for each composite service instance, there is a need to seek for 3*3=9 times, and each search process needs to look up the relation matrix once. The search size is 10*10=100 times and the time overhead is 9*100=900 times. Now let us assume that the number of tasks is n, and the number of candidate services is m. The time complexity is $O(n^2 * m^2)$. When n or m is large, modelling the service composition with AR is computationally expensive. In order to solve this problem, this paper utilizes the theory of quotient space, and optimizes time and space overhead of query in alliance with granular computing.

Taking Fig.2 as an example, the granular result of AR is described as shown in Table II.

Then the elements in the domain are sorted by ascending order of the number of alliances contained in its structure. For a composite service instance, our algorithms search the elements in the domain reversely and match them through the services in the example. The privilege value is obtained if the match succeeds. For instance, the composition

Table II: The granular result of AR

Domain	Structure	Attribute
1	No alliance	0
2	$< s_{1,1}, s_{4,1} >$	153
3	$< s_{1,2}, s_{4,2} >$	217
4	$< s_{1,2}, s_{2,3} >$	185
5	$< s_{2,2}, s_{4,4} >$	216
6	$< s_{1,2}, s_{2,3} >, < s_{1,2}, s_{4,2} >$	402

$s_{1,2} \to s_{2,3} \to s_{4,2}$ only contains the alliance $< s_{1,2}, s_{2,3} >$ and $< s_{1,2}, s_{4,2} >$, so the privilege value can be directly obtained as 402. Although the domain elements 3 and 4 are also contained, their matching value can be determined as 6 through reverse lookup.

For any composite service instance in Table II, the maximal number of alliance queries based on quotient space is $3 * 3 * 6 = 54$, which improves by $900/54 = 16.7$ times, compared to the original alliance relation matrix. Its time complexity is $O(n^2 * k)$, where k denotes the number of elements in the domain. Due to the sparsity of alliance relation, k is much smaller than the total number of candidate services, i.e., $k \ll m^2$. Therefore, utilizing the quotient space can significantly reduce the time complexity of queries and calculation for AR.

IV. Modelling AR in Quotient Space

Quotient space is one of the three major theories of granular computing [12]. In this paper, we apply the theory of quotient space to solving the alliance-aware SCO problem. Through the hierarchical granulation thought, the service composition process is abstracted from the atomic service (basic granularity) to the task layer (function granularity), thereby obtaining the task sets (business granularity) to satisfy the business needs and then rising to the service relation layer (relation granularity) and finally obtaining the optimal alliance-aware composite service. This granulation process is shown in Fig.3

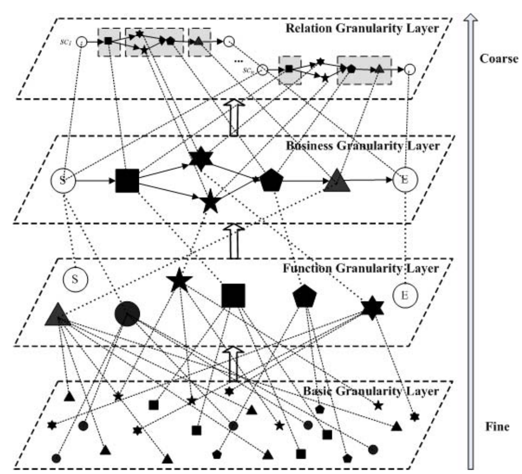

Fig. 3: The model of hierarchical granulation process for SCO problem

In this paper, we focus on considering the AR between services, constructing relation granularity layer and simplifying the procedure of finding the optimal composite service through granular computing. Based on the basic granularity, function granularity and business granularity, the construction of the AR and relation granularity quotient space model are discussed below.

A. Related Definitions and Properties of AR

Definition 1 (Quotient Space [10]): If X is a domain and R is an equivalence relation on X, then $[X]$ is a quotient set under R. Regarding $[X]$ as a new domain, we have a new domain that is coarser than X. We say that X is granulated by R. The original problem space is transformed into a new problem space at a new abstraction level. We call $([X], [f], [T])$ a quotient space of (X, f, T). We have a coarser space with a more coarse-granularity domain $[X]$, attribute $[f]$, and structure $[T]$, called the quotient set, quotient attribute, and quotient structure, respectively.

Definition 2 (Basic Granularity, BaG): The basic granularity is the smallest granularity for the SCO problem (i.e., the atomic service), which can be denoted as a five-tuple $s = (id, I, desc, O, Q)$, where id represents the unique identification of each granularity, I represents the input parameters, $desc$ represents the functional description, O represents the output parameters and Q represents the QoS attribute of each granularity.

Definition 3 (Function Granularity, FG): Function granularity is the clustering of basic granularities possessing function attributes, which corresponds to a specific task in the SCO problem. Function granularity contains basic granularity and is described in a four-tuple $t = (S, I, desc, O)$. Where S represents the basic granularity set, I represents the input parameters, $desc$ represents the functional description and O represents the output parameters.

Definition 4 (Business Granularity, BuG): The business granularity is the clustering set of the function granularities that fulfil the users' business requirements for a business process, denoted by a three-tuple $CS = (desc, T, fitness)$, where $desc$ represents the description of the business process. T represents the function granularity set for achieving a certain business process, and $fitness$ represents the evaluation function of the service granularity.

Definition 5 (Alliance Relation, AR): Alliance Relation refers to the collaboration between the basic granularity in the business granularity, which is described by a three-tuple $AR = (s_n, s_m, h)$, where s_n denotes the predecessor basic granularity of the AR in the business granularity, s_m denotes the subsequent basic granularity, and h denotes the privilege function between s_n and s_m. The privilege function is defined as $h : h(s_n, s_m) = \chi$. $h(s_n, s_m)$ is denoted as χ, also called privilege policy. $h(s_n, s_m) = \emptyset$ is recorded as the zero relationship and $h(s_n, s_m) = all$ is recorded as the 1 relationship.

For the SCO problem, the predecessor basic granularity and subsequent basic granularity represent the predecessor service and subsequent service of the AR in the selected composite services instance. In particular, for $AR = (s_n, s_m, h)$, $Serv(s_n, s_m, h)$ is used to represent the service matrix between logical positions of s_n and s_m (including the boundary) in service plan for convenience. $|AR| = |(s_n, s_m, h)|$ denotes the rank of matrix $Serv(s_n, s_m, h)$. In composite service instance, it indicates the number of the selected services between s_n and s_m in the composite service.

According to the above definition, assume that a service composition instance sc contains three basic granularities including s_n, s_m, s_k and there exist the following properties.

Property 1 (Symmetry): For any AR, $AR = (s_n, s_m, h) \iff (s_m, s_n, h)$ is satisfied.

Property 2 (Transmissibility): If there are $AR_1 = (s_n, s_m, h), AR_2 = (s_m, s_k, h)$, the business granularity s_n and s_k have indirect AR, denoted as $AR = (s_n, s_k, h)$.

Property 3 (Reflexivity): For any service s_n, there is always its own alliance, i.e., $AR = (s_n, s_n, all)$ is permanent.

B. Relation Granularity Quotient Space

Based on the AR, for any composite services instance cs, if all the alliances with s_n as the predecessor service satisfy $|(s_n, s_j, h)| \geq |(s_n, s_i, h)|$, (s_n, s_j, h) can be called the *Domination Relation* (DR) of service s_n, denoted as $DR = (s_n, s_j, h)$. Through DR, each service composition instance of business granularity can be divided into different modules and each module belongs to and only belongs to the covering domain of one DR. In the same coverage, all basic granularities are equivalent and the privilege policy of DR includes all the internal AR. Through the properties and the definition of DR above, the equivalence class R can be obtained. For the services contained in DR, after adding a special alliance to any two services s_n and s_m without alliance, i.e., zero relationship, the set satisfies the equivalence relation and belongs to the same equivalence class.

Definition 6 (Relation Granularity Quotient Space): A service composition instance in business granularity can be divided by equivalence relation R into relation granularity quotient space, denoted as $([X], [f], [T])$, where $[X]$ denotes the equivalence class set of service composition instances divided by DR, $[f]$ denotes the privilege policy of AR divided by equivalence relation R, $[T]$ represents the topological relation between the selected basic granularities divided by DR in the SCO domain and $[T]$ is the topology between the services in the service composition instance. The relation granularity quotient topology can be expressed as $[T] : \{u|p_1^{-1}(u) \in T, u \in [T]\}$, where p_1 represents the natural projection, denoted as $p_1 : T \to [T]$.

In the relation granularity quotient space, the domain is the subset of the power set of AR. When the domain scale is equivalent to the number of AR, there is a one-to-one correspondence between the domain elements and the AR.

This paper mainly discusses the SCO problem based on AR and simplifies the complexity of query and calculation of AR using the relation granularity quotient space.

In the theoretical system of quotient space, the relation granularity quotient space should fulfil the falsity preserving principle and the truth preserving principle [11] to ensure the completeness of the problem solving process.

Theorem 1 (Falsity Preserving Principle): For any composite service instance sc, if it is the optimal solution to the SCO problem (X, f, T) on the BaG layer, it is also the optimal solution in the relation granularity quotient space $([X], [f], [T])$. It can be also described that, if sc is not the optimal solution in the relation granularity quotient space $([X], [f], [T])$, it is neither the optimal solution to the SCO problem on the BaG layer (X, f, T).

Proof: given that sc is the optimal solution in (X, f, T), it is assumed that it contains $AR : r = (s_i, s_j, h)$ and $p : T \rightarrow [T]$ is recorded as the partition from the basic granularity quotient space to relation granularity quotient space. Then the mapping of r, $p(R)$ is in relation granularity quotient space $[T]$. Therefore, sc is also the optimal solution in the relation granularity quotient space $([X], [f], [T])$. ■

Theorem 2 (Truth Preserving Principle): It is assumed that the composite service instance sc is the optimal solution in relation granularity quotient space $([X], [f], [T])$ and for any $AR : [r]$, $p^{-1}([r])$is a connected set of T. Then sc is also the optimal solution to the SCO problem on the BaG layer (X, f, T).

Proof: given that sc is the optimal solution in $([X], [f], [T])$, it is assumed that it contains $AR : r = (s_i, s_j, h)$. Then for the mapping $p : [T] \rightarrow T$, it can be obtained that $p^{-1}([r])$ belongs to the basic granularity quotient space (X, f, T). In the basic granularity quotient space sc also has $AR : p^{-1}([r])$, and is connected. Therefore, sc has the same AR in the basic granularity quotient space, and is still the optimal solution. ■

V. ASC_QS Approach

This section presents our ASC_QS approach based on AR and the relation granularity quotient space, and finally solves the SCO problem using the RG-PSO algorithm.

A. Construction of relation granularity quotient space

In the relation granularity quotient space, the core task is the acquisition of equivalent classes. The detailed construction process is described by Algorithm 1.

In Algorithm 1, $sc.count$ denotes the length of the business process, i.e., the number of tasks. $rm[i][j]$ denotes the element on the ith row and the jth column in the relationship matrix, i.e., the privilege value between service i and service j. If the value is -1 which indicates no AR, a 0 relationship is added. $maxLength$ indicates the subscript of the subsequent service of DR. The result of Algorithm 1 is not only used to build the relation granularity quotient space, but can also to obtain the equivalence relation which is the matching basis for solving the AR for the composite service instance.

Algorithm 1 Relation granularity quotient space

Input: business granularity sc and relation matrix rm
Output: the element of relation granularity quotient space

for each $i \leq sc.count - 1$ **do**
  initialize $maxlength \leftarrow 0$
  for $j = sc.count; i \leq j - 1; j - -$ **do**
    $Querry\ maxlength$;
  end for
  for $j \leftarrow i + 1; j \leq maxlength; j + +$ **do**
    if $rm[i][j] == -1$ **then**
      $rm[i][j] = 0$; $//add\ the\ zero\ relationship$
    end if
    for $k \leftarrow j + 1; k \leq sc.count; k + +$ **do**
      $Keep\ maxlength = max(maxlength, k)$;
    end for
  end for
  $Calculate\ \gamma$;
  $rm[i][maxlength] = \gamma$;
  $//jump\ to\ the\ next\ equivalence\ class$
  $i = maxlength + 1$;
end for

B. A Case Study

This subsection presents a case study to illustrate the construction of the relation granularity quotient space and the query and calculation process of the optimized AR.

The shopping case shown in Fig.1 is abstracted to a composite service instance sc, which contains three relation granularities (s_1, s_3, h), (s_2, s_4, h) and (s_6, s_8, h), where $h(s_1, s_3)$ is the self-support privilege value of the E-business platform, $h(s_2, s_4)$ is the discount rate between E-business platform service providers and the payment methods and $h(s_6, s_8)$ is the influence value between the type of packaging and the mode of distribution. The corresponding business granularity is shown in Fig.4.

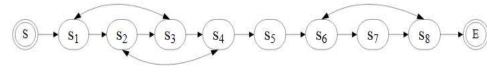

Fig. 4: Business granularity before AR processing

According to the construction method of the equivalent relation R, the following processing is carried out based on Fig.4.

Step1: Find the DR and the current $DR = (s_1, s_3, h)$ is obtained;

Step2: Find service s_2 contained in $Serv(DR)$. It is easy to determine that there is no AR between s_2 and DR predecessor service s_1. Such services are referred to as orphaned services. If they are isolated from each other, go to Step3. Otherwise, Step4.

Step3: Add a zero relationship between s_1 and s_2, i.e., add the relation granularity $AR = (s_1, s_2, \emptyset)$, the intermediate process can then be obtained as shown in Fig.5.

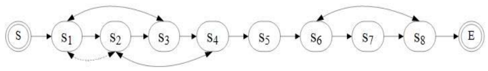

Fig. 5: Business granularity after adding zero relationship

Step4: The DR is expanded through *Property 2* and is updated as $DR = (s_1, s_4, h)$, shown in Fig.6.

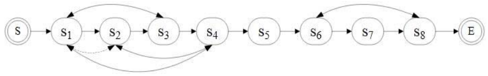

Fig. 6: Updating the business granularity of DR

Step5: Repeat the process above until all services are processed. The results are shown in Fig.7.

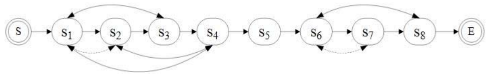

Fig. 7: The business granularity to be divided

In the business process above, all the domination relations of this business can be concluded to $DR_1 = (s_1, s_4, h)$, $DR_2 = (s_5, s_5, h)$ and $DR_3 = (s_6, s_8, h)$. According to the DR, the corresponding equivalence classes are constructed as shown in Fig.8.

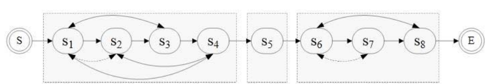

Fig. 8: Business equivalence classes

From the above process it can be seen that, during the construction of the relation granularity, granularity coarsens continuously and is converted from the previous business granularity composed of eight function granularities to the business granularity composed of three relation granularities. As a result, its scale is reduced greatly. Through adding the generated equivalence classes to the relation granularity quotient space domain, the privilege value of AR can be obtained directly. Thus the complexity of the query and computation of AR is significantly reduced.

C. RG-PSO

Based on the established relation granularity quotient space, through the equivalence classes of DR we can: 1) search each equivalence class on the business granularity layer of the composite service; 2) obtain the privilege value of its DR; 3) summarize them to obtain the privilege value of each composite service instance in the business granularity, i.e., to utilize the privilege value of AR to calculate the privilege value of service instance in business granularity, thereby simplifying the lookup process of AR privilege value. Next, our RG-PSO algorithm calculates the fitness through relation granularity quotient space, selects the elitists and obtains the optimal solution by successive iteration. The RG-PSO algorithm is illustrated in Algorithm 2. The method for fitness calculation is shown as Eq.1 on the basis of relation granularity:

$$fitness = \sum_{i=1}^{n} c_i (Cost_i - h) \quad (1)$$

where c_i represents the weight of each QoS attribute, $Cost_i$ represents the $Cost$ attribute of the ith candidate service and h represents the alliance privilege function.

Algorithm 2 RG-PSO

Input: business quotient space and relation quotient space
Output: the optimal composite service in the BuG

```
Initialization parameters: swarm size sizepop, the number of iterations iter, the number of subtasks n and the optimal service granularity g[n]
Generate x[sizepop, n] and v[sizepop, n]
for each t ≥ iter do
    Initialize pbest[sizepop] and p[sizepop, n];
    for each i ≤ sizepop do
        Calculate individual fitness value: fit[i]
        [pbest[i] p[i, j]] = min(fit[i], pbest[i]), j ∈ [1, n];
        [gbest g[j]] = min(pbest[i], gbest), j ∈ [1, n];
    end for
    Update c1, c2, ω;
    for each i ≤ sizepop do
        Update v[i, j], x[i, j], j ∈ [1, n];
    end for
end for
Output gbest and g[j], j ∈ [1, n]
```

In algorithm 2, where $min(fit[i], pbest[i])$ represents the minimum value of $pbest[i]$ and $fit[i]$, and the minimum granularity $p[i, j]$. The operation scheme of traditional PSO is adopted to update the learning factor, inertia weight, granularity velocity and the position of granularity [4].

VI. Experiments and Analysis

In order to evaluate the effectiveness of RG-PSO algorithm, we carried out four sets of experiments. The first set evaluated the influence of AR. The second set evaluated the $improve\_rate$ of fitness by comparing our approach with several representative existing approaches. The third set evaluated the efficiency of our approach. The fourth set of experiments evaluated the scalability of our algorithms by observing the dispersion degree of the results using the root mean square error (RMSE).

We realized the proposed algorithms using C#. Experiments were implemented on a computer with Intel Dual-Core CPU i5 3.2GHZ, 8G of RAM, and Windows 7.

A. Datasets

In the experiment, we used two date sets, the QWS data set [13] and the random data set (RWS). The RWS data set was generated based on $N(0, 1)$ normal distribution [4]. Assuming that the task submitted by users can be divided into 5 subtasks.

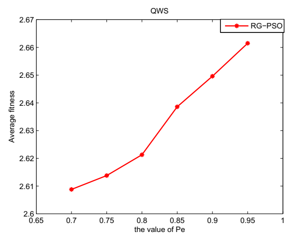

Fig. 9: The influence of Pe

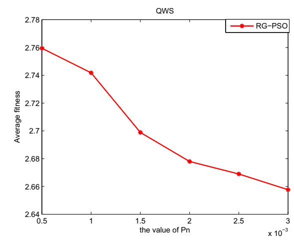

Fig. 10: The influence of Pn

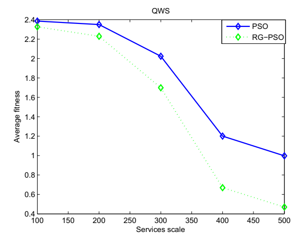

Fig. 11: QWS's fitness

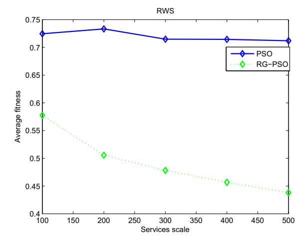

Fig. 12: RWS's fitness

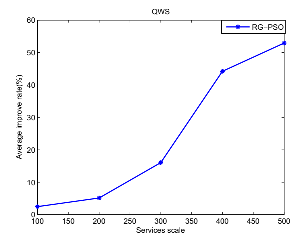

Fig. 13: QWS's improve_rate

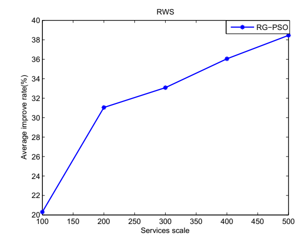

Fig. 14: RWS's improve_rate

B. The influence of AR

We analyze the influence of the proportion of AR on the service composition by using the alliance existing threshold and the number threshold of AR. The alliance existing threshold is a measure of the probability of alliance existence of each candidate service, denoted as Pe, In this experiment, the probability of the alliance existence for each candidate service is $p \in [0, 1]$, if $p > Pe$, it is considered that the service has alliance relationship. The number threshold of AR refers to the maximum ratio of the alliance number of each candidate service, which is denoted as Pn. The calculation method of the alliance number of each service is $num = \lfloor Pn * n * m \rfloor$, where, n represents the number of tasks and m represents the number of candidate services. The experiment is implemented 100 times, the average solution of optimization for QWS data are demonstrated in Fig.9 and Fig.10.

Fig.9 and Fig.10 demonstrate that the different proportions of AR have direct impacts on the optimization results. As can be seen from Fig.9, with the increase of the threshold value of Pe, the fitness value increases gradually, which indicates that the experimental result is gradually getting worse. This is because a greater the threshold Pe results in less alliance. As can be seen from Fig.10, with the increase of the number of Pn threshold, the fitness value gradually becomes smaller. This is because a greater threshold Pn results in more ARs. Therefore, with the increase in the number of ARs, the optimization result is better, so the AR has an important influence on the service composition.

C. Effectiveness

To validate the solvability of alliance-aware SCO model based on the relation granularity quotient space, the effectiveness of RG-PSO was evaluated in terms of the fitness value and the improve rate when the number of candidate services set varied from 100 to 500. The calculation of the $improve\_rate$ is shown as follow [14]:

$$improve\_rate = \frac{fitness_{old} - fitness_{new}}{fitness_{old}} \tag{2}$$

where $fitness_{old}$ represents the fitness value without taking into account the AR and $fitness_{new}$ represents the fitness value considering the AR.

Fig. 11 and Fig. 12 compare RG-PSO and an existing PSO algorithm [20]. From the varying trend of the curves, it can be seen that RG-PSO can obtain a smaller fitness value. Furthermore, the gap between RG-PSO and PSO becomes larger as the number of the candidate services increases. In particular, in the RWS data set, without considering AR, PSO cannot obtain an optimal solution under the same condition. It is because each service in the RWS data set was generated randomly. Although the overall result changes in according with the normal distribution, the internal link between services is little and PSO could not obtain the optimal solution. With the introduction of AR, the relation between services becomes tighter and the result of RG-PSO is significantly better than that of PSO. To describe the optimization difference between RG-PSO and PSO more intuitively, Fig.13 and Fig.14 demonstrate the $improve\_rate$ of RG-PSO against PSO. Moreover, it can be observed that the $improve\_rate$ with the increase in the number of the candidate services. The size of $improve\_rate$ is affected by the ratio of alliance and privilege.

D. Efficiency

To evaluate the efficiency of RG-PSO, we changed the number of candidate services from 100 to 500 in steps of 100. Fig.15 and Fig.16 compare the time overhead between RG-PSO and PSO based on the QWS and RWS data set respectively. The time complexity of RG-PSO not only is influenced by the number of candidate services, but also the alliance ratio. It is because a greater the alliance, leads to a larger query time overhead of the privilege relation in the composite services instance. The query and calculation of AR also effect the time overhead of RG-PSO. The results demonstrate that as the number of the candidate services increases, the growth rate of the time overhead of RG-PSO is higher than that of PSO, however always within two seconds in all cases. Considering the advantage of RG-PSO in optimization over PSO, slight increases in time consumption are acceptable. In addition, there are two ways

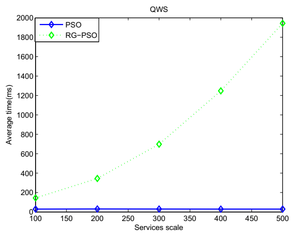

Fig. 15: QWS's time cost

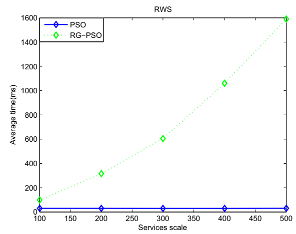

Fig. 16: RWS's time cost

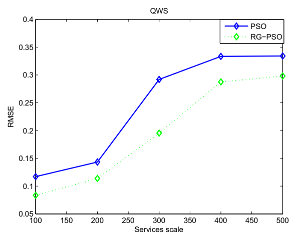

Fig. 17: QWS's RMSE

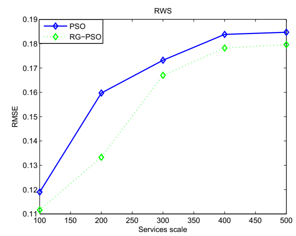

Fig. 18: RWS's RMSE

to build the relation granularity quotient space. The first one is synchronous construction, and the other is asynchronous construction. In order to verify the maximum time overhead of RG-PSO, we adopted the dynamic construction scheme and measured the time overhead of constructing the relation granularity quotient space during the optimization. The total time consumption is no more than two seconds.

E. Scalability

To evaluate the scalability of RG-PSO, we increased the number of candidate services from 100 to 500 and calculated the RMSE, which is calculated using formula (3):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}} \tag{3}$$

where $\bar{x}$ represents the average fitness, x_i represents the ith optimal fitness.

From Fig. 17 and Fig. 18, we can see that, RG-PSO is more scalalbe than PSO in solving the alliance-aware SOC problem. With the increase in the number of candidate services, the increase in the RMSE of RG-PSO becomes slower, and finally, stabilizes. It indicates that the more candidate services, the more stable RG-PSO.

VII. Conclusion and Future Work

In this paper, we proposed a novel method, named ASC_QS, for solving the problem of alliance-aware service composition. This method is the first one that systematically considers the AR between services. Through the truth preserving principle and the falsity preserving principle, on the basis of ensuring the completeness of relation granulation, ASC_QS can greatly improve the quality of the composite services by taking into account the ARs between services. Our experiments show that our method based on relation granulation quotient space is effective, efficient and scalable. For the purpose of simplicity, we only considered one QoS dimension when solving the SCO problem. In our future work, we will investigate the alliance-aware SCO problem based on multiple dimensional QoS.

Acknowledgment

This work was supported by the National Key Technology R&D Program (No.2015BAK24B01), the Natural Science Foundation of Anhui Province of China (No.1408085MF132) and the Educational Commission of Anhui Province of China (No.KJ2016A038).

References

[1] Y. Ma, S. Wang, P.C.K. Hung et.al, "A Highly Accurate Prediction Algorithm for Unknown Web Service QoS Values," *IEEE Transaction on Service Computing*[online]. DOI 10.1109/ TSC.2015.2407877.

[2] Q. He, X. Xie, F. Chen, et.al, "QoS-Aware Service Selection for Customisable Multi-Tenant Service-Based Systems: Maturity and Approaches," *Proc. of IEEE International Conference on Cloud Computing (Cloud2015)*. New York, USA, 2015, pp.237-244.

[3] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient web service composition with end-to-end qos constraints," *ACM Trans. Web*. vol.6, no.2, pp.1-31, 2012.

[4] Y. Zhang, G. Cui, S. Zhao et.al, "IFOA4WSC: A quick and effective algorithm for QoS-aware service composition," *International Journal of Web and Grid Services*. vol.12, no.1, pp.81-108, 2016.

[5] V. Uc-Cetina, F. Moo-Mena, R. Hernandez-Ucan, "Composition of Web Services Using Markov Decision Processes and Dynamic Programming," *The Scientific World Journal*. 2015.

[6] V. Gabrel, M. Manouvrier, and C. Murat, "Optimal and automatic transactional web service composition with dependency graph and 0-1 linear programming," *in Proceedings of the 12th Intl.Conf. on Serv. Ori. Comp*. Springer, 2014, pp.108-122.

[7] F. Wagner, A. Klein, B. Klopper, et.al, "Multi-objective Service Composition with Time-and Input-Dependent QoS," *In Web Services (ICWS), 2012 IEEE 19th International Conference on*. 2012, pp.234-241.

[8] S. Deng, H. Wu, D. Hu, J.L Zhao, "Service Selection for Composition with QoS Correlation," *IEEE Trans on Services Computing*[online]. DOI: 10.1109/TSC.2014.2361138.

[9] Q. Wu, Q. Zhu, M. Zhou, "A correlation-driven optimal service selection approach for virtual enterprise establishment," *J Intell Manuf*. vol.25, pp.1441-1453, 2014

[10] S. Zhao, L. Zhang, X Xu, et.al, "Hierarchical description of uncertain information," *Information Sciences*. vol.268, pp.133-146, 2014

[11] L. Zhang, B. Zhang, "The quotient space theory of problem solving," *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer Berlin Heidelberg, pp.11-15, 2003

[12] Y. Zhang, L. Zhang, C. Xu, "The Property of Different Granule and Granular Methods Based on Quotient Space," *Information Granularity, Big Data, and Computational Intelligence*. Springer International Publishing, pp.171-190, 2015

[13] E. Al-Masri, and Q.H Mahmoud, "Discovering the best web service," *The 16th International Conference on World Wide Web (WWW)*. Banff, Canada, 2007, pp.1257-1258

[14] S. Deng, H. Wu, D. Hu et.al, "Mobility-Enabled Service Selection for Composite Services," *IEEE Trans on Services Computing*[online]. DOI: 10.1109/TSC.2014.2365799.

[15] P. A. Bonatti and P. Festa, "On Optimal Service Selection," *Proc of 14th International Conference on World Wide Web (WWW)*. Chiba, Japan, 2005, pp.530-538.

[16] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-Aware Service Selection for Service-based Systems Based on Iterative Multi-Attribute Combinatorial Auction," *IEEE Transactions on Software Engineering (TSE)*. vol.40, no.2, pp.192-215, 2014.

[17] R. Gulati, M. Sytch, P. Mehrotra, "Breaking up is never easy: Planning for exit in a strategic alliance," *California Management Review*. vol.50, no.4, pp.147-163, 2008.

[18] M. Chen and Y. Yan, "QoS-aware Service Composition over Graphplan through Graph Reachability," *IEEE International Conference on Services Computing (SCC)*. Anchorage, AK, USA, 2014, pp. 544-551

[19] G. Canfora, M. Di Penta, R. Esposito, et.al., "An approach for QoS-aware service composition based on genetic algorithms," *Proceedings of the 7th annual conference on Genetic and evolutionary computation. ACM*. Washington, DC, USA, 2005, pp.1069-1075

[20] S. Wang, Q. Sun, H. Zou, et.al., "Particle swarm optimization with skyline operator for fast cloud-based web service composition," *Mobile Networks and Applications*. vol.18, no.1, pp.116-121, 2013.