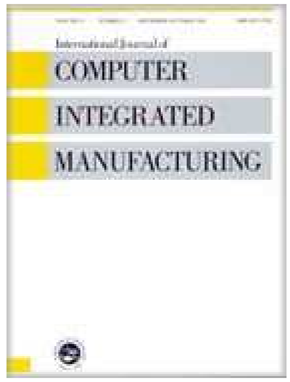




# A Trust-Based Approach to Selection of Business Services


| | |
|---|---|
| Journal: | *International Journal of Computer Integrated Manufacturing* |
| Manuscript ID: | Draft |
| Manuscript Type: | Original Manuscript |
| Date Submitted by the Author: | n/a |
| Complete List of Authors: | Li, Sufen; Tsinghua University, Automation<br>Fan, Yushun; Tsinghua University, Automation<br>Li, Xitong; MIT, Sloan School of Management |
| Keywords: | SERVICE MANAGEMENT, SERVICE QUALITY |
| Keywords (user): | services selection, trust |

# A Trust-Based Approach to Selection of Business Services

Sufen Li, Yushun Fan, Xitong Li
(Department of Automation, Tsinghua University, Beijing 100084, China)
(MIT Sloan School of Management, Cambridge, MA 02142, USA)

**Abstract:** With the increasing popularity of the service-oriented thinking in the business area, an overwhelming number of business services have arisen. These business services collaborate with each other and form a business network or a services-oriented business ecosystem (SOBE). It is a key issue to select appropriate business services to share and integrate them in the SOBE. Trust degrees play an important role in the selection of business services. However, existing researches mainly focus on QoS-based selection methods and few of them take trust degrees into account. In order to select services more accurately, a trust-based approach to selection of business services is proposed based on the formal definition of the service-oriented business ecosystem. First, the method for acquisition and calculation of trust values is developed. The method also considers other parameters including QoS attributes, the physical distance between users and business services, and the waiting time. Second, one node in a business flow may need to select two or more business services to achieve the user's requirements. Thus, the multi-service selection method for one node is presented. Third, a fuzzy chance-constrained programming model is proposed for the selection of business services by considering four kinds of factors: QoS attributes, trust relation, physical distance and waiting time. Using the characteristic of fuzzy constraints, a function is developed to convert the fuzzy chance constraints into its crisp equivalents and then the equivalent crisp model is solved by the GA-based algorithm. For the case in which no satisfying solution can be found by the GA-based algorithm, a negotiation method is proposed to reach at a satisfying solution. Finally, a case study is conducted to demonstrate the feasibility and effectiveness of the proposed approach.

**Key words:** business services; trust; services selection; business ecosystem; fuzzy programming

## 1 Introduction

With the increasing popularity and application of service-oriented thinking in business areas, business services, as a new form of services, have emerged. Business services have some characteristics, such as large quantities, variety and varying non-function qualities. The selection of business services is often affected by the user's subjective factors and the location of business services. How to manage these business services effectively has become a key issue for resource sharing and business integration under the network environment.

A business service is a kind of products provided by business components and business services are an important feature of the business component (Fan 2008). The implementation of business services can be purely manual or (semi-)automatic processes. Business services in different enterprises form a business network or a services-oriented business ecosystem (SOBE) by dynamic combination and collaboration of them (Fan 2010, Desai *et al*. 2007, Kohlborn *et al*. 2009).

In SOBE, there are many business services which have similar business functions but different non-functional parameters, such as execution time, execution cost, reliability, location. It is thus a key issue for SOBE to select appropriate business services meeting the functional and non-functional requirements. Currently, the existing researches mainly focus on how to optimize services selection results based on QoS constraints (Menascé 2002, Menascé 2004, Li *et al*. 2007). Menascé (2002) introduced four kinds of QoS attributes including availability, security properties, response time and throughput and discussed the QoS issues from the perspectives of the user and the provider, respectively. Menascé (2006) discussed the dynamic composition of Web services and gave the method to calculate QoS attributes for composition services. Li *et al*. (2007) proposed an approach to multi-QoS constrained Web services selection. Most of the efforts in this thread focus on QoS-based services selection, but ignore the trust relationship among the services providers and consumers. The research in this paper is motivated to fill in the gap.

Trust is not an objective property of an agent but a subjective *degree of belief* about specific agents (Misztal *et al*. 1996), within a specific context, that range from complete distrust to complete trust (Gambetta, 1990). Trust plays a major role in conducting business of the internet in service-oriented environments (Azzedin *et al*. 2007). There are a large number of services belonging to the same or different enterprises in SOBE, and subjective trust relations result from various reasons between enterprises, such as friendship, kinship, or old customer relations. Subjective trust relations between users and business services often affect the result of business services selection.

In the literature only a few researches take trust degrees into account. Dai *et al*. (2009) investigated the trust-aware component service selection and established mathematical models, but they did not give a method to calculate trust. Hammadi *et al*. (2009) studied how to determine the trustworthiness of the composite service and used probability theory to determine the trustworthiness of the composite service in the composite form of parallel and series. However, probability theory is usually used to solve objective uncertain problems (Liu 2009) and inappropriate to be used to calculate trust, because trust is a subjective *degree of belief* (Misztal *et al*. 1996). The approach discussed by Azzedin *et al*. (2007) used fuzzy logic to model trust in a P2P environment. This approach requires inference rules to weight first-hand and second-hand information. It is costly and time-consuming to construct and verify the inference rules. In contrast, the trust degrees in our approach are calculated by a function according to the information in the business network. Because fuzzy set theory can help handle the subjective imprecise nature and uncertainty of trust (Liu 2009), our approach uses the fuzzy number to describe trust degrees between the user and each business services.

Meanwhile, geographical locations of business services can affect users' services selection results. For example, in case of the galvanizing business in Metals manufacturing industry, users are more willing to select the right business service located close to users' location so as to reduce transport costs and time of metals. In the current researches, the factors about geographical locations are largely ignored. The services selection model in this paper introduces two kinds of parameters including transport times and transport costs to describe the geographical locations requirements from users.

In addition, there is a general assumption that every business node in flow just needs one specific business service. However, in the actual business collaboration, a business node may need to choose two or more business services to complete the business requirements including functional and non-functional needs. For instance, the garment manufacturing process is often composed by many business nodes. The business node of "printing and dyeing" is an important node in the process which may have much workload in a certain period of time. Although there are many business services providing the service of "printing and dyeing", any single service cannot complete the workload in the required time. In such case, it is required to select two or more business services to meet customer needs. However, few of existing researches take this point into account. This paper will present a multi-service selection method and address this issue.

In general, multi-QoS constrained Web Services composition, with or without optimization, is a NP-complete problem on computational complexity and cannot be solved in polynomial time (Yu *et al*. 2005, Garey *et al*. 1979). Therefore, many heuristic and approximation algorithms with polynomial- and pseudo polynomial-time complexities have been designed to deal with this problem. As a kind of approximation algorithms, our approach uses genetic algorithm to solve the selection problem of composition services.

Furthermore, the solution set of the business services selection model might be an empty set with some given business requirements from users, which means that no result can be achieved for the present user. In this case, there is a need for business providers and the user to negotiate over various non-functional requirements to obtain an integrative (near) optimal solution that fits the needs of both parties. To date, existing researches in the field of Web services mainly focus on Web service discovery. Studies in web negotiations mostly focus on messaging and protocol languages for negotiation (Keller *et al*. 2003, Hung *et al*. 2004). Few have been done on web service negotiation processes, particularly techniques for automatically generating counter-offers during the negotiation. Patankar *et al*. (2008) presents an offset-based automated negotiation approach to support web service procurement. The approach employs an iterative tradeoff mechanism for evaluating opponent's offers and generating counter-offers of mutual gain based on selected QoS parameters.

In summary, this paper makes four important contributions as follows:

1) This paper provides a method to calculate trust degrees including the direct trust degree and the indirect trust degree. If the current user's trust degree is "unknown" for a business service, then the indirect trust degree is used in the services selection model.

2) Considering the fact that one node in a business flow may need to select two or more business services to achieve the user's requirements, a multi-service selection method for one node is presented. Our approach assumes the parallel work mode of services and gives a method to calculate the total QoS attributes and select services for one business node.

3) A fuzzy chance-constrained programming model is proposed for the selection of business services by considering four kinds of factors: QoS attributes, trust relation, physical distance and waiting time. According to the characteristic of fuzzy constraints, a function is developed to convert the fuzzy chance constraints into its crisp equivalents and then the equivalent crisp model is solved by the GA-based algorithm.

4) For the case in which no satisfying solution can be found by the GA-based algorithm, this paper presents an offset-based automated negotiation approach to support the business services selection. This approach is implemented by a dynamic negotiation process with information exchange between providers and users. The aim of the process is to find a satisfying solution for the present user.

The rest of the paper is organized as follows. The approach to business services selection is explored based on ecosystem classification mode. Section 2 defines the problems of interest. Section 3 describes the formal definitions of the service-oriented business ecosystem. Section 4 discusses the method for acquisition and calculation of trust values. Section 5 presents the multi-service selection method for one node and establishes a fuzzy chance-constrained programming model. Section 6 describes the GA-based algorithm to solve the proposed mathematical model. Section 7 develops a business services negotiation algorithm. Section 8 presents a case study and finally Section 9 presents the conclusions and future work.

## 2 Notations and the Problem statement

To describe the approach more clearly, some denotations are defined in Table 1.

Table 1 Notation Definition of the business services selection model

| | |
|---|---|
| $i$ | The node number in a business flow, $i$=1,2,3,…$I$, $I$ is a positive integer. Without loss of generality, it is assumed that the business services in population $i$ can achieve the business functional requirements of the node $i$ in a business flow. So $i$ can also denote the population number. |
| $j$ | The business service individual number in a population, $j$=1,2,3,…$J_i$, $J_i$ is a positive integer, $i$=1,2,3,…$I$ |
| $Y_{i,j}$ | 0-1variables, the decision variables, $i$=1,2,3,…$I$, $j$=1,2,3,…$J_i$. If the individual $j$ in population $i$ is selected, then $Y_{i,,j}$=1; Otherwise, $Y_{i,,j}$=0 |
| $c_{i,j}$ | The cost of the individual $j$ in population $i$ for one-time execution |
| $t_{i,j}$ | The time of the individual $j$ in population $i$ for one-time execution |
| $a_{i,j}$ | The availability of the individual $j$ in population $i$ |
| $r_{i,j}$ | The reliability of the individual $j$ in population $i$ |
| $dc_{i,j}$ | The transport cost from the location of the individual $j$ in population $i$ to the user's location |
| $dt_{i,j}$ | The transport time from the location of the individual $j$ in population $i$ to the user's location |
| $wt_{i,j}$ | The waiting time of the individual $j$ in population $i$ for the current user |
| $\tilde{tr}_{i,j}$ | The user's trust degree for the individual $j$ in population $i$ |
| $\alpha$ | The confidence level, $\alpha \in [0,1]$ |
| $C_i$ | The maximum execution cost users expected for the node $i$ in a business flow |
| $T_i$ | The maximum execution time users expected for the node $i$ in a business flow |
| $A_i$ | The minimum availability users expected for the node $i$ in a business flow |
| $R_i$ | The minimum reliability users expected for the node $i$ in a business flow |
| $DC_i$ | The maximum transport cost users expected for the node $i$ in a business flow |
| $DT_i$ | The maximum transport time users expected for the node $i$ in a business flow |
| $WT_i$ | The maximum waiting time users expected for the node $i$ in a business flow |
| $TR_i$ | The minimum trust degree users expected for the node $i$ in a business flow |
| $C_0$ | The maximum total execution cost users expected of the entire process |
| $T_0$ | The maximum total execution time users expected of the entire process |
| $WT_0$ | The maximum total waiting time of the entire process |
| $C$ | The total execution cost under a services selection scenario |
| $T$ | The total execution time under a services selection scenario |
| $WT$ | The total waiting time under a services selection scenario |

Each node in a business flow is called a business node, and each business service in every business population is called a business service individual in this paper. Business services selection is to choose business services individuals from the corresponding populations as $Pop_i$, $i = 1, 2,... I$, to meet the users' business flow needs in this paper, see in Fig 1. Users' needs may be a single business node, and also may be the form of a business process constituted by multiple nodes. For each node in the business process, it may need to select two or more business services to meet the needs, such as the need of the execution time. Factors considered in this paper are mainly QoS attributes, trust degrees, physical distance between users and business services, and waiting time of business services. These factors can affect the services selection results. The trust degree is used to describe the subjective trust relationship between the user and the business service individual, which have the feature of subjective uncertainty. Fuzzy logic can help handle the subjective imprecise nature and uncertainty of trust. We use the trapezoidal fuzzy number to describe the trust degree based on fuzzy set theory. We establish a fuzzy chance-constrained programming model for the business services selection by considering four kind factors involving QoS attributes, trust degrees, physical distance and waiting time. The model enables that two or more services can be selected for one business if it is needed.

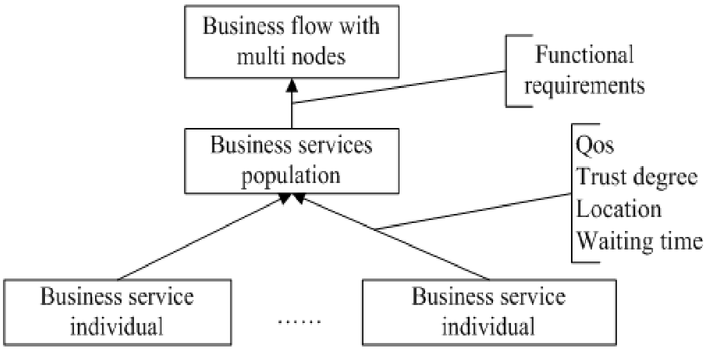


Fig. 1 The process of business services selection

## 3 Formal definitions of the service-oriented business ecosystem

Under service-oriented enterprise environment, business services from various enterprises and their surroundings together constitute a business ecosystem (Fan 2010). We call the ecosystem as service-oriented business ecosystem (SOBE) and the individuals in SOBE as business services. The SOBE is used to describe the collaborative relationship and operation mode between business services. The SOBE describes a new kind of business network including a variety of business services, which are interdependent, mutual coordination, competing, just as the individuals' activities in natural ecosystems.

The SOBE is a logical collection of business services and relationships between them. The component elements of the SOBE are mainly business service individuals and business service populations. In this section, we give the following formal definitions of the SOBE.

**Definition 1 [Business service individual]** The Business service individual is a service description entity generated by parsing the service description file from the service provider, which can be described as a triple:

$$\text{Bs} = (\text{Function}, \text{QoS}, \text{Location})$$

where $\text{Function}$ is the functional description of the individual including the business functions description and input-output description, $\text{QoS}$ is the description of an individual's quality of service which includes the service execution time, cost, availability, reliability, etc. $\text{Location}$ is the description of an individual's geographical position.

Every business service individual has certain business functions and non-functional attributes. It is the smallest business unit according to business function in the SOBE.

**Definition 2 [Business service population]** Business service population is a collection of business individuals with the same or similar business functions in the SOBE, which can be described as:

$$Pop = (Bs_1, Bs_2, \dots, Bs_q, \dots, Bs_{J_i})$$

Individuals in the same population have the same business functions, but they may have different non-functional attributes, such as cost, time, reliability and availability. Therefore, there is competition between individual business services. The value of business individuals' non-functional attributes may change with users' feedbacks or competition from other individuals.

**Definition 3 [Trust degree]** The trust degree is used to describe the subjective trust relationship between the user and the business service individual, which is defined by a set of fuzzy numbers:

$$Trust = (\tilde{tr}_{1,1}, \tilde{tr}_{1,2}, \dots \tilde{tr}_{i,j}, \dots \tilde{tr}_{I,J_i})$$

where $\tilde{tr}_{i,j}$ describes the user trust degree for the individual *j* in population *i*, which is depicted by a trapezoidal fuzzy number in this paper, that is, $(tr_{i,j}^1, tr_{i,j}^2, tr_{i,j}^3, tr_{i,j}^4)$, $tr_{i,j}^{1,2,3,4} \in [0,1]$, $i$=1,2,3,…$I$, $j$=1,2,3,…$J_i$. Different users may have different trust degrees for the same individual. And trust degrees are determined in the light of the personal relationship and historical cooperation experience between users and business service providers.

**Definition 4 [Service-oriented business ecosystem]** The Service-oriented business ecosystem is a collection of business individuals and the relations among them under certain environment, which can be described as:

$$Com = (Environment, BS, Relation, Trust)$$

where $Environment$ is the description of the environment that every individual lies in. $Relation = BS \times BS$ refers to the collection of the relations between individuals. $Trust$ has been defined in Definition 3. $BS$ is the collection of all individuals in the SOBE, which can be described as:

$$BS = (Bs_1, Bs_2, \dots, Bs_m, \dots, Bs_M)$$

Where $M = \sum_{i=1}^{I} J_i$ and $BS = Pop_1 \cup Pop_2 \cup \dots Pop_i \cup \dots Pop_I$, $i$=1,2,3,…$I$.

## 4 The acquisition and calculation methods of parameters

Four kinds of parameters are considered in the services selection model: trust degrees, QoS attributes, distances costs and waiting time.

### 4.1 Calculation methods of trust degrees

The trust degree is the parameter to describe the subjective relation between user and business service. Trust is a social phenomenon. "*Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action*" (Gambetta 1990). Trust decisions are made based on the agent's knowledge and experiences (Jøsang 1996).

Trust has the following properties.

① Trust is subjective: Trust is a user's belief in a service's behaviors, and different users may have different trust degrees on the same business service individual.

② Trust is dynamic: Trust is a dynamic concept evolving with the change of the context and time.

③ Trust is asymmetric (Sha *et al*. 2002): The event that A trust B does not means that B trust A. Even if A trust B and B trust A, the degree may be different.

④ Trust is decaying with time: Trust will decay over time. The more distant the trust evaluation, the worse its persuasive. For example, if x trusts y at level p based on past experience five years ago, the trust level today is very likely to be lower unless they have interacted since then (Azzedin *et al*. 2007).

According to the access channel, Trust can be classified into two categories: the direct trust and the indirect trust. Direct trust means that the two entities have had a direct transaction, or have other trust relations, such as friendship, kinship. Trust value generates from the user's subjective evaluation of services (Azzedin *et al*. 2007). Indirect trust is established based on other users' trust relations on the current services. The value of indirect trust can be calculated according to other users' direct trust degree. The calculation methods of trust can be evidence theory-based method (Yu *et al*. 2004, Sun 2008), fuzzy logic-based method (Song *et al*. 2005) and probability-based method (Beth *et al*. 1994, Jøsang 1999).

As trust is a subjective degree of belief about specific business services and fuzzy set theory can help handle the subjective imprecise nature and uncertainty of trust, fuzzy number will be used to describe the trust degree in this paper. We propose a model for determining trustworthiness of services for the current user based on the direct experience (direct trust) and the recommendations from other users (indirect trust). The corresponding evaluating approach is described in the following sections.

① Evaluating direct trust

The current user's trust degree of business services individuals is quantified by the dynamic parameter trust level that ranges over a set of linguistic label values from very untrustworthy to very trustworthy as illustrated in Table 2. For each trust level, we associated a trapezoidal fuzzy number that enables us to specify a range for a given trust level instead of giving it a particular discrete value. A trapezoidal fuzzy number is defined as a quadruple $DL = (dl^1, dl^2, dl^3, dl^4)$, where $dl^{1,2,3,4} \in [0,1]$ and $dl^1 \leq dl^2 = dl^3 \leq dl^4$.

Table 2 The mapping relations between trust levels and trapezoidal fuzzy numbers

| number | trust level | trapezoidal fuzzy number( $DL_{i,j}$ ) |
|---|---|---|
| 1 | very untrustworthy | (0, 0.1, 0.1, 0.2) |
| 2 | untrustworthy | (0.1, 0.2, 0.2, 0.3) |
| 3 | common | (0.3, 0.5, 0.5, 0.7) |
| 4 | trustworthy | (0.6, 0.7, 0.7, 0.8) |
| 5 | very trustworthy | (0.8, 0.9, 0.9, 1.0) |
| others | unknown | (0.5, 0.5, 0.5, 0.5) |

② Evaluating indirect trust

The indirect trust degree is calculated by aggregating the other users' direct trust degrees, and the function (1) is the proposed calculation function.

$$IL_{i,j} = \sum_{l=1}^{L_{i,j}} \omega_{l,i,j} \, DL_{l,i,j} \qquad (*)$$

where $DL_{l,i,j}$ denotes the trust evaluation for individual *j* in population *i* by user *l*, and $DL_{l,i,j} \neq (0.5,0.5,0.5,0.5)$, which is to say that all the other users' direct trust degrees participating in the calculation of the indirect trust degree cannot be "unknown". $L_{i,j} = max(N, L'_{i,j})$, *N* is a positive integer, $L'_{i,j}$ denotes the total number of users with $DL_{l,i,j} \neq (0.5,0.5,0.5,0.5)$ for the individual *j* in population *i*.

The next work is to determine $\omega_{l,i,j}$ in the function (*) which is critical for evaluating indirect trust. Based on the trust properties described above, trust will decay over time. Related research show decay as a power function of time in social relations (Burt 2000). We describe the time-decay function by the following regression equation:

$$f(t) = (t+1)^{\gamma}$$

where *t* means a time period and $\gamma$ is a constant. Let

$$\omega'_{l,i,j} = f(t_{l,i,j} - \tau_{l,i,j}) \times E(DL_{l,i,j})$$

where $\tau_{l,i,j}$ denotes the current time, and $t_{l,i,j}$ denotes the trust evaluation time for individual *j* in population *i* by user l. *f*(·) is a time-decay function as defines above. And *E*(·) means the mathematical expected value of the fuzzy number. According to Liou *et al*. (1992), the expected value of the trapezoidal fuzzy number $DL_{l,i,j} = (DL^1_{l,i,j}, DL^2_{l,i,j}, DL^3_{l,i,j}, DL^4_{l,i,j})$ can be calculated by the following equation:

$$E\left(DL_{l,i,j}\right) = \lambda \frac{DL^1_{l,i,j} + DL^2_{l,i,j}}{2} + (1-\lambda) \frac{DL^3_{l,i,j} + DL^4_{l,i,j}}{2}, \quad 0 \leq \lambda \leq 1$$

Then, we can get the weight by the following equation:

$$\omega_{l,i,j} = \frac{\omega'_{l,i,j}}{\sum_{l=1}^{L_{i,j}} \omega'_{l,i,j}}, \quad \sum_{l=1}^{L_{i,j}} \omega_{l,i,j} = 1$$

③ The trust degree application approaches in services selection model

There are three cases to apply the trust degree in services selection model.

**Case1**: If users want to use the direct trust degree only, then $\tilde{tr}_{i,j} = DL_{i,j}$.

**Case2**: If $DL_{l,i,j} \neq (0.5,0.5,0.5,0.5)$, then $\tilde{tr}_{i,j} = DL_{i,j}$.

If $DL_{l,i,j} = (0.5,0.5,0.5,0.5)$, then $\tilde{tr}_{i,j} = IL_{i,j}$.

**Case3** (Azzedin *et al*. 2007) : $\tilde{tr}_{i,j} = \beta_1 DLi,j + \beta_2 IL_{i,j}$ where $\beta_1 + \beta_2 = 1$.

In this paper, we use the case 2, that is, if the current user's trust level is "unknown" for the individual *j* in population *i*, then the indirect trust degree will be used. Otherwise, the direct trust degree will be used.

### 4.2 The acquisition methods of other parameters

The methods for calculating trust-related parameters have been described in section 4.1, then the acquisition methods of other parameters including QoS attributes, distance costs and waiting time are introduced in this section.

For QoS attributes, we mainly focus on the service's execution time, execution cost, availability and reliability. The value of these parameters can be found from the Business service individual description which has been defined by $\mathrm{Bs} = (\mathrm{Function}, \mathrm{QoS}, \mathrm{Location})$ in Section 3**.**

Physical distances will cause the distance-related costs and distance-related time, such as transport costs and transport time, which are determined by the location of business services and the user. The value of this parameter can also be obtained from the Business service individual description file.

For waiting time, the parameter values can be determined by the service's one-time execution time and the current operating state. The former one, as one of the QoS attribute, can be found from the Business service individual description file. For the later one, the operational status of business services include idle and busy. If it is idle, the waiting time will equal to zero. If it is busy, the state will be analyzed by two cases: no queue or have a queue. If it is busy without queue, let the waiting time equal to the service's one-time execution time. If it is busy with a queue and the queue length is $q$, let the waiting time equal to $q$+1 multiple the service's one-time execution time.

## 5 The business services selection model

### 5.1 The method to select multiple business services for one business node

In the current research for services selection method, there is a general assumption that every business node in flow just need to select one specific business service in the research of services selection. However, in actual business collaboration, a business node may need to choose two or more business services to complete the business requirements including functional and non-functional needs. For instance, in Garment manufacturing processes composing by many business nodes, the business node of "printing and dyeing" is an important node which may have much workload in a certain period of time. Although there are many business services providing "printing and dyeing" services, any one service cannot complete the workload in the required time. In this case, it is required to select two or more business services to meet customer needs. This case is very common in clusters of Zhejiang in China. It is needed to develop a method to select multiple business services for one business node, and these services will work in parallel mode. Therefore, we propose the approach to solve this problem in this section.

To describe the approach more clearly, some denotations are defined in Table 3.

Table 3 Notation Definition of multiple business services selection method

| | |
|---|---|
| $c_i$ | The total execution cost for business node *i* with one or multiple selected individuals |
| $t_i$ | The total execution time for business node *i* with one or multiple selected individuals |
| $a_i$ | The total execution availability for business node *i* with one or multiple selected individuals |
| $r_i$ | The total execution reliability for business node *i* with one or multiple selected individuals |
| $dc_i$ | The total distance cost for business node *i* with one or multiple selected individuals |
| $dt_i$ | The total distance time for business node *i* with one or multiple selected individuals |
| $wt_i$ | The total waiting time for business node *i* with one or multiple selected individuals |
| $\tilde{tr}_i$ | The total trust degree for business node *i* with one or multiple selected individuals |

where $i$=1,2,3,…$I$, $I$ is a positive integer.

The implementation process of the proposed approach is depicted in the following steps.

Step1: Estimate the average execution time of services in each population *i*. If the average execution time is greater than *m* times and less than (*m*+1) times of the maximum expected execution time ($T_i$) by the user for the node *i* in a business flow, *m* business services will be chosen from the population *i* for the node *i*. The *m* business services will work in parallel mode to reduce the execution time.

Step2: Calculate parameters' values for *m* parallel services. The corresponding parameters will be calculated by the following equations.

① The execution time

For $j=J_i$, If $Y_{i,j} \neq 0$

$$\text{Let } t_i' = \sum_{j=1}^{J_i} 1/(t_{i,j} Y_{i,j}), \text{ Then } t_i = 1/t_i', \quad i=1,2,3,\ldots I$$

② The execution cost

$$c_i = \sum_{j=1}^{J_i} c_{i,j} Y_{i,j}, \ i=1,2,3,\ldots I$$

③ Availability

$$a_i = min(a_{i,j} Y_{i,j}, j = 1,2,3, \ldots J_i, Y_{i,j} \neq 0), \ i=1,2,3,\ldots I$$

④ Reliability

$$r_i = min(r_{i,j} Y_{i,j}, j = 1,2,3, \ldots J_i, Y_{i,j} \neq 0), \ i=1,2,3,\ldots I$$

⑤ The distance time

$$dt_i = max(dt_{i,j} Y_{i,j}, j = 1,2,3, \ldots J_i, Y_{i,j} \neq 0), i = 1,2,3, \ldots I$$

⑥ The distance cost

$$dc_i = \sum_{j=1}^{J_i} dc_{i,j} Y_{i,j}, \ i=1,2,3,\ldots I$$

⑦ The trust degree

$$\tilde{tr}_i = \wedge(\tilde{tr}_{i,j} Y_{i,j}, \quad j = 1,2,\ldots,J_i, Y_{i,j} \neq 0), \ i=1,2,3,\ldots I$$

⑧ The waiting time

$$wt_i = max(wt_{i,j} Y_{i,j}, j = 1,2,3, \ldots J_i, Y_{i,j} \neq 0), i = 1,2,3, \ldots I$$

**5.2 The business services selection mathematical model**

Goal programming is a branch of multi-objective optimization and is an optimization programe to handle multiple, normally conflicting objective measures. Each of these measures is given by a goal or target value to be achieved. Unwanted deviations from this set of target values are then minimized in an achievement function. This can be a vector or a weighted sum dependent on the goal programming variant used (Wikipedia).

The total execution time and cost of a business flow are two key indicators concerned by users and the total waiting time can influence the decision of services selection. We will take the total execution time, the total execution cost and the total waiting time as object elements of the model.

A trust-based business services selection model are proposed by using the goal programming method in this section. The goal is to minimize the differences between the total execution time, the total execution cost, the total waiting time and the corresponding expectation values of the user in lexicographic order. The constraints are functions of QoS attributes, trust degrees, distance costs and waiting time, respectively. The proposed trust-based business services selection fuzzy programming model (RBBSSFPM) are described as functions (1)-(13).

$$lex\min(\mathrm{d}_c^+ \vee 0, \mathrm{d}_t^+ \vee 0, \mathrm{d}_{wt}^+ \vee 0) \quad (1)$$

S. t.

$$\mathrm{d}_c^+ = C - C_0 \quad (2)$$

$$\mathrm{d}_t^+ = T - T_0 \quad (3)$$

$$\mathrm{d}_{wt}^+ = WT - WT_0 \quad (4)$$

$$wt_i \leq WT_i, \quad i = 1,2,3, \dots I \quad (5)$$

$$Cr\{\tilde{tr}_i \geq TR_i\} \geq \alpha, \quad i = 1,2,3,\dots,I \quad (6)$$

$$dc_i \leq DC_i, \quad i = 1,2,3, \dots I \quad (7)$$

$$dt_i \leq DT_i, \quad i = 1,2,3, \dots I \quad (8)$$

$$c_i \leq C_i, \quad i = 1,2,3, \dots I \quad (9)$$

$$t_i \leq T_i, \quad i = 1,2,3, \dots I \quad (10)$$

$$a_i \leq A_i, \quad i = 1,2,3, \dots I \quad (11)$$

$$r_i \leq R_i, \quad i = 1,2,3, \dots I \quad (12)$$

$$\sum_{j=1}^{J_i} Y_{i,j} \geq 1, \quad i = 1,2,3, \dots I \quad (13)$$

where function (1) is the object function and (2) - (4) are the object constraints of the goal programming model. Function (5) and (6) are the waiting time and trust degrees constraints respectively. Function (7) and (8) are distance costs constraints, and (9) - (12) are the QoS constraints. Finally, function (13) describes the constraints of decision variables, which means that one or more business services can be selected for one business node. All the denotations used in the model have been defined in Section 2 and 5.1.

In the RBBSSFPM model, variables including $WT$, $DC$, $DT$, $C$ and $T$ can be classified into two categories: time category and cost category. The former one includes: $WT$, $DT$ and $T$, and the later one includes: $DC$ and $C$. The values of them need to be calculated according to the flow structure. There are mainly four kinds of basic flow structures in the business flow: sequential, parallel, branch and loop. Based on Menascé (2004), we give the calculation functions of $T$ and $C$ as the following equations, and the calculation functions of other variables are similar to that of $T$ and $C$.

① sequential: Assume that the business flow is formed by $I$ nodes in sequential structure, then:

$$T = \sum_{i=1}^{I} t_i, \ C = \sum_{i=1}^{I} c_i$$

② parallel: Assume that the business flow is formed by $I$ nodes in parallel structure, then:

$$T = max(t_i, i = 1,2,3, \dots, I), \ C = \sum_{i=1}^{I} c_i$$

③ branch: Assume that the business flow is formed by $I$ nodes in branching structure, and the probability of each node is $\rho_i$, then:

$$T = \sum_{i=1}^{I} (\rho t_i), \ C = \sum_{i=1}^{I} (\rho c_i)$$

④ loop: Assume that the business flow is formed by $I$ nodes in sequential structure, and there is a loop from the last node to the first node with the loop probability $p$, then:

$$T = \frac{1}{1-p} \sum_{i=1}^{I} t_i, \ C = \frac{1}{1-p} \sum_{i=1}^{I} c_i$$

### 5.3 The crisp equivalent

The proposed model in Section 5.2 is a fuzzy chance-constrained programming model. One way of solving the fuzzy chance-constrained programming is to convert the chance constraint into its crisp equivalent and then solve the equivalent crisp model by the traditional solution process.

In order to get the crisp equivalent, we introduce the Theorem 5.10 from Liu (2009):

"*Assume that the function* $g(x,\xi)$ *can be rewritten as,*

$$g(x,\xi)=h_1(x)\xi_1+h_2(x)\xi_2+\cdots+h_t(x)\xi_t+h_0(x)$$

*Where,* $\xi_k$ *are trapezoidal fuzzy variables* $(r_{k1},r_{k2},r_{k3},r_{k4})$, $k=1,2,...,t$, *respectively. We define two functions* $h_k^+(x)=h_k(x)\vee 0$ *and* $h_k^-(x)=-(h_k(x)\wedge 0$ *for* $k=1,2,...,t$. *Then we have*

*(a) when* $\alpha<\frac{1}{2}$, $Cr\{g(x,\xi)\le 0\}\ge\alpha$ *if and only if*

$$(1-2\alpha)\sum_{k=1}^{t}[r_{k1}h_k^+(x)-r_{k4}h_k^-(x)]+2\alpha\sum_{k=1}^{t}[r_{k2}h_k^+(x)-r_{k3}h_k^-(x)]+h_0(x)\le 0$$

*(b) when* $\alpha\ge\frac{1}{2}$, $Cr\{g(x,\xi)\le 0\}\ge\alpha$ *if and only if*

$$(2-2\alpha)\sum_{k=1}^{t}[r_{k3}h_k^+(x)-r_{k2}h_k^-(x)]+(2\alpha-1)\sum_{k=1}^{t}[r_{k4}h_k^+(x)-r_{k1}h_k^-(x)]+h_0(x)\le 0$$"

Constraint (6) in RBBSSFPM can be re-written as:

$$Cr\{-\tilde{tr}_i+TR_i\le 0\}\ge\alpha \tag{14}$$

According to the Theorem 5.10 from Liu (2009), the crisp equivalent of constraint (6) can be written as:

$$\begin{Bmatrix} (1-2\alpha)[tr_i^1h^+(x)-tr_i^4h^-(x)]+2\alpha[tr_i^2h^+(x)-tr_i^3h^-(x)] & \\ +TR_i\le 0 & \alpha<1/2 \\ (2-2\alpha)[tr_i^3h^+(x)-tr_i^2h^-(x)]+(2\alpha-1)[tr_i^4h^+(x)-tr_i^1h^-(x)] & \\ +TR_i\le 0 & \alpha\ge 1/2 \end{Bmatrix} \tag{15}$$

Where, $h(x)=-\sum_{j=1}^{J_i}Y_{i,j}$, $h^+(x)=h(x)\vee 0=0$, $h^-(x)=-(h(x)\wedge 0)=\sum_{j=1}^{J_i}Y_{i,j}$.

Therefore, function (15) can be written as:

$$\begin{Bmatrix} (1-2\alpha)[-tr_i^4\sum_{j=1}^{J_i}Y_{i,j}]+2\alpha[-tr_i^3\sum_{j=1}^{J_i}Y_{i,j}]+TR_i\le 0 & \alpha<1/2 \\ (2-2\alpha)[-tr_i^2\sum_{j=1}^{J_i}Y_{i,j}]+(2\alpha-1)[-tr_i^1\sum_{j=1}^{J_i}Y_{i,j}]+TR_i\le 0 & \alpha\ge 1/2 \end{Bmatrix} \tag{16}$$

The proposed model can be converted into its crisp equivalent by replacing constraint (6) by formula (16), which can be solved by the traditional solution process. The services selection and composition problem is a kind of NP problem (Yu *et al*. 2005, Garey *et al*. 1979). As an intelligent optimization algorithm, genetic algorithm (GA) has been widely used in solving various NP-Complete problems. In this paper, we will use GA to solve the crisp equivalent of the proposed model in section 5.2.

## 6 The GA-based algorithm

Generic algorithm (GA) is a stochastic search method for optimization problems based on the mechanics of natural selection and natural genetics. GA has demonstrated considerable success in providing good solutions to many complex optimization problems and received more and more attentions during the past decades.

In this section, we design a GA to solve the crisp equivalent of the proposed model in Section 5.2. First, we encode all nodes' numbers of the business flow as a chromosome. Second, we do genetic operations including selection, crossover and mutation operations on the chromosome to generate a new chromosome with better fitness value. Then we repeat the second step to achieve the parallel and global search in the solution space. When the algorithm stops, we get a set of chromosomes which is the solution set of the model and also is the set of service scenarios. The algorithm process is as follows.

**Step 1** Initialize *pop-size* chromosomes at random.

We define an integer *pop-size* as the number of chromosomes and initialize *pop-size* chromosomes randomly. Every chromosome is constituted by $I$ numbers, each of which is generated in the interval [1, $J_i$] randomly, $i$=1,2,3,…$I$. And let the generation $g$=0;

**Step 2** Check the terminating condition, if it is satisfied, the algorithm can output the optimized results and the program stops.

**Step 3** Generate the new chromosomes by tournament selection, two-point crossover and SWAP mutation operations.

**Step 4** Check the execution time constraint for each business node to select more business services if needed.

The detailed realization steps see in section 5.1.

**Step 5** Calculate the objective values for all chromosomes,

The objective function is defined as $f=100d_c^{+}+ 10d_t^{+}+ d_{wt}^{+}$, the coefficient in which can be adjusted if needed.

**Step 6** Compute the fitness of each chromosome via the objective values.

The fitness value is calculated by adding constraints punishments values to the objective value.

**Step 7** Generate the next generation, and let $g$=$g$+1, turn to step 2.

According to step 5, the chromosome or individual with the best fitness value will be in the next generation. The other chromosomes of the next generation are chose randomly from the current generation and the new generated chromosomes by genetic operations in Step 3. At last, the total individual number of the next generation is same as the current generation.

## 7 A method for business services negotiation

After discussing the business services selection method in previous sections, this section focuses on business services negotiation. The solution set of the business services selection model (RBBSSFPM) might be an empty set with some given business requirements from users, which means that no result can be achieved for the present user. Some special solutions may be achieved when the GA ends which make the object value of RBBSSFPM be zero, but some constraint cannot be satisfied. This kind of solutions is called sub-satisfactory result and all the sub-satisfactory results constitute a sub-satisfactory result set. For this case, there is a need for business providers and the user to negotiate over various non-functional requirements to obtain an integrative (near) optimal solution that fits the perspectives of both parties.

This section presents an offset-based automated negotiation approach to support the business services selection. This approach is implemented by a dynamic negotiation process with information exchange between providers and users. The aim of the process is to find a satisfactory solution for the present user. Assume that the sub-satisfactory result set is not empty. Items in the sub-satisfactory result set are used to the negotiation process successfully until a satisfied solution is reached. The negotiation process is described as following.

**Step 1** Get $K$ different solutions with less fitness value from the GA-based business services discovery algorithm. Every solution stands for a proposal of business services selection for the user's business flow. All the $K$ different solutions constitute the sub-satisfied result set.

**Step 2** Sequence all the $K$ different solutions by increasing fitness value, and number them successfully.

**Step 3** Negotiate the solutions in turn.

**Step 3.1** For $k$=1,…$K$, get the solution $k$.

**Step 3.2** For the solution $k$, set a value for the ratio($\varepsilon$), which is used to adjusting parameters values. Assume that $\varepsilon = 0.1, 0.2, 0.3, \ldots, 1$.

**Step 3.3** For the solution $k$, in order to express the negotiation process in convenience, we denote: $X_{1,i} = C_i,\ X_{2,i} = T_i,\ X_{3,i} = A_i,\ X_{4,i} = R_i,\ X_{5,i} = DC_i,\ X_{6,i} = DT_i,\ X_{7,i} = WT_i$ and $x_{1,i} = c_i,\ x_{2,i} = t_i,\ x_{3,i} = a_i,\ x_{4,i} = r_i,\ x_{5,i} = dc_i,\ x_{6,i} = dt,\ x_{7,i} = wt_i$.

**Step 3.4** For the solution $k$, let $\Delta_{u,i} = x_{u,i} - X_{u,i},\ u \in \{1,2,5,6,7\}$, and $\Delta_{u,i} = -(x_{u,i} - X_{u,i})$, $u \in \{3,4\}$, where, $i$=1,2,3,…$I$.

**Step 3.5** For $i$=1,2,3,…$I$,

if $\Delta_{u,i} \geq 0, u \in \{1,2,3,\ldots,7\}$, change parameters values for the corresponding service provider. These new parameters values are: $x_{u,i} = X_{u,i},\ u \in \{1,2,3,\ldots,7\}$.

If $\Delta_{u,i} < 0, u \in \{1,2,3,\ldots,7\}$, change parameters values for the corresponding service provider. These new parameters values are: $x_{u,i} = x_{u,i} - \Delta_{u,i} \times \varepsilon,\ u \in \{5,6\}$, and $x_{u,i} = x_{u,i} + \Delta_{u,i} \times \varepsilon$, $u \in \{3,4\}$.

**Step 3.6** For the solution $k$, Calculate the total value of all parameters for the provider according to the provider's function. Assumed the provider's function is a weighted sum of all the parameters except for $a_i$ and $r_i$, which will be replaced by $1/a_i$ and $1/r_i$ in the sum function. If the total value by calculating the sum function is acceptable for the provider, the negotiation for business node $i$ is success. Else, go to step 3.2 to get a new ratio until the ratio reaches at 1.

**Step 3.7** For the solution $k$, if negotiations for all business node $i$ are success, the solution $k$ with new parameters values will be a satisfactory solution for the present user. Go to step 4. If no solution $k$, $k$=1,…$K$, is negotiated successfully in the process, go to step 5.

**Step 4** Print the solution $k$ with new parameters values, and the negotiation process is end.

**Step 5** There is no satisfactory solution can be obtained from the negotiation process, and the negotiation process ends.

# 8 Case study

## 8.1 The business flow

In this section, an instance based on the Zhejiang business model is illustrated to discuss the multi factor-based business service selection method in SOBE.

There are as many as 800 business clusters in 85 cities of 88 cities in Zhejiang Province, including the well-known small wares in Yiwu, leather shoes in Wenzhou, ironwares in Yongkang, ties in Shengzhou and clothing in Ningbo etc. According to statistics, about 2/3 of the measurement device in China are produced in Yongkang. Clusters of these enterprises have greatly improved their industrial competitiveness and promote the rapid development of regional economy. Clusters of enterprises as in Zhejiang Province can be regarded as SOBE substantially. Different business units in enterprises are capsulated as business component, through which enterprises cooperate by service supplying and consuming in the whole value network.

Taking the Eastern Hongye clothing factory as an example, there are almost 30 embroidery factories, 20 dyeing factories ,accessories manufacturers and 10 Washing plant clustering around. These small factories survive by supping outsourcing services to the big factories, which in turn improves the big factories' competitiveness in dealing with the orders more efficiently (Kou 2004).

An instance basing on the business process of Ningbo clothing factories is illustrated below to explain the service selection method basing on boundary rationality in SOBE. As shown in Fig. 2, the whole producing process is composed of designing, manufacturing, and selling. Every node in the figure is defined as a business unit. Therefore, there are 17 business units totally, each of which is carried out by consuming services either inside or outside the factory.

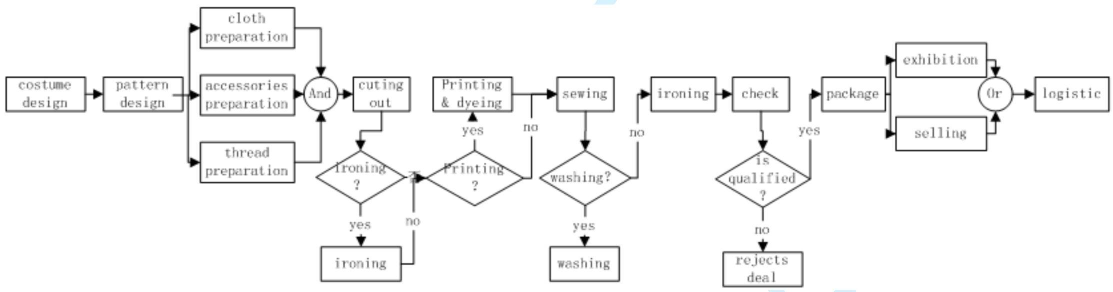


Fig. 2 The business flow

According to Fig. 2, the clothing design unit designs the clothing style and makes a set of sizes according to most people's navigation shape. After the design is confirmed by customers, different patterns of this design will be drawn in different sizes, and then patterns for production and layout table should be made. Then the lining, accessories and sewing thread should be prepared. The lining will be cut into different pieces according to the design in next unit. The pieces may be ironed, embroidered and washed if necessary afterward, and finally quality are checked. The qualified clothing then are packed, exhibited, delivered and sold.

## 8.2 Parameters initialization

It is assumed that every business service population has the same number of business services individuals. Parameters values are generated randomly in some real number intervals except the trust, reliability and availability. The corresponding intervals are $t_{i,j} \in (0,10]$, $c_{i,j} \in (0,20]$,

$wt_{i,j}\in[0,60]$, $dc_{i,j}\in[0,20]$, $dt_{i,j}\in[0,20]$, $i$=1,2,…$I$, $j$=1,2,3,…$J_i$. Herein, $I$=17 in this case study.

Parameters initialization of the trust is achieved in the following procedures. Before introduce the procedures, let numbers 1, 2, 3, 4, 5 denote trust levels “very untrustworthy”, ” untrustworthy”, ” unknown”, ” trustworthy”, ” very trustworthy”, respectively.

Step 1: Generate an integer in the interval of [1, 10] randomly for each individuals.

Step 2: If the random number generated in step 1 is less than 6, map the random number generated in step 1 with a trust level, else, let the trust level be “unknown”.

Step 3: Map the trust level with a trapezoidal fuzzy number according to Table 2 (see in section 4.1).

Step 4: For the current user, if the trapezoidal fuzzy number equal to (0.5, 0.5, 0.5, 0.5), then calculate the indirect trust degree according to the method proposed in section 4.1.

Step 5: Let the trust degree $\tilde{tr}_{i,j}$ equal to the trapezoidal fuzzy number found in step3 or step 4, $i$=1,2,3,…$I$, $j$=1,2,3,…$J_i$.

Parameters initialization of the reliability is similar to that of the trust. First, let numbers 1, 2, 3, 4, 5 denote values of reliability which are 0.1, 0.3, 0.5, 0.7, 0.9, respectively. Second, generate an integer in the interval of [1, 10] randomly for each individuals. Third, if the random integer is less than 6, map the random integer with a value of reliability depicted in the first step, else, let the value of reliability equal to 0.5. At last, Let the reliability value $r_{i,j}$ equal to the value determined in the third step, $i$=1,2,3,…$I$, $j$=1,2,3,…$J_i$.

For the availability, parameters initialization approach is same as that of reliability.

Then the user’s requirement-related parameters for each indicator are given in Table 2 with three user cases.

Table 4 the user’s requirement-related parameters ($i$=1,2,3,…$I$)

| parameters | $C_i$ | $T_i$ | $A_i$ | $R_i$ | $WT_i$ | $DC_i$ | $DT_i$ | $TR_i$ | $C_0$ | $T_0$ | $WT_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Case 1 | 19 | 9 | 0.5 | 0.5 | 55 | 19 | 19 | 0.5 | 140 | 55 | 260 |
| Case 2 | 17 | 9 | 0.5 | 0.5 | 50 | 17 | 18 | 0.5 | 140 | 55 | 260 |
| Case 3 | 17(30) | 9(3) | 0.5 | 0.5 | 50 | 17 | 18 | 0.5 | 140 | 55 | 260 |
| Case 4 | 15 | 7 | 0.5 | 0.5 | 45 | 17 | 17 | 0.5 | 120 | 50 | 230 |

where the parameters of Case 2 and Case 3 are the same except $C_i$ and $T_i$. In order to simulate the multi-services selection method for one node proposed in Section 5.1, we assume that the expected time value of one node in the business flow is less than half of the average value of services in the corresponding population in Case3. So let $T_6$=3 and $C_6$=30.

### 8.3 Calculation results and the analysis

According to the parameters initialization method in Section 8.2, the GA designed in Section 6 is used to solve the proposed model in Section 5. The experimental hardware platform is a PC with AMD Athlon (tm) II 2.71 GHz CPU and 1.87GB memory. The GA is programmed with VC++6.0. The crossover probability, mutation probability and population size of GA are taken as 0.3, 0.1 and 30, respectively. According to the business flow, see in Figure 1, let $I$ equal to 17, and suppose $J_i$=25, $i$=1,2,3,…$I$, $i\neq 13$, $I$ =17. The node 13 in the business flow(see in Figure 1) is

“Rejects deal“, which is not considered in this paper, so we let $i \neq 13$.

Five experiments are designed to illustrate the feasibility and effectiveness of the algorithm. For the first experiment, GA runs under three different cases (see in table 4) respectively, and services selection results is in Table 5. For the second experiment, GA runs with four different confidence level values respectively, and services selection results is in Table 6. The third experiment is designed to illustrate the convergence of the GA, the calculation results of which are in Table 7 and Table 8. The fourth experiment is used to show the effectiveness of the algorithm by testing the CPU execution time. And in this experiment, the CPU execution times are calculated under three different cases and six different generations, such as 50, 100, 150, 200, 300 and 400, respectively. In the last experiment, Case 4 in Table 4 is considered and the GA cannot return a solution with the fitness value equaling to zero. In this case, the negotiation algorithm is used to reach a satisfied solution for the present user.

Table 5 Business services selection results under three cases ($\alpha = 0.6$)

| Node number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 14 | 15 | 16 | 17 | Fitness |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case1 | 17 | 12 | 13 | 21 | 7 | 17 | 2 | 21 | 15 | 21 | 16 | 4 | 19 | 21 | 10 | 21 | 0 |
| Case2 | 7 | 21 | 5 | 20 | 16 | 17 | 9 | 20 | 18 | 2 | 21 | 19 | 19 | 21 | 23 | 21 | 0 |
| Case3 | 7 | 9 | 5 | 12 | 16 | 6, 18 | 9 | 20 | 12 | 17 | 16 | 18 | 19 | 21 | 25 | 21 | 0 |

Table 6 Business services selection results with different confidence level values (under Case2)

| Node number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 14 | 15 | 16 | 17 | Fitness |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha = 0.5$ | 8 | 12 | 5 | 8 | 5 | 17 | 5 | 20 | 12 | 17 | 9 | 25 | 19 | 4 | 24 | 21 | 0 |
| $\alpha = 0.6$ | 7 | 21 | 5 | 20 | 16 | 17 | 9 | 20 | 18 | 2 | 21 | 19 | 19 | 21 | 23 | 21 | 0 |
| $\alpha = 0.7$ | 5 | 21 | 19 | 20 | 16 | 17 | 9 | 20 | 12 | 9 | 21 | 19 | 4 | 21 | 17 | 21 | 0 |
| $\alpha = 0.8$ | 7 | 21 | 5 | 20 | 16 | 17 | 9 | 20 | 12 | 2 | 21 | 19 | 19 | 21 | 25 | 21 | 0 |

where numbers in medium parts of Table 5 and 6 denote the selected services’ numbers. For instance, the number 5 in the third row and fourth column of Table 6 denotes that the individual 5 in the population 3 is selected for the business node 3.

Table 5 shows that the fitness values under three cases all equal to zero, which means that the selected services set can satisfy each user’s needs under each case. It can be also seen that the services sets under different cases are different, which illustrates that each services set can satisfy the user’s need, but they are not necessarily an optimal services selection results for the business flow. Therefore, the proposed services selection method can not only choose satisfied services sets for users but also can reduce the queue length for the optimal services. Furthermore, for Case 3 in Table 4, two services are selected for the business node 6 which indicates the method of selecting multiple services for one business node advanced in Section 5.1.

It can be seen from Table 6 that all the fitness values are equal to zero with different values of the confidence level, which also shows that the selected services sets can satisfy users’ needs under each case respectively. Services sets are different with different values of the confidence level.

Results of the first and second experiments illustrate the feasibility of the proposed business services selection method.

Table 7 Fitness values of GA with different generations under three cases ($\alpha = 0.6$)

| GA generation | 30 | 50 | 70 | 100 | 150 | 200 | 300 |
|---|---|---|---|---|---|---|---|
| Case1 | 416 | 103 | 81 | 0 | 0 | 0 | 0 |
| Case2 | 956 | 552 | 420 | 130 | 19 | 0 | 0 |
| Case3 | 990 | 580 | 543 | 165 | 22 | 0 | 0 |

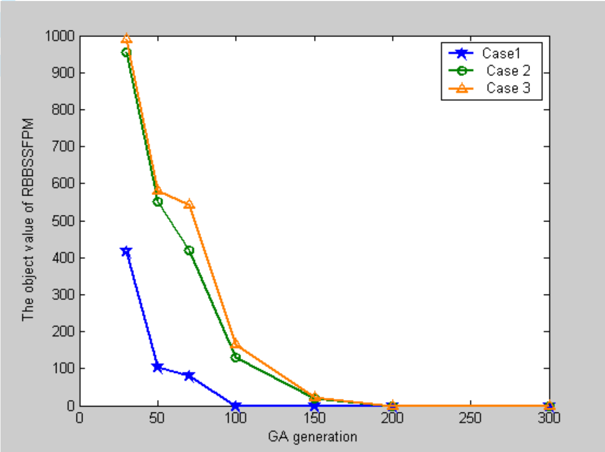


Fig. 3 Fitness values of GA with different generations under three cases ($\alpha = 0.6$)

Table 8 The execution time of GA with different generations under three cases ($\alpha = 0.6$)

unit: second

| GA generation | 50 | 100 | 150 | 200 | 300 | 400 |
|---|---|---|---|---|---|---|
| Case1 | 0.703 | 1.359 | 2.062 | 2.671 | 3.953 | 5.234 |
| Case2 | 0.672 | 1.312 | 1.921 | 2.578 | 3.828 | 5.093 |
| Case3 | 0.609 | 1.203 | 1.843 | 2.453 | 3.718 | 4.875 |

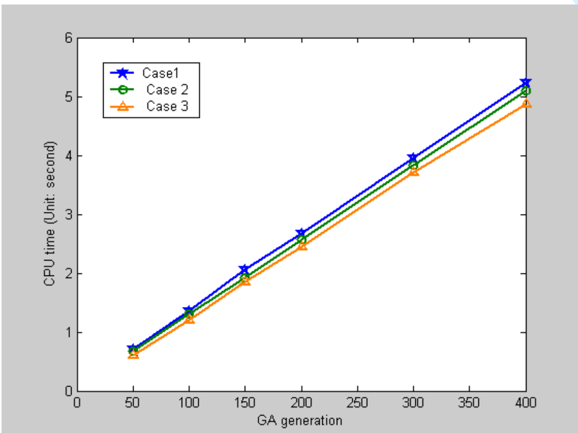


Fig. 4 The execution time of GA with different generations under three cases ($\alpha = 0.6$)

Table 7 shows that the fitness value of the RBBSSFPM model decreases gradually with the increase of the GA generation. When it is at about 200 generation, the GA fitness values have reached zero, which means that the corresponding satisfactory services set have been found for users. It can be seen from Fig. 3 that the line of Case 1 evolves faster than that of the other two cases and reaches 0 at 100 generation, as well as the line of Case 3 evolves slower than that of the other two cases. Considering that the user requirements are up-grading from Case 1 to Case 3, Fig. 3 illustrates that the GA evolution is decreases gradually with the increase of users' requirements.

It can be seen from Table 8 and Fig. 4 that the GA execution time increases with the raise of GA generation with linear growth rate. The GA execution time is almost the same with different GA generations under different cases. Consequently, the execution time is affected by the GA generation, but is not influenced by user requirement cases.

Under Case 4 in Table 4, no satisfactory solution with fitness value equaling to zero can be found by GA, and some solutions are achieved with object value equaling to zero. Some these kinds of solutions are listed in table 9.

Table 9 Business services selection results under Case 4 ($\alpha$ =0.6)

| Node number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 14 | 15 | 16 | 17 | Fitness |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Solution1 | 15 | 21 | 19 | 8 | 13 | 7 | 20 | 10 | 12 | 17 | 9 | 25 | 12 | 21 | 9 | 7 | 100 |
| Solution2 | 15 | 21 | 19 | 8 | 13 | 7 | 20 | 10 | 12 | 17 | 9 | 25 | 12 | 13 | 9 | 7 | 109 |
| Solution3 | 15 | 21 | 19 | 8 | 13 | 7 | 20 | 12 | 12 | 17 | 9 | 25 | 12 | 13 | 9 | 7 | 115 |

where the object value of each solution equal to zero, and the fitness value of each solution is more than zero. The fitness values of these solution are different each other.

After the execution of the negotiation algorithm proposed in Section 7, solution 1 in Table 9 changes to a satisfactory solution for the present user by adjusting some parameters values with the negotiation algorithm. Table 10 and Table 11 list the parameters values of services discontenting some constraints in RBBSSFPM before and after negotiating respectively.

Table 10 Parameters values of services before negotiating with solution 1 in Table 9

| $(i, j)$ | $c_{i,j}$ | $t_{i,j}$ | $a_{i,j}$ | $r_{i,j}$ | $dc_{i,j}$ | $dt_{i,j}$ | $wt_{i,j}$ |
|---|---|---|---|---|---|---|---|
| (6, 7) | 6.91855 | 4.44533 | 0.5 | 0.5 | 13.1895 | 10.4599 | 47.1474 |
| (9, 13) | 14.5665 | 0.352489 | 0.5 | 0.5 | 0.657979 | 17.7502 | 28.3273 |
| (17, 7) | 0.0402844 | 0.838343 | 1 | 1 | 7.84509 | 19.0283 | 52.9704 |

Table 11 Parameters values of services after negotiating with solution 1 in Table 9

| $(i, j)$ | $c_{i,j}$ | $t_{i,j}$ | $a_{i,j}$ | $r_{i,j}$ | $dc_{i,j}$ | $dt_{i,j}$ | $wt_{i,j}$ |
|---|---|---|---|---|---|---|---|
| (6, 7) | 8.53484 | 4.95626 | 0.5 | 0.5 | 13.9516 | 11.7679 | 45 |
| (9, 13) | 14.6532 | 1.68199 | 0.5 | 0.5 | 3.92638 | 17 | 31.6618 |
| (17, 7) | 3.03223 | 2.07067 | 0.9 | 0.9 | 9.67607 | 17 | 45 |

Before the negotiation, the object value of solution 1 in Table 9 with parameters value listed in

Table 10 is zero, but the fitness value of the solution 1 is more than zero, which means that the solution 1 cannot satisfy the present user's requirements absolutely by violating some constraints in RBBSSFPM. After the negotiation, the fitness value of solution 1 in Table 9 with new parameters value listed in Table 11 reaches at zero, which means this solution is a satisfied business services selection solution now. This illustrates that the negotiation method can help users to get satisfied solutions by negotiating with some parameters values when no satisfied solutions with fitness value equaling to zero can be found by the discovery method in some cases.

In conclusion, the results of the above five experiments demonstrate the feasibility and effectiveness of the proposed business services selection method.

## 9 Conclusions

The business services selection is a key issue for sharing and integrating of business services in the SOBE. Existing researches mainly focus on QoS-based selection methods. Trust degrees also play an important role in the selection of business services. In order to select services more accurately, a trust-based approach to selection of business services is proposed in this paper. After analyzing the methods of acquisition and calculation of parameters, a fuzzy chance-constrained programming model is established for the business services selection by considering four kinds of factors involving QoS attributes, trust relation, physical distance and waiting time. Furthermore, for the case in which no satisfactory solution can be found by the GA-based algorithm, a negotiation method is developed to reach at a satisfying solution. Finally, five experiments have been conducted to test the proposed approach. The results demonstrate the feasibility and effectiveness of the proposed approach.

The services selection is bound up with not only QoS attributes and trust degrees but also relationships among services. In the future, we plan to explore the relationships among services so that a comprehensive feasible services selection solution can be achieved. Also, a services relationships network needs to be developed to facilitate the services selection and composition.

## Acknowledgement

This work is supported by National Natural Science Foundation of China (Key Program, No. 61033005), the National High Technology Research and Development Program of China (863 Program, No. 2009AA010308) and the National Key Technology R&D Program (No. 2008BAH32B03-1).

## References

Azzedin, F., Ridha, A., and Rizvi, A., 2007. Fuzzy Trust for Peer-to-Peer Based Systems. *World Academy of Science, Engineering and Technology*, 27, 123-127.

Beth, T., Borcherding, M., and Klein, B., 1994. Valuation of trust in open network. *In: Proceedings of the European Symposium on Research in Computer Security(ESORICS)*, 07-09 November, Brighton, United Kingdom. Brighton: Springer-Verlag, 875, 3-18.

Dai, G., and Wang, Y., 2009. Trust-aware Component Service Selection Algorithm in Service Composition. *In: 2009 International Conference on Frontier of Computer Science and Technology*. 17-19 December, Shanghai, China. 613-618.

Desai, N., Mazzoleni, P., and Tai S., 2007. Service Communities: A Structuring Mechanism for Service-Oriented Business Ecosystems. *In*: *2007 IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2007)*. 21-23 Feb, Cairns, Australia. 122-127.

Fan, Y., 2008. *The strategy and method of Informatization management*. Beijing: tsinghua university press.

Fan, Y., 2010. Architecture and key technology of serviced-oriented enterprise. *Aeronautical Manufacturing Technology*, 3, 26-31.

Gambetta, D., 1990. Can We Trust Trust? *In: D. Gambetta, ed. Trust: Making and Breaking Cooperative Relations*, Basil Blackwell. Oxford, 213-238.

Garey, M., and Johnson, D., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H.Freeman and Company.

Hammadi, A., Dillon, T., and Chang, E., 2009. Business Service Composability on the Basis of Trust. *In: 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*. 1-3 June, Istanbul, Turkey. 437-440.

Hung, P.C.K., Li, H., and Jeng, J.J., 2004. WSNegotiation: an overview of research issues. *In: Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS'04)*, 5-8 January, Big Island, Hawaii . 1-10.

Jøsang, A., 1996. The Right Type of Trust for Distributed Systems. *In: Proceeding of the 1996 New Security Paradigms Workshop*. New York, USA. ACM Press, 119-131.

Jøsang, A., 1999. Trust-Based Decision Making for Electronic Transactions. *In: Proceedings of the 4th Nordic Workshop on Secure Computer Systems(NORDSEC'99)*. Kista: Stockholm University Press, 1-21.

Keller, A., and Ludwig, H., 2003. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management, Special Issue on E-Business Management*, 11(1) , 1-21.

Kohlborn, T., Korthaus A., Riedl C., et al., 2009. Service Aggregators in Business Networks. *In*: *Proceedings of 1st Workshop on Service-Oriented Business Networks and Ecosystems*. 1-4 September, University of Auckland, New Zealand. 195-202.

Kou, Q., 2004. *The investigation of Ningbo textile industry's competitiveness by GEM model*. Shanghai: Donghua University.

Li, L., Wei, J., and Huang, T, 2007. High Performance Approach for Multi-QoS Constrained Web Services Selection. *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 4749, 283-294.

Liou, T.S., and Wand M.J., 1992. Ranking Fuzzy Numbers with Integral Value . *Fuzzy Sets and Systems*, 50(3), 247-255.

Liu, B., 2009. *Theory and practice of uncertain programming*. 2nd Edition. Berlin: Springer.

Menascé, D.A., 2002. QoS Issues in Web Services. *IEEE Internet Computing*, 6, 72–75.

Menascé, D.A., 2004. Composing Web Services: A QoS View. *IEEE Internet Computing*, 6, 88-90.

Misztal, and Barbara, A., 1996. *Trust in Modern Societies*. Cambridge, UK: Polity Press.

Patankar, V., and Hewett, R., 2008. Automated Negotiations in Web Service Procurement. Internet and Web Applications and Services. *The Third International Conference on Internet and Web Applications and Services*, 8-13 June, Athens, Greece. 620 – 625.

Ronald S Burt, R.S., 2000. decay function. *Social Networks*. 22(1), 1-28.

Song, S., Hwang, K., Zhou, R., et al., 2005. Trusted P2P trans-actions with fuzzy reputation aggregation. *Internet Computing*, 9 (6), 24-34.

Sun, D., 2008. *Researches on Ubiquitous Computing Security Based on Trust and Service Model*. Shanghai: Donghua University.

Yu, B., Singh, M.P., and Sycara, K., 2004. Developing Trust in Large-Scale peer-to-peer Systems. *In: Proceedings of the lst IEEE Symposium on Multi-Agent security and survivability*, 30-31 August, Drexel, PA, USA. 1-10.

Yu, T., and Lin, K., 2005. Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. *Lecture Notes in Computer Science*. Springer, Heidelberg, 3826, 130–143.

Goal programming. http://en.wikipedia.org/wiki/Goal_programming.