A Fast Searching Approach for Top-k Partner Selection in Dynamic Alliances of Virtual Enterprises

Bishan Ying[1], Pingping Zhu[2] and Ye Gu[2]
1 Wasu Media Network Co., Hangzhou, China
2 The College of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China
billsun_ying@hotmail.com, 18768132808@163.com, zhuge011@126.com

Abstract—**The success of supply chain management largely depends on establishing the partnership in dynamic alliance. In the past, many researches focus on selection of indexes and establish selection model according to the indexes. In order to speed up the process of selection, this paper introduces this optimized idea of Top-k into the selection of partner and classifies the original indexes. The paper presents algorithm of OP (Optimize Procedure) based on specific index and the experiment shows that these algorithms can efficiently improve the process of selecting partners. Furthermore, the paper also presents an algorithm of IMOP (Improve Optimize Procedure) based on OP algorithm. It can effectively overcome the false alarm rate of the OP algorithm and improve the accuracy of partner selection.**

Index Terms—**Dynamic alliance; Sorting model; Top-k Query; Virtual enterprise**

I. INTRODUCTION

Dynamic alliance, also known as Virtual Enterprise (VE), is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks [1]. The members of virtual enterprise could locate at different places. Different product capacities and technologies of each member make a large contribution to the alliance through its own core competence. Enterprises with diverse core resources are the foundation of dynamic alliance. Having complementary competencies, the enterprises demand seeking partners. As a result, the selection of partners in a virtual enterprise must obey the core competence principal. It requires the leading enterprise to select partners on the basis of needed core competences, while any participant must have the ability and contribute its own core competence to the alliance. Moreover the core competence is unique.

However, a selection process is a complex and hard task. The leading enterprise which is in charge of the operation of the whole virtual enterprise confirms the outsourcing tasks according to its current situation, and then selects the potential enterprises according to their legal tenders. In addition, there are a huge number of enterprises attending the process, each of which has distinctive characteristics. There are many mathematics models and selecting methods to identify the partner. The objective is to identify 1-3 or more candidates. This is also a sorting problem.

This paper focuses on how to select an expected partner more quickly, accurately and efficiently based on known selection methods and the demands of leading enterprise. What the leading enterprise seeks is to obtain instant aids from some participants, not all of them, by selection. Therefore, it is better to utilize the concept of Top-k selection to minimize the time of selection process.

The rest of paper is organized as follows. Section 2 discusses the related works. In the section 3, selection factors are discussed. Our OP and IMOP algorithms are presented in section 4. Some experiment results are given in section 5 to evaluate the performance of the proposed method. We conclude in section 6.

II. RELATED WORK

Nowadays, researchers have already presented a wide investigation of virtual enterprise (dynamic alliance) from diverse perspectives. Virtual enterprise as a new agile manufacture paradigm keeping pace with external environment has been paid an attention by both academic and business communities. During the formation of virtual enterprise, partner selection is a crucial and extremely complex work, due to the partners' independency and provisional cooperation relationship between them. Lots of researches on this issue have been done. According to the task precedence relationships, Wang [2] establishes a non-analytical mathematical model based on the time and cost and gives the solution on a GA / FD algorithm. In light of multi-tiers indication system, AHP method has been applied widely to determine the factor weigh [3, 4]. However, during the formation of judgment matrix, the AHP method has its limitation. When there are many decision-making factors and many candidates, it is necessary to provide more

Manuscript received Oct. 10, 2013; revised January 7, 2014; accepted Feb. 10, 2014.
Corresponding author: Bishan Ying, Wasu Media Network Co., Hangzhou, China

doi:10.4304/jcp.9.9.2013-2021

complicated information and knowledge to the decision makers. Meanwhile, experts from various fields have to join the decision making process to have an appraisal to achieve a reasonable decision. Mun [5] proposes a trust evaluation method of supporting enterprise collaboration and maximizing the satisfaction of cooperation. From their point of view, trust means the goal achievement probability. Trust value of an enterprise can be obtained by a fuzzy inference system whose rule-base is based on the top-level goal of a VE. According to the selector's preference, various rules can be applied to trust evaluation. Zhang [6] introduces the theory of Fuzzy Cognitive Time Maps (FCTMs) into modeling and evaluating trust relationships and shows how relevant is the inter-organizational trust based on trust sources and their credibility. The contribution is a methodology by taking dynamic nature of trust into account to analyze evolution of trust in the VE's setting. Blankenburg [7] describes a cooperative game approach which allows the members to make full use of their resources and to maximize the utility of coalition.

The members can control and bound the risk caused by the possible failure or default of some partner agents by spreading their involvement in diverse coalitions. Ono [8] explores the roles of learning and evolution in a non-cooperative autonomous system through a spatial IPD (Iterated Prisoner's Dilemma) game. When enterprise using of non-cooperative game behavior, learning is not effective for achieving mutual cooperation except under certain special conditions. The learning process depends on the spatial structure. So it will ultimately affect the realization of inter-enterprise profits and equitable distribution. An approach to fair imputation mechanism of gains of coalitions is discussed in [9]. On one hand, the mechanism has to make every member satisfied; on the other hand, it has to make every member have such a concept that it is not reasonable to betray its own alliance.

Algorithms proposed in this paper are to select the most appropriate partner effectively based on some existing approach (e.g., scoring function). This research field is an issue of partner selection sorting, which is a quite new aspect in the field of partner selection. There are wide researches on Top-k sorting issue by international scholars. The applications and improvements of FA algorithm [10] are in [11, 12, 13, 14]. FA algorithm is a visit of paralleling operating all procedures and evaluation objects, until at least k objects over all visits, while it has visited after k-object scores over a default threshold. Nowadays, sorting retrieve algorithm has been developed, including supporting any condition link [15], text retrieve based on possibility [16], P2P retrieval [17] etc. Gravano's work [19, 20] is the evaluation of the Top-k selection queries. The sorting algorithm proposed in this paper is based on a user-defined procedure, with a similar principle as to [18]. But in the complex system factor, each factor calculation is 'user-defined'. We do not calculate all object factor values. The retrieving efficiency is improved by only calculating the useful ones for the comparison. Chang [18] processes all the factors by same methods, ignoring some personalized demand from the clients. This paper would improve the problem.

III. Selection Factors

There are numerous appraisal factors on the enterprises' comprehensive ability, from various perspectives. Generally speaking, Time (T), Quality (Q), Production Capacity (PC) and Cost (C) are fundamental factors leading to success in an international market. In VE, various enterprises form an agile dynamic alliance system based on some product demand. The production process is based on cooperation from several enterprises, each of which is responsible for one part of the outsourcing task. Each enterprise would have different influence on all the others. When it comes to partner selection, it must consider interaction and geographic location, beside the internal factors. Besides Time, Production Capacity, Quality, Cost, other essential factors which must be referred during the selection process are Adaptability (A) and Flexibility (F), since it must have an immediate response to the emerging products for sustaining an agile dynamic alliance.

Agile logistics (AL) is a key factor as well. The enterprises forming the alliance require Agile Logistics. The Advancement (AD) of computer and information technologies among these enterprises has to be consistent. If there is much difference on information technologies among partners, it must be an obstacle on a successful alliance. Under such a circumstance, it could not be a stable collaboration among the partners, since the developed enterprise has to support the developing one. Environment (E) plays a more and more important role in adapting the requirement of sustainable development. Green manufacture has been considered as one assessment factor. In addition, Creditability (CA), is an essential requirement to join the alliance. Each member must complete its own assignment promptly. Otherwise the whole supply chain would be delayed. Creditability consists of two parts, prestige and performance. Prestige is a foundation of mutual credits. The reason we lay special emphasis on the prestige is that dynamic alliance forms randomly, which is an interim cooperation organization usually. The relationship among members is not a superior-subordinate. Although a protocol may be initialed to regulate their actions, it could not cover any potential risk and opportunity in the inconstant market. While people have a pursuit to the fashionable products, opportunities come up and each alliance member would have the same opportunity cost accordingly. Once the opportunity cost is above the threshold, a repulsive force would be conceived and lead to a suspension or failure of the current alliance, which results in a disastrous outcome. Other factors such as local legal system and business custom etc. are also regarded as factors. Above all, crucial decision factors of partner selection could be presented as follows: T (Time), Q (Quality), CA (Creditability), PC (Production capacity), F (Flexibility), A (Adaptability), AL (Agile logistics), and E (Environment).

IV. TOP-K SELECTION ALGORITHM

A leading enterprise has some demands for candidate partners. For example, the leading enterprise has its own valuation for some outsourcing tasks which will be allocated to candidate partners. So, the leading enterprise may study the complete time, geographical location, and daily production for each candidate, and then estimate risk cost (also called risk score) which the leading enterprise may spend money on each candidate. If one candidate is a large enterprise which is nearly the leading enterprise, it can get a small risk score by leading enterprise. A leading enterprise scores every candidate partner by its appraisal. Then it can select k members according to their scores.

A. User-defined Procedure

In fact, most DBMS (Microsoft SQL Server, IBM DB2, Oracle, PostgreSQL) supports user defined functions, and allows users to define these functions (procedure) in general programming languages. But these functions defined by users (or originated from external database property) are regarded as expensive procedures, because it could not present query result immediately when being invoked. As the assessment criteria in dynamic alliance mentioned, nearly each factor is an expensive and user-defined procedure. And thus, it is necessary to make a comparison of these objective enterprises, calculating the procedures which need the comparison and omitting the rest.

B. Sorting Model

We introduce a sorting model for an effective and efficient process.

1) Query process

The leading enterprise is to seek a partner with minimum risk cost. And we call the sorting model as *leading cost risk model*. In addition, we use a monotonic function (scoring function) for evaluating each enterprise, and get k object enterprises.

Assume a table describes information of enterprise which includes columns as distance, quantity, time, history creditability, quality, and scale. We use the query criteria shown in Table I to look for 3 satisfactory partners.

TABLE I.

QUERY CRITERIA

Select id from enterprise Where PQ(quality) c_1, Cost(distance, quantity, time) c_2, CA(history creditability, scale) c_3 Order by $\max(c_1, c_2, c_3)$ ascend stop after 3

Let $f(u) = \max(c_1, c_2, c_3)$, where u represents some enterprise and c_1, c_2, c_3 represent procedures. The scoring function f is the MAX function, which returns the maximum value of these procedures. The return values of the procedures, e.g. c_1, c_2, c_3, are from 0 to 1. We imply each procedure as a risk cost. Intuitively, we think that the smaller the value of each procedure is, the less the cost spends. For example, product quality is 1 which means first class, i.e., PQ(1)=0.2. With distance =30km, quantity =10000 items, time = 7 days, we have Cost(30,10000,7)=0.3; If history credibility and scale are both 3 which means moderate, we have CA(3,3)=0.5. And therefore the scoring function is max(0.2,0.3,0.5)=0.5, which returns the largest value of the three variables.

Table II shows the values of user-defined procedures converted from original data where OID is enterprise id. The fifth column is the returning value of the MAX function. And OID is enterprise id. The two enterprises with least cost enterprises are c and e to their leading enterprise.

TABLE II.

THE DATA OF SEARCHING PROCESS ACCORDING TO TABLE I

OID	c_1	c_2	c_3	$f(c_1, c_2, c_3)$
a	0.9	0.85	0.75	0.9
b	0.8	0.78	0.9	0.9
c	0.7	0.75	0.2	0.75
d	0.6	0.9	0.9	0.9
e	0.5	0.7	0.8	0.8

2) Traditional method

During the traditional sorting process, we firstly calculate each procedure value, and get the maximum values from the procedures, and finally sort these values in ascending order and get the first k least values. c_1, c_2, c_3, which represent user-defined procedures. Calculation of these procedures need some cost. We define the cost of c_i as e_i, which is calculated for n_i times, and thus the whole program cost is calculated by Eq.(1).

$$PE(P) = \sum_{i=1}^{m} n_i e_i\ , \quad (0 \le n_i \le N) \qquad (1)$$

Where, m is the number of user-defined procedures. Generally, we calculate nearly each procedure to look for the k objects. Assume there is N items in the data table, then we have Eq(2) from Eq(1).

$$PE(P) = \sum_{i=1}^{m} Ne_i\ . \qquad (2)$$

Eq.(2) shows that the cost does not matter with k value. It is obviously unreasonable, especially when k=3 or less, large work of calculation is a waste. Therefore, this sorting process needs to be optimized.

3) Necessary calculation

We have to confirm the target procedures for compressing the calculation. The necessary calculation would have an impact on the final k object results. How can we identify a necessary procedure calculation? To simplify the process and speed up the process of selection, we assume c_1 is not a user-defined procedure in Table II, which is not expensive. c_2, c_3 are user-defined procedures. Let $k = 1$. $f(u) = \max(c_1, c_2, c_3)$, and u represents an objective enterprise. In addition, Eq.(3) is the maximum value in current procedure set.

$$\hat{f}[z_1,\cdots,z_n](u)=\begin{cases} z_i=c_i(u) & if \quad c_i\in Z \\ z_i=0 & otherwise \end{cases} \tag{3}$$

Where, $Z\subseteq\{c_1,\cdots,c_n\}$, z_i represents corresponding procedure value, which is 0-1, for any u in Z, while $\hat{f}_Z(u)$ represents the maximum value. When some c_i is not in Z, then $z_i=0$.

Let us sort each object in ascending order by c_1 from table II. And the result is shown in Table III.

TABLE III.
SORTING ASCENDINGLY BY C1

OID	c_1	c_2	c_3	$f(c_1,c_2,c_3)$
e	0.5	0.7	0.8	0.8
d	0.6	0.9	0.9	0.9
c	0.7	0.75	0.2	0.75
b	0.8	0.78	0.9	0.9
a	0.9	0.85	0.75	0.9

Let us start with the first row. We have $\hat{f}(e)=\max(0.5,0,0)\geq c_1(e)=0.5$. Since we are not sure, whether $\hat{f}(e)$ would increase after introducing c_2, calculating $c_2(e)$ is necessary. After calculating $c_2(e)=0.7$, $\hat{f}(e)=\max(0.5,0.7,0)=0.7$ is hold. Now, the minimum value is $\hat{f}(d)=\max(0.6,0,0)\geq c_1(d)=0.6$. Similarly, it is necessary to calculate $c_2(d)$. After calculation, $c_2(d)=0.9$, and $\hat{f}(d)=\max(0.6,0.9,0)=0.9$. The current two least values are $\hat{f}(e)=0.7$ and $\hat{f}(c)=0.7$. Because c_3 is not referred, it is necessary to calculate $c_3(e)$ and $c_2(c)$. Consequently, $\hat{f}(e)=\max(0.5,0.7,0.8)=0.8$ and $\hat{f}(c)=\max(0.7,0.75,0)=0.75$. And the minimum value is $\hat{f}(c)=0.75$. Calculating $c_3(c)$ is essential too. Finally, $\hat{f}(c)=\max(0.7,0.75,0.2)=0.75$, and c is what we need. The total number of user-defined procedure calculation is reduced to 5, which is more effective than a traditional method with 10 calculations.

4) *Algorithm OP*

Based on the process describe above, the necessary calculation principle is: in a sort query process, assume there is a scoring function f, which needs to return k results. u is an object, and c_j is the next calculated procedure. When k objects do not exist, for any object o_i, $\hat{f}_{T_u}(u)<\hat{f}_{T_{o_i}}(o_i)$, c_j will be calculated, where $T_u=\{c_1,\cdots c_{j-1}\}$, $T_{o_i}\subseteq\{c_1,\cdots c_n\}$, and n is the number of procedures.

Based on the principle above, the sorting algorithm is in Figure 1.

In Figure 1, statement 1-8 is the initialization process. Statement 9-16 is the main process.

Algorithm $OP(f,k,C,D)$ //OP=Optimization Procedure
Input: f : scoring function;
k : number of returning object;
D : original dataset;
C : $\{c_1,\cdots c_n\}$ is procedure set.
Output: R // k objects totally

Procedure:
1. sort the original data in D in ascending order based on some non user-defined procedure, recorded in queue C_1; // C_1 is an ascending stack (or queue) based on some non user-defined procedure
2. assume $R=\varphi$, $F=\varphi$. // R : output set, F is a priority queue, which is an ascending one based on the score.
3. flag = false //flag is used to identify whether stop current procedure calculation; when flag=false, pursue calculating; whereas flag =true, stop calculating
4. while($C_1\neq\varphi$) {
5. $u=C_1.top()$ // the first object value in C_1 is assigned to one intermediate variable u.
6. $T_u=\{c_1\}$, $u.scoring=\hat{f}_{T_u}(u)$
7. $F.insert(u,u.scoring)$
8. } // after the completion of while loop, $F=\{f_{c_1}(u_1),\cdots,f_{c_1}(u_{|D|})\}$, the score priority is based on c_1. $|D|$ represents the number of original data set.
9. while(flag = = false){
10. $u=F.top()$ // u is an object
11. If u has completed final calculation of c_n then u join R, and $|R|=|R|+1$
12. else: {
13. calculating $c(u)$ // c is the next calculated procedure, $c\in C$
14. $T_u=T_u+\{c\}$, $u.scoring=\hat{f}_{T_u}(u)$
15. $F.insert(u,u.scoring)$ // insert based on ascending }
16. if($|R|\geq k$) then flag =true }
17. return R

Figure 1. Optimization procedure (OP)

5) *Algorithm IMOP*

Sometimes, a leading enterprise may concern some factors but ignore the rest in the assessment.

In table IV, assume that we have three factors, environment c_1, credibility c_2, cost c_3, and need top-1 selection. The leading enterprise concerns the risk score of the cost factor indeed, so the cost c_3 becomes preferential. The result of above algorithm is a, but what the leading enterprise needs is b.

This paper considers all factors as a risk cost which the leading enterprise spends on its partners, and thus adopts $f(u)=\max(c_1,\cdots,c_n)$ as scoring function, where n is the number of factors. For example, for each object, find the maximum cost among all the fields, and sort all the objects by the maximum value in an ascend order. The cost of the rest fields of a chosen object must be lower than its maximum cost. We do not doubt scoring function idea (which is maximum-minimum idea), but what we do is how to find the most ideal partners based on the leading enterprise's requirement. Accordingly, we

TABLE IV.

EXISTING PRIORITY FACTORS INQUERY THE DATA

OID	c_1	c_2	c_3	$f(c_1,c_2,c_3)$
a	0.6	0.6	0.3	0.6
b	0.7	0.8	0.2	0.8
c	0.8	0.9	0.2	0.9

Algorithm $IMOP(f,k,C,D,M)$ //IMOP=IMproved Optimization Procedure

Input: f : scoring function;

k :number of return object;

D : original dataset; $C:\{c_1,\cdots c_n\}$ procedure set;

M : remark array

Output: R // k objects totally

Procedure:

1.Through array M, confirm user-defined treating procedure set C', $C'\subseteq C$, n' is the number of C'

2.if($C'=\varphi$)

3. sort the original data in D ascendingly based on some non user-defined procedure, recorded in queue C_1 // C_1 is an ascending stack(or queue) based on some non user-defined procedure

4. else

Calculating $\forall c\in C'$ all object values $\{c(u)\mid u\in D\}$,recorded in C_1

5.assuming $R'=\varphi$, $F'=\varphi$, $W=\varphi$ .// R':output set; F' is a priority queue, which is a ascending one based on the score; W is a set reserving k results' scores

6.flag = false //flag is used to identify whether stop current procedure calculation, when flag=false, purse calculating; whereas flag=true, stop calculating

7. while($C_1\neq\varphi$) {

8. $u=C_1.top()$ //the 1st object value in C_1 is assigned to one intermediate variable u.

9. $T_u=\{c_1\}$, $u.scoring=\hat{f}_{T_u}(u)$

10. $F'.insert(u,u.scoring)$

11. } // after the completion of while loop, $F=\{f_{c_1}(u_1),\cdots,f_{c_1}(u_{|D|})\}$,the score priority is based on c_1. $|D|$ represents the set number in original data set

12. while(flag= =false){

13. $u=F'.top()$ // u is an object

14. If u has completed final calculation of $c_{n'}$ then u join R, $u.scoring$ join W ,and $|R|=|R|+1$

15. else: {

16. calculating $c(u)$ // c is the next calculated procedure, $c\in C'$

17. $T'_u=T'_u+\{c\}$, $u.scoring=\hat{f}_{T_u}(u)$

18. $F.insert(u,u.scoring)$ //insert based on ascending

}

19. if($|R|\geq k$) then flag =true

}

20. SM(W, F', k, $|D|$, n')// the final object confirmed

21. $R=R-R\cap R'$ // R' from SM

22. $OP(f,1+t,C-C',R')$

23. return R

Figure 2. IMproved Optimization Procedure (IMOP)

introduce a marking array M, and identify the factor preference in advance, where n the number of elements is in M, and the element value is 0 or 1. The algorithm OP is improved as Figure 2.

In Figure 2, statement 1-11 is the initialization process. Statement 2-19 is a necessary calculation of factors favored by the leading enterprise. Statement 20-22 is a process dealing with a situation when same values appear. Because it is not necessary to calculate the whole factors entirely, we can choose some factors which the leading enterprise concern. However, if the same value appears, the priority could not be confirmed. In Table 4, if enterprise c is in front of enterprise b, we would possibly choose enterprise c under top-1 situation; whereas enterprise b's overall ability is better than enterprise c's. The procedure SM deals with the situation of same value in Figure 3. Statement 1 is initialization process. Statement 2-6 is searching same value of scoring object. Finally, statement 7 is returning result. After the procedure SM, if there are still objects with same value and all the factors have been calculated, we would select the object among those with same values according to the time of formal tender submitted by small enterprise.

Algorithm SM $(W, k, F', |D|, n')$ //SM=Searching Method

Input: W :is a set reserving k results' scores; k :the number of returning object; $|D|$:represents the set number in original data set; F' is a priority queue, which is an ascending one based on the score; n' is the number of C'

Output: R',t // R' the final object confirmed, t is the number of value equal to k th value

Procedure:

1. $m = |D|$, $x = W[k]$, $i = k-1$,j=k, t=0
2. while(i!=0){ //seeking object with same score upward
3. if(x == W[i-1]) then i-- ,t++, u join R';
   }
4. while(j!=m) {//seeking object with same score downward
5. $u = F'.top()$;
6. if u completes the calculation of final $c_{n'}$ {
   if(x = =u.scoring) then u join R', j++ ;
   }
   }
7. return R' and t

Figure 3. Procedure SM

V. SIMULATION

According to the query in Table V, This experiment simulates a real scenario. It produces 1000 data records, which represent 1000 object enterprises. Each of enterprise has 8 properties: Time, Cost, Quality, Credibility, Product Capacity, Flexibility, Adaptability, Agile Logistics, Environment, Distance, and Quantity.

TABLE V.
SEARCH STATEMENT

Select id from enterprise
Where c_1 (AL), c_2 (D,QU,T), c_3 (CA,PC,E), c_4 (Q,F,A)
Order by $\max(c_1, c_2, c_3, c_4)$ ascend stop after k

We introduce a concept of calculation rate (cr) in this experiment in Eq.(4).

$$cr(i) = \frac{N_{actual}}{N_{total}} \quad (4)$$

Where i represents the i th factor. N_{actual} is the number of some factors that has been calculated. $N_{total} = n|D|$ .We have 3 user-defined factors c_2, c_3, c_4 in this experiment, so $N_{total} = 3000$. OP algorithm is applied to this search. The result is shown in as Figure 4.where cr is y axis, k is x axis.

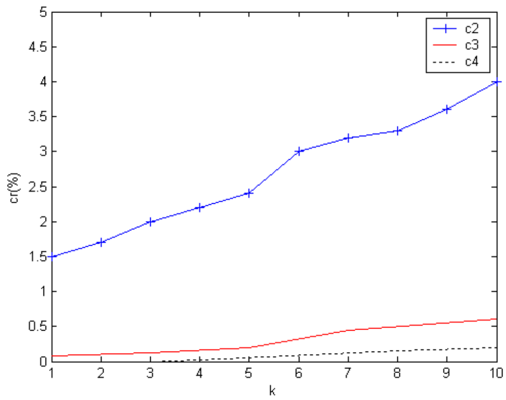
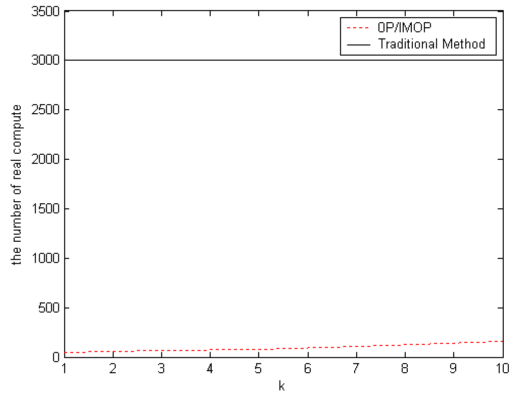


Figure 4. Calculation rate (cr) & the number of returning results k

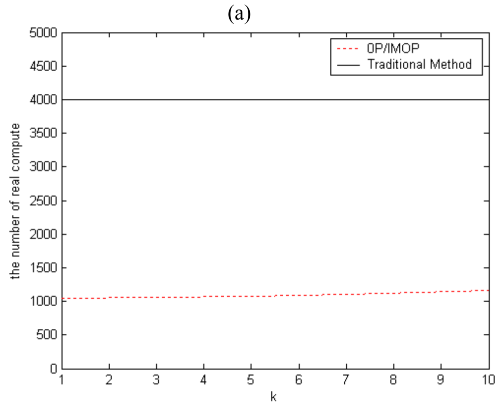
As shown in the Figure 4, when k=10, the cr value of c_2 is only 4%, and c_3 and c_4 are less than 1%. Figure 5(a) indicates that in traditional method (according to Eq (2)), the number of calculation is 3000, and whatever the k value is. Comparatively, the number of calculation is only 155 in OP or IMOP algorithm when k=10.

Assume c_1, c_2, c_3 and c_4 in Table 5 are all user-defined factors. We must calculate a certain factor for all the enterprises first. The number of the procedure calculations of traditional method is 4000. When k=10, it is 1155 in OP or IMOP algorithm as shown in Figure 5(b).

In the second experiment, we make a comparison of the accuracy between OP and IMOP when a leading enterprise emphasizes some factors. For example, assume we have 6 object enterprises, 3 factors, and leading enterprise concern c_3 most and need 2 enterprises. Obviously, the best result is $\{c, d\}$. The simulation data is shown in Table VI.



(a)



(b)

Figure 5. Real calculation work and the number of returning results k. (a)with three user-defined factors (b) with four user-defined factors

TABLE VI.

DATA OF SECOND EXPERIMENT

OID	c_1	c_2	c_3	$f(c_1, c_2, c_3)$
a	0.6	0.6	0.3	0.6
b	0.7	0.95	0.2	0.95
$c*$	0.8	0.8	0.2	0.8
$d*$	0.9	0.85	0.2	0.9
e	0.93	0.5	0.5	0.93
f	0.97	0.7	0.4	0.97

The result of OP is $\{a, c\}$. IMOP algorithm considers the factors concerned by the leading enterprise first. So the results is $\{b, c\}$. Then, it adjusts the result according to SM algorithm which is included in IMOP. In the algorithm of SM, starting with the second result c which is in $\{b, c\}$, it will upward seek an object which is equal to c's scoring value. So, we get b. Similarly, it will seek an object which is equal to c's scoring value downward, and get d. After that, calculate three objects $\{b, c, d\}$ in OP algorithm again which is included in IMOP. The final result is $\{c, d\}$. IMOP improves the selection accuracy accordingly.

We also compare the number of computations of algorithm OP, IMOP, and the traditional method in Table VI. The result is shown at left bars (group 1) in Figure 6. We employ *sr* which is a rate between the actual computation and complete computation which is the computation of traditional method. We can find the computation work of IMOP larger than OP a little, but smaller than a traditional method at left bars in Figure 6 when the computation of tradition method is 18.

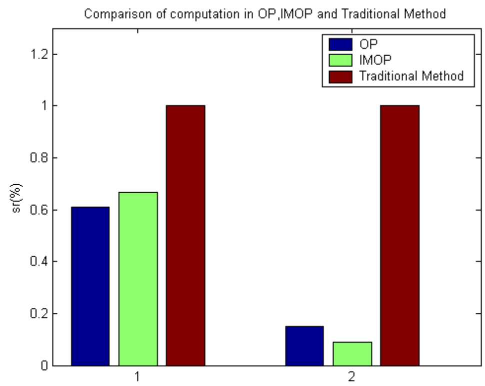


Figure 6. Comparison of computation in OP, IMOP and Traditional Method

Intuitively, the calculation depends on the value of k and the size of procedures. The larger the value of k is, the larger the calculation is, and the more the number of procedures is, the larger the calculation is. According to Eq.(1), we can get the computation of the OP algorithm where n_i is the times for calculating c_i. The computation of IMOP algorithm can get from Eq.(5) which is composed of two parts.

$$PE(IMOP) = \sum_{i=1}^{l} n_i' e_i + \sum_{i=1}^{m-l} n_i'' e_i \qquad (0 \le n_i' \le N)$$
$$(0 \le n_i'' \le s_{up} + s_{down} + 1) \tag{5}$$

Where l is the number of factors which the leading enterprise may concern, and $l < m$. s_{up} and s_{down} are respectively the number of the upward and downward objects which have the same value with the kth. To simplify the analysis, we assume any e_i =1, then, according Eq.(1) and Eq.(5), we can get Eq.(6) and Eq.(7) as follows.

$$PE(OP) = \sum_{i=1}^{m} n_i \tag{6}$$

$$PE(IMOP) = \sum_{i=1}^{l} n_i' + \sum_{i=1}^{m-l} n_i'' \tag{7}$$

Where Eq.(6) will return k results and Eq.(7) return $s_{up}+1$ results, and $s_{up}+1<k$ obviously. The computation relates to the order of procedures which are in $Z \subseteq \{c_1, \cdots, c_n\}$, namely, different orders will get different computation [18]. For making comparison analysis easier, we set the order of first l procedures in OP the same as the order of procedures in IMOP, namely

$\{c_1,\cdots,c_l,\cdots,c_m\}$ .so we can infer that the computation in first l procedures is

$$n_i \approx n_i\,' \ , \ \ i = 1,\cdots l \tag{8}$$

So, Eq.(6) and Eq.(7) can be derived that

$$PE(OP) - PE(IMOP) = \sum\nolimits_{i=1+l}^{m} n_i - \sum\nolimits_{i=1}^{m-l} n_i\,''$$

$$(0 \le n_i \le N)\,,(0 \le n_i\,'' \le s_{up} + s_{down} + 1) \tag{9}$$

Next, we compare $\sum\nolimits_{i=1+l}^{m} n_i$ and $\sum\nolimits_{i=1}^{m-l} n_i\,''$. From Figure 4, we can find when the order of one procedure is later in Z , the frequency of its calculation is less. The maximum value of $\sum\nolimits_{i=1}^{m-l} n_i\,''$ is $\sum\nolimits_{i=1}^{m-l} (s_{up} + s_{down} + 1)$ and the minimum is 0. Because the value of s_{up}, s_{down}, m and l is limited, so $\sum\nolimits_{i=1}^{m-l} n_i\,''$ is not very big. However, $\sum\nolimits_{i=1+l}^{m} n_i$ is related to k . The larger k is, the larger $\sum\nolimits_{i=1+l}^{m} n_i$ is. So we can infer that when k is small, the result of Eq.(9) is a negative number and when k is not small, the value of Eq.(9) is a positive number. Assume the enterprise items=10000, $k = 20$, $m = 10$ and $l = 3$ From right bars (group 2) in Figure 6, , the computation work of IMOP is smaller than OP, where the complete computation is 10000.

VI. Conclusions

Establishing the partnership in dynamic alliance is a popular issue of supply chain management. In this paper, we explore the methods on user-defined procedures to speed up the process of selection. We also propose the algorithm of OP which reduces the number of calculations on user-defined procedures as much as possible. And the algorithm of IMOP improves the accuracy of the partner selection based on OP algorithm.

We have implemented the sorting model in this paper. The simulations demonstrate that these algorithms can improve the process of choice partners efficiently and save cost effectively. It can eliminate much computation on user-defined procedures for k=10. And from the second experiment, we can find that the calculation of IMOP is almost as the calculation of OP when k is not very large. In this way, the leading enterprise is to concern these candidates which are worthy to cooperation.

Acknowledgement

This work was supported in part by the Program for Zhejiang Leading Team of Science and Technology Innovation, and the Science and Technology Planning Projects of Zhejiang Grants 2010C33045 and 2011C13006-1. This work was also industrially supported by the National Development and Reform Commission, China under Special Grants "The Operation System of Multimedia Cloud Based on the Integration of Telecommunications Networks, Cable TV Networks and the Internet".

References

[1] Wikipedia. http://www.wikipedia.org/.

[2] D. W. Wang, K. L. Yung, W. H. Ip. “A heuristic genetic algorithm for subcontractor selection in global manufacturing environment”.IEEE Transaction on System, Manufacturing and Cybernetics-Part C: Applications and Reviews, 2001, 31(2):189-198.

[3] X. N. Chu, S. K. Tso, W. J. Zhang,et al. “Partnership synthesis for virtual enterprise”.Journal of Advanced Manufacturing Technology, 2002,19(5):384-391.

[4] L. Mikhailov. “Fuzzy analytical approach to partnership selection in formation of virtual enterprise”.OMEGA, 2002, 30(5):393-401.

[5] J. Mun, M. Shin, K. Lee, et al. “Manufacturing enterprise collaboration based on a goal-oriented fuzzy trust evaluation model in a virtual enterprise”. Computers & Industrial Engineering, Volume 56, Issue 3, April 2009, Pages 888-901.

[6] W. Zhang, L. Liu, Y. C. Zhu. “Using fuzzy cognitive time maps for modeling and evaluating trust dynamics in the virtual enterprises”. Expert Systems with Applications, Volume 35, Issue 4, November 2008, Pages 1583-1592.

[7] B. Blankenburg , M. He, M. Klusch ,et al. “Risk bounded formation of fuzzy coalitions among service Agents”.Lecture Notes in ComputeScience,2006,4149 :332-346

[8] M. Ono, M. Ishizuka. “Learning and evolution affected by spatial structure”. Computer Science, 2006(7):651-660.

[9] A. S. Belenky. “Cooperative games of choosing partners and forming coalitions in the marketplace”.Mathematical and Computer Modelling, 2002, 36(1):1279-1291.

[10] R. Fagin,“Combining Fuzzy Information from Multiple Systems,”Proc. ACM Symp. Principles of Database Systems, 1996.

[11] R. Fagin, A. Lotem, and M. Naor, “Optimal Aggregation Algorithms for Middleware,” J. Computer and System Sciences,vol. 66, no. 4, 2003.

[12] N. Bruno, L. Gravano, and A. Marian, “Evaluating Top-k Queries over Web-Accessible Databases,” Proc. Int’l Conf. Data Eng., 2002.

[13] A. Marian, N. Bruno, and L. Gravano, “Evaluating Top-k Queries over Web-Accessible Databases,” ACM Trans. Database Systems,2004.

[14] S.-W. Hwang and K.C.-C. Chang, “Optimizing Access Cost for Top-k Queries over Web Sources: A Unified Cost-Based Approach,”Proc. Int’l Conf. Data Eng., 2005.

[15] A. Natsev, Y.-C. Chang, J.R. Smith, C.-S. Li, and J.S. Vitter,“Supporting Incremental Join Queries on Ranked Inputs,” Proc. 27th Int’l Conf. Very Large Data Bases, 2001.

[16] M. Theobald, G. Weikum, and R. Schenkel, “Top-k Query Evaluation with Probabilistic Guarantees,” Proc. 30th Int’l Conf. Very Large Data Bases, 2004.

[17] P. Cao and Z. Wang, “Efficient Top-k Query Calculation in Distributed Networks,” Proc. 23rd Ann. ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing (PODC ’04), 2004.

[18] K.C. Chang and S.W. Hwang, “Probe Minimization by Schedule Optimization:Supporting Top-k Queries with Expensive Procedures,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 5, May 2007.

[19] L. Gravano, S.Chaudhuri, 1999. “Evaluating Top-k selection queries”. In Proceedings of the 25th International Conference on Very Large Databases (VLDB’99).

[20] L. Gravano, N. Bruno, A. Marian. 2002b. “Evaluating Top-k queries over web-accessible databases”. In Proceedings of the 2002 International Conference on Data Engineering (ICDE’02).

Bishan Ying (1968-), was born Ningbo, Zhejiang. He is a Ph.D. and senior engineering. His research interests include Cloud Computing, Ubiquitous Network and the application of Big Data. Now, he is a Director of Strategic Planning in Wasu Media Network Co.