J. Software Engineering & Applications, **2010, 3, 548-555**
doi:10.4236/jsea.2010.36063 Published Online June 2010 (http://www.SciRP.org/journal/jsea)

Tabu Search Solution for Resource Confidence Considered Partner Selection Problem in Cross-Enterprise Project

Hanchuan Xu, Xiaofei Xu, Ting He

School of Computer Science & Technology, Harbin Institute of Technology, Harbin, China.
Email: {xhc, xiaofei, xuantinghe}@hit.edu.cn

Received April 1st, 2010; revised April 21st, 2010; accepted April 23rd, 2010.

ABSTRACT

Cross-enterprise project is the main implementation form in multi enterprises collaborative production environment. Minimizing the risk of failure and tardiness caused by the uncertainty of partner's resources in partner selection is the key problem to ensure success in Cross-enterprise project. In this paper, considering the factors and constraints of sub-project processing times, precedence of sub-project and project due date, especially the resource confidence, a 0-1 *integer programming model was presented with the objective to minimize the risk of failure and the tardiness of the project. A project scheduling algorithm was designed to search and evaluate selection solutions, and the project scheduling algorithm was embedded into a Tabu search algorithm to solve the model. Simulation experiments and comparisons with other algorithms showed that the proposed approach was possible to find the optimal solution with a faster speed and higher probability.*

Keywords: *Cross-Enterprise Project, Partner Selection, Resource Confidence, Tabu Search*

1. Introduction

To be competitive in the global manufacturing environment with the rapidly increasing competitiveness, strategic collaborations between enterprises are needed in order to meet the market's requirements for quality, responsiveness, and customer satisfaction. Dynamic alliance, virtual enterprise, extended enterprise, and supply chain are the major management philosophies for multi enterprises collaborative production environment. Cross-enterprise project (CEP) management pattern arose as the main implementation form in these management philosophies. CEP is the formation of closer co-ordination in the design, development, costing and the co-ordination of the respective manufacturing schedules of co-operating independent manufacturing enterprises and related suppliers [1,2]. There are four stages in CEP life cycle: formation, operation, evolution and dissolution. A major issue in the formation phase is to select appropriate partners and allocate tasks between partners. During this process, the core enterprise comprehensively evaluates partners according to cost, quality, credit, delivery time, etc., and then based on certain criteria, find the best combination of partners to collaborate to complete the project.

Partner selection has attracted much research attention recently, because it is an important function for information management systems, such as project management (PM), enterprise resource planning (ERP) and supply chain management (SCM). Most of researches about the partner selection problem are based on qualitative analysis methods. However, quantitative analysis methods for partner selection are still insufficient. Many operation research methods, such as analytic hierarchy process (AHP), analytic network process (ANP) and fuzzy synthetic evaluation are widely used to the problem, but mathematical models and optimization methods for partner selection are still a challenge [3,4]. In partner selection process, although there are many factors needed to be considered such as friendship, credit, quality, and reliability, the cost and completion time are the two key factors and focused on by most researchers.

In CEP, the resources belong to different partners which often undertake other projects at the same time, so the available production capacity of partner will be tight during some periods. In addition, unforeseen exceptions also inevitably occur. All of these make resources have a certain degree of uncertainty. Although there are some constraints between enterprises such as contracts, the case that partners can't provide promised resources on time or in right quality is not able to be completely avoided. When this happens, it will seriously affect the production schedule of core enterprise and even cause the whole cross-enterprise project failed. So the confidence level of resource that can be provided by partners on time and in right quality which we defined as *resource confidence* is a very important factor in the partner selection problem. However, most researchers mainly take into account the cost and completion time, and the objective is to minimize the total cost of the project or the project duration. The factor of resource confidence is neglected in most researches.

Yannis and Andreas [5], Sha and Che [6], Mikhailov [7] and other researchers proposed some models and approaches for partner selection by establishing a CEP, where cost, time and distance were considered. However, the other two important factors, the resource confidence and the precedence of sub-project, were not considered in their papers. As Wang *et al.* [8] indicates, in the cooperation relationship of sub-projects contracted by partners, it may be represented by an activity network with precedence. Thus, the problem could be considered as a partner selection problem embedded with project scheduling. Naiqi *et al.* [9] considered the completion time as a constraint and modeled the partner selection problem by an integer programming formulation to minimize the manufacturing cost. The formulation was then transformed into a graph-theoretical formulation and a 2-phase algorithm was developed to solve the problem. Wang *et al.* [8] took into consideration the factors of cost, completion time and precedence of sub-project, described the partner selection problem with a 0-1 integer programming formulation to minimize the total cost of the project. They then developed a fuzzy decision embedded heuristic genetic algorithm to find the solution for partner selection. The experiments showed that the algorithm was possible to quickly achieve optimal solution for large size problems. Taking into account the same factors and objective as Wang [8], W. H. IP *et al.* [10] and Zhibin *et al.* [11] separately proposed branch and bound solutions for partner selection problem in virtual enterprises and their solutions were especially effective to medium or small size problems. In all these papers mentioned above, their objectives were minimizing the total cost of project and didn't consider the impact of resource confidence to project implementation risk. To minimize risk in partner selection and ensure the due date of a project in virtual enterprise, W. H. IP *et al.* [12] described and modeled a risk-based partner selection problem. They developed a rule-based genetic algorithm with embedded project scheduling to solve the problem. In their paper, they assumed that each candidate partner had a fail probability of its contracted sub-project. In fact, the fail probability of the sub-project closely related to whether the partner's available resources were tight or not in the duration of the sub-project. The tighter the resources are, the higher the fail probability is, and vice versa. However, the production load of partner is varied in different periods, so the available resource and fail probability are also different. They used a whole fail probability for all periods and didn't consider the difference of fail probability in different periods.

To solve the partner selection problem in CEP, we first describe it with a 0-1 integer programming model considering the factors of process time, precedence of sub-projects, and resource confidence. Then a project scheduling algorithm is proposed to calculate the project completion time and the time window of each sub-project under a feasible solution. From this, we embed the project scheduling algorithm into a Tabu search algorithm to obtain the optimal partner selection solution. The computation results showed that the proposed approach is efficient to achieve the optimal solution.

The paper is structured as follows. In Section 2 the problem and the model are introduced. Then the solution space reduction method and the project scheduling algorithm to evaluate selection solution are presented in Section 3. The Tabu search algorithm embedded the project scheduling is presented in Section 4. Section 5 reports an experimental example and computational results obtained by testing the algorithm on some test instances. Finally, Section 6 presents our overall conclusions.

2. Model for Partner Selection Considering Resource Confidence

The problem of partner selection for CEP considering resource confidence can be described as follows:

Assuming an enterprise (core enterprise) obtain a big project consisting of several sub-projects. It is not able to complete the big project by itself from its own resources and has to find some partner enterprises to collaboratively finish the project. The partner selection procedure is divided into two phases. Firstly, the enterprise determines the payment and some basic requirements for the process time and quality of each sub-project. The partners who can accept the conditions will propose the process time they need to finish the sub-project according to their own capacity. They constitute the candidate partner set. In the second phase, the enterprise comprehensively evaluates all candidates and calculates the confidence of resource provided by the partner for the sub-project. At last the enterprise selects the most appropriate partner for each sub-project. There exists plenty methods to evaluate the individual candidate partner and calculate the resource confidence, for an extensive review we refer to MALONI and BENTON [3] and BOER *et al.* [4]. In this paper, we just focus on the second phase, *i.e.*, how to select partners according to the resource confidence.

Based on the risk-based partner selection model presented by W. H. IP *et al.* [12] and considering the resource confidence, we present a 0-1 integer programming model MPCPS for the partner selection problem. Suppose that the project consists of n sub-projects, there are precedence relationship between these sub-projects and they form a precedence activity network H. If sub-project k can only begin after the completion of sub-project i, we call sub-project i as the immediate predecessor of sub-project k and define the connected sub-project pair by $<i,k>\in H$. For the convenience of description, we label these sub-projects such that $i<k$. Without the loss of generality, the final sub-project is labeled as sub-project n. Thus, we can define that the completion time of final sub-project c_n is the completion time of the project. Each sub-project has some candidate partners, for sub-project i, $i=1,2,...,n$, there are m_i candidate partners, and for the candidate partner j of sub-project i, its processing times is q_{ij} periods. The resource confidence of candidate j to sub-project i in period t is noted as $d_{ij}(t)$ and there $d_{ij}(t)\in(0,1]$. The due date of the project is *D*. To simplify the problem, we assume that the core enterprise will select only one candidate to undertake one sub-project.

The objective is to select the optimal combination of partners for all sub-projects in order to maximize the whole resource confidence of the project and to finish the project within the due date.

The following decision variable is defined.

$$x_{ij}(t)=\begin{cases}1 & \text{candidate } j \text{ is selected to sub-project } i \text{ at period } t\\ 0 & \text{otherwise}\end{cases}$$

Then the problem can be modeled as follows:
MPCPS

$$\max_{x} F_d(x)=\prod_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{t=1}^{c_n}x_{ij}(t)[\frac{1}{q_{ij}}\sum_{\tau=t}^{t+q_{ij}-1}d_{ij}(\tau)]+\beta(1-[[c_n-D]^{*}]^{-}) \tag{1}$$

$$\text{s.t.}\quad \sum_{j=1}^{m_i}\sum_{t=1}^{c_n}x_{ij}(t)=1;i=1,2,...,n \tag{2}$$

$$(t+q_{ij})\sum_{j=1}^{m_i}\sum_{t=1}^{c_n}x_{ij}(t)\le t\sum_{j=1}^{m_k}\sum_{t=1}^{c_n}x_{kj}(t);t=1,2,...,c_n,\forall<i,k>\in H \tag{3}$$

$$\sum_{j=1}^{m_n}\sum_{t=1}^{c_n}(t+q_{nj})x_{nj}(t)=c_n \tag{4}$$

$$x_{ij}(t)\in\{0,1\}\forall i,j,t \tag{5}$$

where $[x]^{+}$ stands for $\max\{0,w\}$, $[y]^{-}$ stands for $\min\{1,y\}$ and β is the tardiness penalty coefficient. Formula (1) is the objective function, where $\frac{1}{q_{ij}}\cdot\sum_{\tau=t}^{t+q_{ij}-1}d_{ij}(\tau)$ is the mathematical expectation of resource confidence for candidate j of sub-project i in the q_{ij} continuous periods, and abbreviated as E_{ij} bellow. Constraint (2) ensures that each sub-project will be contracted to only one partner and constraint (3) is the precedence constraint of sub-projects. Constraint (4) gives the method to calculate the completion time of the project.

It is obvious that the operator $[x]^{+}$ and $[y]^{-}$ are non-analytical and the objective function is not continuous and differentiable, so it is difficult to treat the model by traditional mathematical programming methods. Therefore, we develop a project scheduling embedded Tabu search (PSTS) algorithm to solve this problem.

3. Solution Space Reduction and Evaluation Algorithm of Solution

3.1 Solution Space Reduction

The partner selection modeled as MPCPS is a complex combinatorial optimization problem. The number of feasible solutions (solution space) is very large, even for a small-scale problem. To simplify the solving process, W. H. IP *et al.* [12] defined the concept of *inefficient candidate* and proved that the optimal solution consists without any inefficient candidate. To efficiently reduce the solution space, all inefficient candidates can be ignored in the solving process without losing the optimal solution. Based on the definition presented by W. H. IP [12] and considering the characteristics of the model MPCPS, we define the *inefficient candidate* to our model as follows.

Definition 1. For the two candidates j and k of sub-project i, if $\forall t\in[ES_i^{\min},LF_i^{\max}]$ $\forall t\in[ES_i^{\min},LF_i^{\max}]$, $q_{ik}\le q_{ij},d_{ik}(t)>d_{ij}(t)$, or $q_{ik}<q_{ij},d_{ik}(t)\ge d_{ij}(t)$, then the candidate j is called inefficient candidate.

It is easy to prove that there exists at least one optimal solution which doesn't include any inefficient candidate. Therefore all the inefficient candidates are not considered in the model without loss of the optimal solution. In definition 1, $ES_i^{\min}$ is the earliest possible start time and $LF_i^{\max}$ is the latest allowable finish time of sub-project i. The longest possible time window $[ES_i^{\min},LF_i^{\max}]$ of sub-project i is only a part of the whole project time window $[1,c_n]$, so to judge whether a candidate is inefficient or not, we only need to do the judgment in time window $[ES_i^{\min},LF_i^{\max}]$ according to definition 1, rather than in the whole project time window. This method improves the satisfiability of definition 1 and the effect of solution space reduction. Let $q_i^{\min}$ and $q_i^{\max}$ be the shortest and longest process time of sub-project i respectively, $q_i^{\min} = \min\{q_{i1,}q_{i2,...,}q_{im_i}\}$, $q_i^{\max} = \max\{q_{i1,}q_{i2,...,}q_{im_i}\}$, then $ES_i^{\min}$ and $LF_i^{\max}$ can be calculated with $q_i^{\min}$ and $q_i^{\max}$. This is a project scheduling problem with the objective of minimizing project makespan, *i.e.*, $PS|prec|C_{\max}$ and there exists some polynomial time algorithms [13].

3.2 Project-Scheduling-Based Solution Evaluation Algorithm

In our TS algorithm, the natural number string is selected as code description. Let $x = [x_1, x_2, ..., x_n]$, where x_i is an integer between 1 and m_i. This stands for that candidate x_i is selected for sub-project i. Thus, $x = [x_1, x_2, ..., x_n]$ is called a selection. For example, [3 4 1 5 2 3 4] is a partner selection of a project with 7 sub-projects. In the selection, the candidate 3 is selected for the sub-project 1, and the candidate 4 is selected for the sub-project 2, and so on.

Once a selection fixes the candidates for all sub-projects, to obtain the object function value, a project-scheduling-based solution evaluation algorithm *PSLP* can be done to calculate the variable values of c_n and E_{ij}, and also the object function value. The procedure of the algorithm PSLP is described as follows:

Algorithm PSLP

Step 1: Calculate the earliest starting time ES_i and the earliest finish time EF_i of each sub-project $i(i = 1,2,...,n)$. If do not exist $(k,i) \in H, \forall k \in \{1,2,...,n\}$, then $ES_i \leftarrow 0$; Else, $ES_i \leftarrow \max\{EF_k, \forall(k,i) \in H\}$. $EF_i \leftarrow ES_i + q_{ix_i}$.

Step 2: Calculate the project completion time c_n. Let $LF_n \leftarrow EF_n$, $LS_n \leftarrow ES_n$, $c_n \leftarrow LF_n$.

Step 3: Calculate the latest starting time LS_i and the latest finish time LF_i of each sub-project $i(i = 1,2,...,n)$. If do not exist $(i,k) \in H$, $\forall k \in \{1,2,...,n\}$, then $LF_i \leftarrow LF_n$; Else, $LF_i \leftarrow \min\{LS_k, \forall(i,k) \in H\}$. $LS_i \leftarrow LF_i - q_{ix_i}$.

Step 4: Calculate the float time FF_i of each sub-project $i(i = 1,2,...,n)$ and judge whether it is critical sub-project or not. $FF_i \leftarrow LS_i - ES_i$.

If $FF_i = 0$, then $U_c \leftarrow U_c \cup \{i\}$, Else $U_{nc} \leftarrow U_{nc} \cup \{i\}$.

Step 5: For each non-critical sub-project in U_{nc}, calculate the maximum mathematical expectation of resource confidence by the step-by-step right-shifted procedure.

Step 5.1: If $U_{nc} = \varnothing$, then go to Step 6; Else get the non-critical sub-project i from U_{nc} which has the maximum earliest start time. $U_{nc} \leftarrow U_{nc} \setminus \{i\}$, $Pos \leftarrow 0$, $E_{ix_i} \leftarrow 0$.

Step 5.2: For $j = 0$ to $FF_i - 1$ Do If $E_{ix_i} < \frac{1}{q_{ix_i}} \sum_{k=j}^{j+q_{ix_i}-1} d_{ix_i}(ES_i + k)$ then $E_{ix_i} \leftarrow \frac{1}{q_{ix_i}} \sum_{k=j}^{j+q_{ix_i}-1} d_{ix_i}(ES_i + k)$, $Pos \leftarrow j$ End For.

Step 5.3: Adjust the float time of each immediate preceding non-critical sub-project of sub-project *i*. For $\forall k \in U_{nc}$, and $(k,i) \in H$, let $FF_k \leftarrow FF_k + Pos$, go to Step 5.1.

Step 6: For each critical sub-project $i \in U_c$, calculate the maximum mathematical expectation of resource confidence, $E_{ix_i} \leftarrow \frac{1}{q_{ix_i}} \sum_{k=0}^{q_{ix_i}-1} d_{ix_i}(ES_i + k)$.

Step 7: Calculate the objective function value. $F_d(x) \leftarrow \prod_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{t=1}^{c_n} x_{ij}(t)E_{ij} + \beta(1-[[c_n - D]^+]^-)$

Step 8: Over.

In the PSLP algorithm, the time window of each sub-project is calculated first. There, $[ES_i, ES_i + 1, ..., LS_i]$ and $[EF_i, EF_i + 1, ..., LF_i]$ represent the starting time window and finish time windows of sub-project *i*, respectively. In addition, the project critical path, the critical sub-project set U_c, the non-critical sub-project set U_{nc} and the float time of each sub-project can also be determined. In the second part, based on the idea of solving the resource levelling project scheduling with fixed project duration problem, and considered the characteristics that the resource confidence is various in each period and non-critical sub-project has float time, a step-by-step right-shifted procedure is employed to find the time section in which the non-critical sub-project has the greatest mathematical expectation of resource confidence. In the procedure, non-critical sub-projects are dealt with in accordance with descending order of the earliest start time. At the last part, the objective function value for a selection is obtained.

4. Tabu Search Algorithm Design

The Tabu search (TS) algorithm is an effective method for solving large-scale combination optimization problem. The TS algorithm can overcome the shortcomings of some common heuristic algorithms which only adapt to special problems and easily sink into local optimal solutions. TS has been used on an increasing number of practice problem and has proven to be effective [14,15].

4.1 The Initial Solution

The initial solution is very important to TS and a good one is very useful to improve the constringency speed to optimal solution. Considering the fact that, in the widest feasible time window $\left[ES, ES+1, ..., LF\right]$ of the sub-project, the greater the resource confidence mathematical expectation of the candidate is, the higher the probability of being selected is, the initial solution generation procedure *PINI* is designed as the following steps.

Step 1: From sub-project $i=1$ to *n*, calculate the widest feasible time window $\left[ES_i, LF_i\right]$.

Step 2: From sub-project $i=1$ to *n*, find candidate x_i which has the greatest mathematical expectation of resource confidence in all the candidates of sub-project i.

Step 3: output the initial solution $x=[x_1, x_2, ..., x_n]$.

It is obvious that the initial solution is feasible.

4.2 Neighbourhood Structure and Candidate Solution

Considering the natural number string employed in this algorithm, neighbor is defined as all feasible solutions obtained through changing the value of one bit of the current solution, *i.e.*, changing a candidate partner of a sub-project. Moving from the current solution to a solution in the neighborhood is called a move. Therefore, one step in a move can change only one partner of the current solution. Let NB be the neighbor set. Evaluation value of each solution in NB can be calculated by the PSLP algorithm. The solution in NB will be selected as the candidate solution with meeting the conditions that it has the greatest evaluation value and the move from the current solution to it is not in the Tabu list.

In the solution evaluation algorithm PSLP described in 3.2, Step 5 is to precisely calculate the maximum mathematical expectation of resource confidence and the step-by-step right-shifted procedure has high CPU time cost. In fact, if a solution causes the whole project delayed, its evaluation value will be penalized with the penalty factor β in the Formula 1. Therefore, the solution has very low probability to be selected as the candidate solution. From this point of view, the following tardiness penalty evaluation procedure of candidate solution *CSTPE* is designed.

Procedure CSTPE:

Step 1: Implement the Steps 1-3 in the algorithm PSLP to calculate the time window of each sub-project and the completion time of the whole project.

Step 2: If the project isn't finished within the due date in the solution, then the Step 5 of PSLP will not be run and the subsequent steps will run with the time window $\left[ES_i, LF_i\right]$ obtained by the Steps 1-3, else the total PSLP algorithm will be run.

Experiments show that this candidate solution evaluation procedure can significantly reduce the time cost and still can find the optimal solution with high probability. The detailed analysis is described in the Section 5.

4.3 Tabu List

The Tabu list (TSL) is composed of a two-dimensional integer array. The number of rows is the length of the Tabu list, the first column is the code of the sub-project, and the second column is the code of the candidate partner corresponding to the sub-project in the first column. The code for every row records a solution in the neighborhood that has been deleted in recent movements. TSL is renewed according to the criterion of first in, first out.

4.4 Longer-Term Tabu List and Tabu Relaxation

To avoid getting into the local optimum and the cycling, two special techniques, longer-term Tabu list (TTL) and Tabu relaxation, are used. TTL is created to dynamically forbid moving overactive nodes in order to get diversification and help to prevent cycling. The algorithm incorporates a move frequency table to record the move frequency of each sub-project. When a sub-project's partner is changed, its move frequency is incremented by 1. If a sub-project's partner *x* has been moved more than two times and TTL is not full, it will be put into TTL. If TTL is full and if some sub-project's candidate partner *y* already in TTL has a lower move frequency than *x*, *y* will be dropped and *x* will be added into TTL.

Another technique used is the relaxation of Tabu lists. If a given number of iterations (*relaxed_tries*) has elapsed and TTL is full since the last best solution was found, which means the search process has plunged into a local optimal solution or a cycling, both TSL and TTL are emptied and using the current solution as the initial solution to continue the search. Relaxation of the Tabu lists will change the neighborhood of the current solution dramatically, which will lead to a rapid downhill movement and may lead to new search spaces.

4.5 The Aspiration Criterion and Stopping Rule

The Tabu status of a move can be overruled if a solution is feasible and is better than any feasible solution known so far.

In our PSTS algorithm, there are two ways of controlling the execution time: the maximum total number of iterations (max_*tries*) and the maximum number of iterations without improvement of the best known feasible solution (max_*unchanged*). The execution of the algorithm is stopped when the number of iterations max_*tries* and max_*unchanged* are both attained, or when the number of iterations doubles max_*tries*. Therefore, the total number of iterations is not known in advance, depending on the evolution of the search. The combination of the max_*tries* and the max_*unchanged* as stopping criterion allows the search to continue if the algorithm is exploring a new promising region. Obviously, to give time for improvement after the restart, the max_*unchanged* should be greater than the *relaxed_tries*.

4.6 Global Description of the Algorithm

Algorithm PSTS:

Step 1: Specify the parameters. Set initial values of max_*tries*, max_*unchanged*, relaxed_*tries*, set iteration counter to $tries \leftarrow 0$, iteration counter for no improvement to $unchanged \leftarrow 0$.

Step 2: Generate an initial solution x using the algorithm PINI and calculate the evaluation value $F(\mathrm{x})$ of x using the CSE procedure. Let the current best solution $x^* \leftarrow x$, the best evaluation value $F^* \leftarrow F(\mathrm{x})$.

Step 3: $tries \leftarrow tries + 1$, if $tries > \max\_tries$ and $unchanged > \max\_unchanged$, or if $tries > 2 \times \max\_tries$, then go to Step 7, else go to Step 4.

Step 4: If $unchanged > \mathrm{relaxed}\_tries$, and *TTL* is full, then empty *TSL* and *TTL*.

Step 5: Find the neighborhood *NB* of x, calculate the candidate solution x_{NB} and the evaluation value $F(x_{NB})$. Calculate the best Tabu solution x_{TSL} and the corresponding evaluation value $F(x_{TSL})$ from *TSL*.

Step 6: Considering the TSL,TLL and the aspiration criterion, generate the new solution x from x_{NB} and x_{TSL}. If $F^* < \max\{F(x_{NB}), F(x_{TSL})\}$, $x^* \leftarrow x$, $F^* \leftarrow \max\{F(x_{NB}), F(x_{TSL})\}$, $unchanged \leftarrow 0$; else $unchanged \leftarrow unchanged + 1$. Go to Step 3.

Step 7: Output x^* and F^* is the optimal solution, algorithm is over.

If the CSTPE procedure is used to evaluate candidate solution, then the PSTS will be named as PSTS-P. Otherwise, if just the completed PSLP is used, the PSTS will be named as PSTS-NP.

5. Experiments Analysis

The algorithm was coded by JAVA and run on a Pentium Dual 2.2 GHz PC.

The example is a project that consists of 14 sub-projects and the core enterprise calls tenderers for the sub-projects. The precedence relationship represented by the active-on-node network is shown in **Figure 1.** The numbers in the parentheses are the number of candidates, the shortest process time and the longest process time, respectively. The project's due date is 24 periods and each sub-project has 3 to 6 candidates. The solution space size is 1.05×10^8. Through identifying and removing the inefficient candidates according to Definition 1, the size is reduced to 2.83×10^7. Different candidates may have different process time to the sub-project that they bid for. The resource confidence of the candidate in each period $d_{ij}(t)$ is calculated by the fuzzy comprehensive evaluation method, and where $d_{ij}(t) \in (0,1]$. For the limitation of the paper size, the detailed data is omitted.

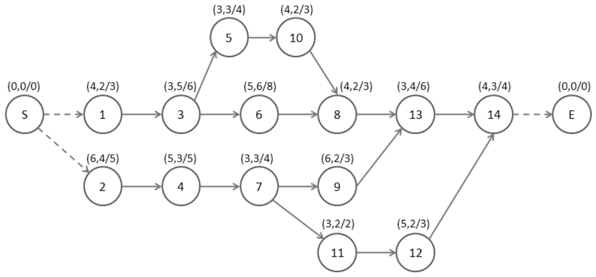

Figure 1. Example of a project represented by active network

The setting for the values of parameters is important for the efficiency of TS algorithm. In our algorithm, the "Best_rate" is used to evaluate and adjust the values of parameters, where "Best_rate" is the rate to reach the optimal solution in a certain number of runs. Based on the algorithms of IP WH *et al.* [10] and Zhibin *et al.* [11], considering the characteristics of our problem, a branch- bound algorithm (B & B) is designed to calculate the optimal solution. For different scale problems, the algorithm was run a certain number times according to the scale of the problem with different random seeds for each parameter setting. Therefore, the parameters with the highest "Best_rate" are selected. To the example in **Figure 1**, the values of the parameters are "max_*tries* = 700", "max_*unchanged* = 80", "relaxed_*tries* = 60", the length of TSL is 18, and the length of TLL is 70. The result of the example is shown in **Table 1**, and the objective value is 0.241.

For testing the performance of the PSTS algorithm, we randomly generated some problems with different scales. The comparison results of the PSTS-P, PSTS-NP and B&B are shown in **Table 2**. Where "N" is the number of sub-projects, "size" stands for the size of solution space, "CPU time" is the average computation time of each running.

The B & B algorithm is a kind of exact algorithm and can always find the optimal solution (best rate is always 100%), but it needs much more running time to deal with large scale problems. The two recommended algorithms, PSTS-P and PSTS-NP, can achieve the optimal solution with a high best rate and the computation time doesn't

Table 1. The selected partner list, processing time and resource confidence

Sub-project no.	Selected partner code	Processing time	Start time	Finish time	Resource confidence
1	A2	2	0	2	0.82
2	B3	4	0	4	0.80
3	C1	5	3	7	0.76
4	D4	4	6	9	0.55
5	E1	3	9	11	0.60
6	F5	7	8	14	0.72
7	G2	3	11	13	0.58
8	H1	3	15	17	0.63
9	I5	2	15	16	0.69
10	J3	2	12	13	0.72
11	K2	2	14	15	0.83
12	L4	3	17	19	0.82
13	M1	5	18	22	0.91
14	N2	4	23	26	0.87

Table 2. The comparison of PSTS-P, PSTS-NP and B & B

N	Size	PSTS-P				PSTS-NP				B & B		
		CPU time (s)	Best rate (%)	Best	Avg	CPU time (s)	Best rate (%)	Best	Avg	CPU time (s)	Best rate (%)	Best & Avg
12	5.24×10^{8}	13.32	100	0.231	0.231	33.32	100	0.231	0.231	48.56	100	0.231
18	4.63×10^{10}	33.67	100	0.256	0.256	69.41	100	0.256	0.256	101.72	100	0.256
24	7.51×10^{14}	46.59	100	0.217	0.217	85.73	100	0.217	0.217	201.34	100	0.217
33	3.04×10^{18}	79.41	98	0.134	0.132	98.41	100	0.134	0.133	579.31	100	0.134
38	1.46×10^{21}	88.76	96	0.158	0.156	108.37	99	0.158	0.157	783.85	100	0.158
45	2.35×10^{22}	97.58	94	0.179	0.176	120.78	97	0.179	0.178	1084.67	100	0.179
48	6.35×10^{24}	113.62	91	0.092	0.089	156.39	96	0.092	0.090	9457.36	100	0.092
52	4.53×10^{26}	130.68	88	0.087	0.083	210.65	94	0.087	0.085	17613.09	100	0.087
58	8.63×10^{27}	149.31	86	0.083	0.078	290.25	93	0.083	0.080	29465.06	100	0.083
64	9.27×10^{35}	173.24	82	0.062	0.053	362.47	90	0.062	0.056	38280.77	100	0.062

grow fast with the problem size increase. For PSTS-P using the CSTPE procedure to evaluate candidate solutions, it can solve large problems faster; on the other hand, PSTS-NP has higher rate to obtain optimal solution. In practice, we can select the appropriate one from the two algorithms according to the different requirements of speed and best rate.

6. Conclusions

Partner selection is a complicated and practical problem in CEP. Minimizing risk caused by the uncertainty of partner's resources in partner selection and ensuring the due date of the project are important to the success of the CEP. This paper introduces a description of the partner selection problem in CEP. The concept of resource confidence is used to characterize the uncertainty of partner's resources, then the non-linear integer programming model (1-5) provides a formal description of the partner selection problem of CEP with considering resource confidence, where the following features different from conventional methods are considered:

1) The precedence activity network describing the precedence relationship between sub-projects

2) The resource confidence of each partner

A project scheduling embedded TS algorithm for the problem was proposed. Its two variants, PSTS-P and PSTS-NP, focus on computation speed and optimizing efficiency, respectively. The computation results show its potential to solve practical partner selection and project management problems.

The suggested future research work includes: a) Find a better way to share information between the core enterprise and partners, and research how to evaluate and calculate the resource confidence of partners more accurately. b) Research project planning model and algorithms for the CEP with considering resource confidence.

REFERENCES

[1] X. F. Xu, “Virtual Organization - the Enterprise Organization Form in the Future,” in Chinese, *China Mechanical Eengineering*, Vol. 7, No. 4, 1996, pp. 15-20.

[2] H. S. Jagdev and J. Browne, “The Extended Enterprise - A Context for Manufacturing,” *Production Planning & Control*, Vol. 9, No. 3, 1998, pp. 216-229.

[3] M. J. Maloni and W. C. Benton, “Supply Chain Partnerships: Opportunities for Operations Research,” *European Journal of Operational Research*, Vol. 101, No. 3, 1997, pp. 419-429.

[4] L. D. Boer, E. Labro and P. Morlacchi, “A Review of Methods Supporting Supplier Selection,” *European Journal of Purchasing & Supply Management*, Vol. 7, No. 2, 2001, pp. 75-89.

[5] A. Yannis, Hajidimitriou and A. C. Georgiou, “A Goal Programming Model for Partner Selection Decisions in International Joint Ventures,” *European Journal of Operational Research*, Vol. 138, No. 3, 2002, pp. 649-662.

[6] D. Y. Sha and Z. H. Che, “Virtual Integration with a Multi-Criteria Partner Selection Model for the Multi-Echelon Manufacturing System,” *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 7-8, 2005, pp. 793-802.

[7] L. Mikhailov, “Fuzzy Analytical Approach to Partnership Selection in Formation of Virtual Enterprises,” *Omega*, Vol. 30, No. 5, 2002, pp. 393-401.

[8] D. Wang, K. L. Yung and W. H. Ip, “A Heuristic Genetic Algorithm for Subcontractor Selection in a Global Manufacturing Environment,” *IEEE Transactions on SMC Part-C*, Vol. 31, No. 2, 2001, pp. 189-198.

[9] N. Q. Wu and P. Su, “Selection of Partners in Virtual Enterprise Paradigm,” *Robotics and Computer-Integrated Manufacturing*, Vol. 21, No. 2, 2005, pp. 119-131.

[10] W. H. Ip, D. Wang and K. L. Yung, “A Branch and Bound Algorithm for Sub-Contractor Selection in Agile Manu-Facturing Environment,” *International Journal of Production Economics*, Vol. 87, No. 2, 2004, pp. 195-205.

[11] Z. B. Zeng, Y. Li and W. X. Zhu, “Partner Selection with a Due Date Constraint in Virtual Enterprises,” *Applied Mathematics and Computation*, Vol. 175, No. 2, 2006, pp. 1353-1365.

[12] W. H. Ip, M. Huang, K. L. Yung, *et al*. ”Genetic Algorithm Solution for a Risk-Based Partner Selection Problem in a Virtual Enterprise,” *Computers & Operations Research*, Vol. 30, No. 2, 2003, pp. 213-231.

[13] P. Brucker, A. Drexl and R. Möhring, “Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods,” *European Journal of Operational Research*, Vol. 112, No. 1, 1999, pp. 3-41.

[14] F. Glover, “Tabu Search: Part I,” *ORSA Journal on Computing*, Vol. 1, No. 3, 1989, pp. 190-206.

[15] F. Glover, “Tabu Search: Part II,” *ORSA Journal on Computing*, Vol. 2, No. 1, 1990, pp. 4-32.