

## Research Article

# Research on the Production Scheduling Optimization for Virtual Enterprises

Min Huang,<sup>1</sup> Ruixian Huang,<sup>1</sup> Bo Sun,<sup>2</sup> and Linrong Li<sup>1</sup>

<sup>1</sup> School of Software Engineering, South China University of Technology, Guangzhou 510006, China

<sup>2</sup> Department of Research, Guangdong University of Foreign Studies, Guangzhou 510420, China

Correspondence should be addressed to Min Huang; [minh@scut.edu.cn](mailto:minh@scut.edu.cn)

Received 12 March 2013; Accepted 2 July 2013

Academic Editor: Orwa Jaber Housheya

Copyright © 2013 Min Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Production scheduling is a rather difficult problem in virtual enterprises (VE) for the tasks of production which would be executed by some distributed and independent members. Factors such as the timing constraints of task and ability restrictions of the members are considered comprehensively to solve the global scheduling optimization problem. This paper establishes a partner selection model based on an improved ant colony algorithm at first, then presents a production scheduling framework with two layers as global scheduling and local scheduling for virtual enterprise, and gives a global scheduling mathematical model with the smallest total production time based on it. An improved genetic algorithm is proposed in the model to solve the time complexity of virtual enterprise production scheduling. The presented experimental results validate the optimization of the model and the efficiency of the algorithm.

## 1. Introduction

Virtual enterprise (VE) is a dynamic alliance consisting of independent enterprises. Production scheduling is one of the most important aspects in VE to seize the market opportunities with the shortest production cycle and maximum profits. Hence, production planning has become an important area in VE.

As Helaakoski et al. defined, VE is defined as a temporary collaborative network consisting of independent enterprises, formed to exploit a particular business opportunity [1]. Goel et al. look into both the paradigms and Enterprise Architecture viewpoint of virtual enterprise. Enterprise Architecture deals with the structure of an enterprise, relationships, and interactions of its units [2]. Capuano et al. propose an overview of the Knowledge Virtual Enterprise model, where the virtual enterprise vision is extended with Knowledge-based assets in order to provide an agreement model to support the interoperability among organizations [3]. Danesh et al. propose a 6-layer framework with multiple components within each layer and present a distributed SOA infrastructure that facilitates peer-to-peer collaboration between

enterprises in a virtual enterprise [4]. Esparcia et al. present an extension of the Environment Dimension of the VE model, which is an Organization Modeling Language to define Organization-Centered Multiagent Systems [5].

Partner selection of virtual enterprise is important to VE and production scheduling of VE. Crispim and Sousa mainly talk about the procedure of partner selection in VE and give a model but cannot solve it absolutely [6]. Jarimo and Salo create a mixed-integer linear programming (MILP) model to solve partner selection in virtual organization [7]. In additionally to fixed and variable costs, we present extensions that accommodate transportation costs, capacity risk measures, and interorganizational dependencies such as the success of past collaboration. Nayak et al. propose a variant of swarm optimization to handle combinatorial problems efficiently compared to its continuous counterpart [8]. Simona and Raluca propose partner selection as a difficult task and involve important decision-making because it includes many factors: quality, cost, trust, delivery time, limitations of geography, and communicate abilities [9]. At last, they solve the problem by genetic algorithm. Mohamed focuses on the solution procedure of the multiobjective

partner selection problem in virtual enterprise where the cost coefficients are expressed as interval by the decision maker and uses a multiobjective algorithm (PSA) to solve [10].

Yalaoui et al. solve a hybrid flow shop scheduling problem and create a new method to solve the problem based on the nature which is the particle swarm optimization method under fuzzy logic controller (FLCPSO) [11]. Dugardin et al. create a model to solve multiobjective scheduling of a reentrant hybrid flow shop problem and can be used for partner selection in VE if improved [12]. Gao and Jiang and Ding et al. establish a mathematical model solved using a hybrid genetic algorithm (GA) to acquire the shortest operating cycle based on the characteristics of the production scheduling of VE [13, 14]. However, this model caused premature convergence because that it only used a single population. Song combines the widely applied “cloud computing” theory to present the cloud of VE production planning and control model [15]. However, the model has not yet been tested in practical applications. Gao and Ding, and Li and Liu discuss an order rarely multiple orders, established a multiorder production scheduling model, and introduce various population genetic-simulated annealing algorithms to solve this model [16, 17]. Although the problem of population diversity is solved, the execution time is very slow. Some researchers such as Camarinha-Matos and Afsarmanesh and Zhao and Zhou apply multiagent technology to solve VE production planning and control [18, 19]. They mainly attempt to establish a framework of the production scheduling system. The VE production planning model and the corresponding algorithm need to be further studied. According to the two-tier scheduling model of VE, Tao and Xie establish a mathematical model with the smallest total operating time and used the ant colony algorithm to solve the model [20]. However, premature convergence and slow execution time were the disadvantages of this model.

The structure of the paper is as follows. Section 2 states the problems of partner selection and production planning in VE and creates two models. Section 3 introduces improved ant colony algorithm (IACO) to solve partner selection model and improved multipopulation genetic algorithm (IMGGA) to solve the production scheduling model. Section 4 presents a numerical simulation to show the feasibility of the algorithm. Finally, our solution to the problem and the performance of IMGGA are discussed. Section 5 concludes this paper and ongoing works.

## 2. Description of the Problem and Model

### 2.1. Partner Selection Model

*2.1.1. Three-Stage Model of Partner Selection.* According to the actual characteristics of the virtual enterprise partner selection and the cycle of the selection, the virtual enterprise partner selection can be roughly divided into three stages as the primary selection, fine selection, and optimized combination, as shown in Figure 1.

Primary and fine selection stage is relatively simple; generally the dominated enterprise establishes a specific

assessment team and filters based on certain indicators, excluding the enterprises which do not have the required core resources. Combinatorial optimization phase occupies an important position in the entire virtual enterprise partner selection, and it is related to the success or failure of the partner selection. Therefore, this paper focuses on the establishment of the stages of the model and use of improved ant colony algorithm to solve it.

#### 2.1.2. Partner Selection Combinatorial Optimization Model

*Description of the Problem.* Assume that the dominant enterprise has decomposed the task into  $J$  different subtasks based on the decomposition of business processes. After the primaries of the first phase and the fine selection of the second phase, we can get the combination of the candidate partner enterprises set  $E^{I \times J}$  that need to optimize, where  $J$  represents the task number and  $I$  is the number of candidate enterprises for each task.  $u_{ij}$  is task  $j$  which selects its candidate partners in the corporate collection of the  $i$ th enterprise as the final implementation of the enterprise,  $i = 1, 2, \dots, I, j = 1, 2, \dots, J$ .

Due to the difference of the factor of the characteristics of the market opportunities and core enterprise defects, consideration of core enterprise during the partner selection is also different. Overall, the time, costs and risk are factors to be considered basically in every virtual enterprise [21]. Therefore, this paper considers the impact of the virtual enterprise partner selection factors from three aspects of time, cost, and risk.

*Determination of the Objective Function.* According to the characteristics of the virtual enterprise partner selection, set the time  $T$ , the cost  $C$ , and the risk  $R$  the three objective functions as follows:

$$\min T = \min \left[ \sum_{j=1}^J \sum_{i=1}^I (T_{ij}H_{ij}) \right], \quad (1)$$

$$\min C = \min \left[ \sum_{j=1}^J \sum_{i=1}^I (C_{ij}H_{ij}) \right], \quad (2)$$

$$\min R = \min \left[ \sum_{j=1}^J \sum_{i=1}^I (R_{ij}H_{ij}) \right], \quad (3)$$

$$H_{ij} = \begin{cases} 1, & \text{the candidate partner } u_{ij} \text{ is selected,} \\ 0, & \text{the candidate partner } u_{ij} \text{ is not selected.} \end{cases} \quad (4)$$

Wherein  $T_{ij}$  indicates the required time when the task  $j$  is the  $i$ th candidate corporate executive,  $C_{ij}$  represents required costs when task  $j$  is the  $i$ th candidate corporate executive,  $R_{ij}$  is the risk that chooses the  $i$ th candidate enterprises to execute task  $j$ .  $J$  represents the total number of tasks;  $I$  is the number of candidate enterprises for the  $j$ th task. Ultimate goal is a minimum time, cost, and risk.

For multiobjective decision-making problems, it is difficult for the decision makers to prepare, give the specific means, and consider the polarity inconsistency of goal.

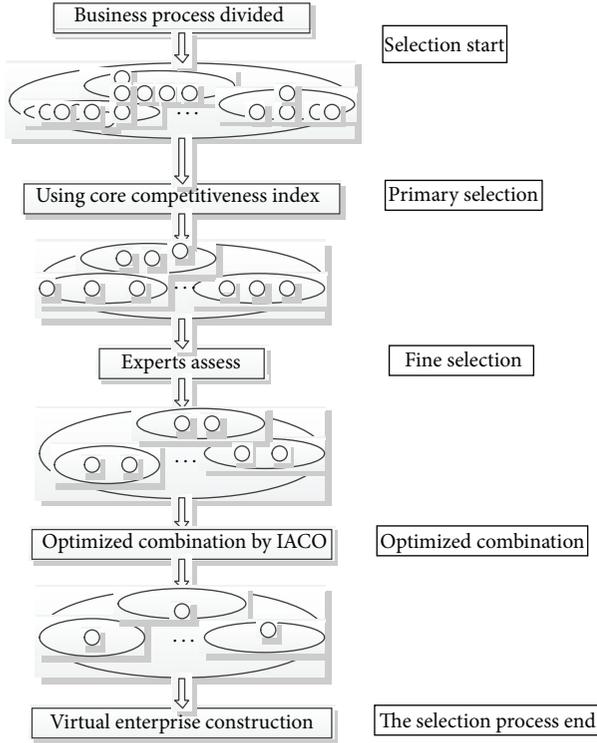


FIGURE 1: Three-stage model of partner selection.

Ideal point method (TOPSIS) is a good method for solving multiobjective decision problems. Guo and Jin applied TOPSIS to solve the problem of multiple index decision grey relation [22]. Ideal point method was first proposed by C. L. Hwang and K. Yoon in 1981 [23], sort ideal point method sort the closeness  $e$  based a finite number of evaluation objects with idealized goal, and the relative merits of existing object evaluation ideal point method is an effective method multiobjective decision analysis, also known as pros and cons of solutions of distance method. Define a measure in space to measure the degree of a program close to the ideal solution and away from negative ideal solution, pros and cons of sort of candidate programs based on the value of this measure to provide a basis for decision-making. This method is based on obtaining a good application.

Use ideal point method, and construct the fitness function of the multiobjective decision problems; decision makers need to give positive ideal value (positive ideal point) of each goal and negative ideal value (negative ideal point). The positive ideal point of the time, cost, and risk in this paper is  $(T^+, C^+, R^+)$ , negative ideal point is  $(T^-, C^-, R^-)$ . This can be constructed out the objective function  $f(t)$ .

$$f(t) = \frac{d^-}{d^- + d^+}, \quad (5)$$

$$d^-(t) = \left\| \left( \frac{T(t) - T^-}{T^+}, \frac{C(t) - C^-}{C^+}, \frac{R(t) - R^-}{R^+} \right) \right\|, \quad (6)$$

$$d^+(t) = \left\| \left( \frac{T(t) - T^+}{T^-}, \frac{C(t) - C^+}{C^-}, \frac{R(t) - R^+}{R^-} \right) \right\|, \quad (7)$$

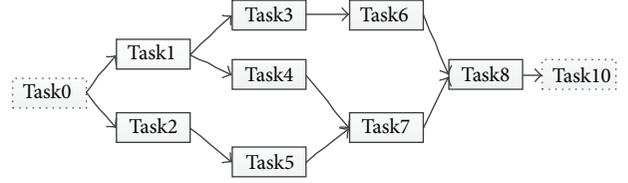


FIGURE 2: The figure of task decomposition.

where  $\| \cdot \|$  means to take norm,  $t$  is the number of ants,  $d^+$  is the positive ideal distance measure, and  $d^-$  point is the negative ideal distance measure. The Euclidean norm can be used to solve it, and the values  $d^+$  and  $d^-$  can be calculated using

$$d^+(t) = \sqrt{(T(t) - T^+)^2 + (C(t) - C^+)^2 + (R(t) - R^+)^2},$$

$$d^-(t) = \sqrt{(T(t) - T^-)^2 + (C(t) - C^-)^2 + (R(t) - R^-)^2}. \quad (8)$$

**2.1.3. Numerical Example.** As shown in Figure 2, it is assumed that the dominated enterprise decomposition market opportunities objectives into Task1 ... Task8. Task0 and Task9 are the starting point and end point.

Through the first phase and the second phase of the selecting, the number of candidate enterprises which can enter into the optimum combination of stage has been narrowed into an appropriate range. The information set of Task1 ... Task8 in optimized combination stage corresponding to candidate companies is shown in Table 1.

According to the information provided in Table 1, calculate each candidate set using the ideal point method. For example, positive ideal points of the candidate set  $\{P_{11}, P_{12}, P_{13}, P_{14}\}$  is (90, 5.5, 0.35), and the negative ideal point is (110, 7.5, 0.45). The calculated ant path hunt to the diagram shown in Figure 3 is carried out for all candidate sets. Calculating the collection of all candidates can get the digraph of ants' path finding.

## 2.2. Production Scheduling Model

**2.2.1. Description of the Problem.** After the dominant enterprise establishes VE, the  $N$  subtasks are assigned to the  $M$  member enterprises. The production plan is developed based on the timing relationships between the tasks and the constraints of the overall scheduling. This model aims at obtaining the maximum profits and deliver customer-specified products with the shortest production cycle [24]. In this paper, we consider that the VE production planning has the following characteristics [25].

- The dominant enterprise specifies the operating time of the decomposition of the task.
- Subtasks in the implementation process without interruption, for author/s of only one affiliation.
- Subtasks with timing constraints.
- When member enterprises are assigned to multiple tasks, considering the member enterprises of their

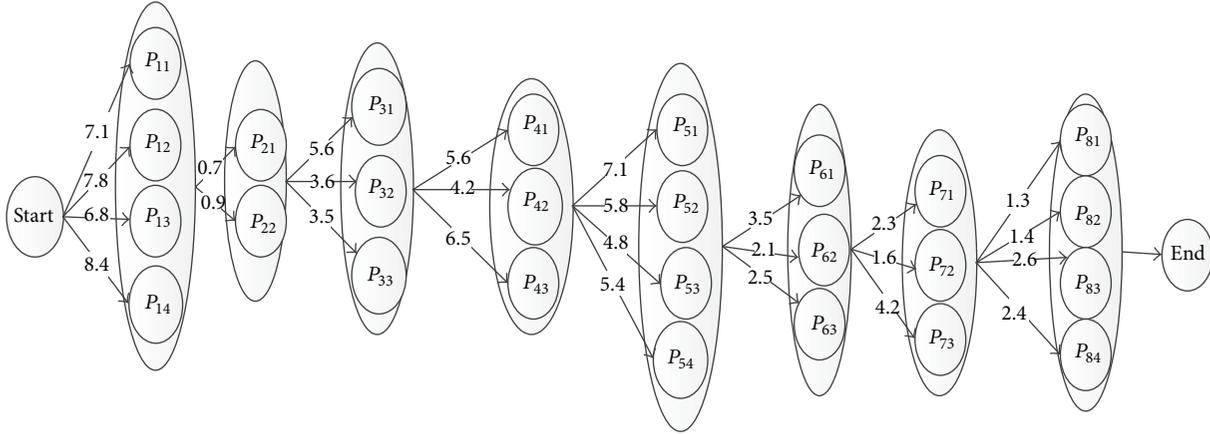


FIGURE 3: Digraph with weight.

TABLE 1: Time, cost, and risk of candidate enterprises.

Task	Candidate enterprise	Cost (10000\$)	Time (month)	Risk
Task1	$P_{11}$	100	6.5	0.41
	$P_{12}$	99	7.1	0.40
	$P_{13}$	110	5.5	0.35
	$P_{14}$	90	7.5	0.45
Task2	$P_{21}$	56	2.7	0.22
	$P_{22}$	77	3.3	0.33
Task3	$P_{31}$	59	5.3	0.50
	$P_{32}$	67	5.5	0.45
	$P_{33}$	50	4.5	0.35
Task4	$P_{41}$	80	5	0.33
	$P_{42}$	88	3.2	0.51
	$P_{43}$	79	3.4	0.52
Task5	$P_{51}$	130	5.2	0.22
	$P_{52}$	139	4	0.33
	$P_{53}$	122	6.2	0.25
	$P_{54}$	140	4	0.20
Task6	$P_{61}$	80	5	0.33
	$P_{62}$	79	3.4	0.52
	$P_{63}$	88	3.2	0.51
Task7	$P_{71}$	56	2.7	0.22
	$P_{72}$	77	3.3	0.22
	$P_{73}$	103	4.1	0.25
Task8	$P_{81}$	59	5.3	0.50
	$P_{82}$	67	5.5	0.45
	$P_{83}$	50	4.5	0.35
	$P_{84}$	66	4	0.25

own resource constraints, a moment can only perform one task.

- (e) Initial time of the production scheduling is 0, and the goal is the shortest operating cycle.

2.2.2. *Symbol Description.* This paper uses the following symbols to describe the mathematical model of VE:

$M$ : the total number of tasks;

$N$ : the total number of enterprises;

$P_i$ : task number,  $i \in [1, N]$ ;

$E_i$ : the number of enterprises,  $i \in [1, M]$ ;

$R_i$ : enterprise  $E_i$  assuming the task set;

$K_i$ : the number of tasks to be undertaken by the enterprise  $E_i$ ;

$st_{ie}$ : the initial production time of  $P_i$  in the enterprise  $E_e$ ;

$ct_{ie}$ : production time of  $P_i$  in the enterprise  $E_e$ ;

$et_{ie}$ : final time of  $P_i$  in the enterprise  $E_e$ ;

$tt_{mn}$ : transport time from enterprise  $E_m$  to  $E_n$ ;

$$x_{ie} = \begin{cases} 1, & \text{task } P_i \text{ in the enterprise } E_i \\ 0, & \text{otherwise;} \end{cases} \quad (9)$$

$$f_{ij} = \begin{cases} 1, & \text{task } P_i \text{ is the preamble of the task of } P_j \\ 0, & \text{otherwise;} \end{cases}$$

$L_x$ : extending task, and  $x \in [1, N]$ ;

$S$ : delaying the task  $L_x$  as a preorder or in connection with tasks performed in the same enterprise;

$S'$ : completed tasks including delayed tasks.

2.2.3. *Model.* Under the constraints, with the initial time of the production scheduling set to 0, the entire production cycle is ensured to be the shortest for each production task in the VE production planning problems by determining the shortest completion time [26]. Delayed by the duration of the task prior to the completion of this task, the subsequent

sequence of tasks and the same enterprise to complete the task to a collection of tasks in  $S$  cannot be executed. The tasks in  $S$  can be reallocated by the dominant enterprise or adjusted dynamically in accordance with the original distribution. VE can be dynamically adjusted according to the sign constraints. The production scheduling model is expressed as follow:

$$\min (T) = \min \left\{ \max_{1 \leq i \leq N} \max_{1 \leq e \leq M} et_{ie} \right\}. \quad (10)$$

The constraints of the initial and final times of a task are shown

$$et_{ie} = st_{ie} + ct_{ie}. \quad (11)$$

The task execution time constraints with direct sequence relationship are expressed in

$$(et_{im} - st_{jn} - tt_{mn}) \cdot f_{ij} \cdot x_{im} \cdot x_{jn} \geq 0. \quad (12)$$

As shown in (13), the same task cannot be performed within different enterprises

$$R_m \cap R_n = \emptyset, \quad \forall m, n \in [1, M]. \quad (13)$$

As shown in (14), all subtasks are allocated to the enterprise

$$R_1 \cup R_2 \cup \dots \cup R_M = P_1 \cup P_2 \cup \dots \cup P_N. \quad (14)$$

The time constraints between tasks before and after in the same enterprise are expressed in

$$(st_{je} - st_{ie}) f_{ij} \geq 0. \quad (15)$$

As shown in (16), the initial time of each task cannot be negative

$$st_{ie} \geq 0, \quad \forall i \in [1, M], \forall e \in [1, M]. \quad (16)$$

As shown in (17), each enterprise can only perform one task at a time

$$\sum_{i=1}^{K_n} x_{ie} = 1, \quad \forall e \in [1, M]. \quad (17)$$

As shown in (18), the finished and unfinished tasks are the sum of all the tasks

$$S \cup S' = P_1 \cup P_2 \cup \dots \cup P_N. \quad (18)$$

### 3. Solving the Models by IACO and IMG A

**3.1. Solving Partner Selection Model by IACO.** Through the above analysis, the use of improved ant colony algorithm (IACO) for virtual enterprise partner selection problem is to use the ants traverse set of candidate companies for each task, each time after only a candidate for a collection of tasks in a candidate enterprises. Improved ant colony algorithms for virtual enterprise partner selection concrete steps are as follows.

**3.1.1. Initialization.** Assume that the core enterprise decomposes the market opportunity goal into  $m$  tasks, and there is a collection of selected partners for each task. The initial number of ants is  $n$ , ants carry the initial pheromone as  $\tau_{ij}(0) = \tau_0$ , ( $\tau_0$  is a constant). The initial size of the pheromone between the enterprise node is obtained by the formula (5). The required value of the reciprocal is called the degree of attraction in this article. For the initial placed node of each ant is set randomly, so the starting point of each ant is not necessarily the same.

**3.1.2. Ants Jump.** Ants exist in discrete states. They use probability transfer rules to move from one node to another node. The path selection probability of ant  $t$  from node  $i$  moving to node  $j$  is

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \varphi_{ik}^\beta(t)}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha \cdot [\varphi_{is}(t)]^\beta}, & \text{if in allowed}_k \\ 0, & \text{else,} \end{cases} \quad (19)$$

where  $\tau_{ij}(t)$  represents the concentration of pheromone of the edge  $(i, j)$  at the  $t$ th search cycle,  $k$  is the attraction of  $j$  point (obtained by the inverse of the value of (5)), parameters  $\alpha, \beta$  represent relative weights, for adjusting the relative importance of  $k$  and  $j$ , and the larger  $\alpha$ , the more ants tend to choose the road section that other ants use, reflecting the collaboration among ants. The greater  $\beta$  represents the more influence to the degree of attraction of jump probabilities, and the probabilities are close to the greedy rule. The  $j$  ( $j \in \text{allowed}_k$ ) denotes the set of ant  $k$  that is allowed to choose the next node, that taboo  $[j] = 1$ . If you simply transfer the maximum probability, it will soon fall into local optimum; the optimal solution cannot be found. Therefore, we choose the roulette algorithm which combines the probability to guide the transfer of ants.

**3.1.3. Update Taboo Table.** For each ant can only access the nodes which haven't been visited before, so a taboo table  $[m][n]$  is set to mark it, in which the value of taboo  $[k][t]$  refers the point accessed by ant  $k$  at the time  $t$ . In this algorithm, one task can only be assigned one partner, so when a node is accessed by ants, all of the other nodes in the collection of the corresponding candidate enterprises would be marked as accessed.

**3.1.4. Update Pheromone.** Ants traverse a candidate enterprise collection to construct a feasible solution, that needs to update the global pheromone. The pheromone update includes two aspects: the pheromone that ants leave and volatile pheromone over time. Ants will leave a certain amount of pheromone on the edge of the path, and the size of the pheromone left by ant  $k$  is calculated by the formula (20) as follows:

$$\Delta \tau_{ij}^k(t) = \frac{Q}{L[k]}, \quad (20)$$

where  $Q$  is a positive constant number, according to the specific circumstances.  $L[k]$  is the length of the path that ant  $k$  traveled through; in this paper, the length is calculated by the degrees of the attraction of each other node size plus 1. At the end of the  $t$ th search cycle, the pheromone on the concentration increment remains in the edge  $(i, j)$  as shown

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t). \quad (21)$$

Pheromone will disappear gradually over time. Assuming that the pheromone retention factor is  $k$ , and the volatiles ratio is  $j$ , which represents the degree of disappearance of the pheromone. So the edge  $(i, j)$  on the global pheromone update is carried out in accordance with

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t), \quad \rho \in (0, 1). \quad (22)$$

After (2), (3), and (4), the ants completed a full traversal; after getting an optimal solution, update the global pheromone. Repeated (2), (3), and (4) operations in accordance with the initial number of ants, and finally choose one path that ant go through most as the optimal solution of the algorithm execution.

Using example of part II, the IAOC is run in MATLAB 7.0; the number of ants in initialization is 30, the constants  $Q = 20$ , and  $C = 0.1$ , and the weight coefficients  $\alpha, \beta$  are 0.7 and 0.9. After all ants finished, 27 ants converge to the path (P14, P22, P31, P43, P51, P61, P73, and P83). So we can select the candidate as the ultimate corporate partners on this path. And the candidate corporate on this path can be selected as the ultimate partners.

In the case of large solution space, the algorithm will not converge and thus cannot get the optimal combination of the optimal solution. We solve this problem through adjust the values of  $\alpha, \beta$ , and ants' initial phenomenon.

### 3.2. Solving Scheduling Model by IMGGA

**3.2.1. Encoding and Decoding.** For simple operation, genetic encoding is based on task numbers. For tasks numbered as  $1 - N$ , genetic sequences are the permutations of the  $N$  numbers. Decoding is the sequence of topological sorting. Therefore, genetic sequences are transformed topologically following the constraint graph of the task.

#### 3.2.2. Genetic Operator

**Population Initialization.** A population size of `pop_size` is randomly generated and then randomly allocated to  $n$  sub-populations. The size of sub-population is `pop_size/n`.

**Improved Multipopulation Genetic Algorithm.** The multipopulation GA improves the performance of GA. This paper designs an IMGGA [27] based on the existing multi-population GA. The entire population consists of a more general population and an excellent population, with a good population of the ordinary population evolution of a certain algebraic cross to guide the evolution of the general population and

improve the quality of the general population. The excellent population continues to absorb the best individual from the general population to improve its quality and hasten the convergence rate of GA.

**Fitness Function.** The production model of this paper is based on the shortest production time for the target. Thus, the fitness function is defined as  $f_i = F_{\max j} - F_{ij}$ , where  $F_{\max j}$  is the maximum execution time of all the genes in the sub-population  $j$  and  $F_{ij}$  is the execution time of gene  $i$  in the sub-population  $j$ .

**Select Operator.** In this paper, the general populations are selected using the roulette wheel selection operator and the elitist strategy to ensure that the highest fitness individuals are copied to the next generation.

**Crossover Operator.** The partially mapped crossover is a good method for increasing the diversity of the offspring through the intersection between the parent individuals. Two cross-points are randomly selected, and the fragments between the cross-points of the parent individuals are exchanged. A conflict occurs if the gene outside the cross-point does not crossover the gene fragment. Otherwise, the genes are determined through gene mapping.

**Mutation Operator.** Mutation in combinatorial optimization problems is commonly used in reverse swap and insert operation. The so-called reversion is reversing two different random position genes in the chromosome [28]. In this paper, the reverse operation is used to ensure that the largest possible chromosomal variation is obtained.

**Simulated Annealing Mechanism.** In simulated annealing algorithm [29], an initial solution is provided. Another solution is randomly generated from the field. The acceptance criteria allows the target function to deteriorate within a finite range, and it is decided by a parameter similar to the temperature control parameter in physical process. Combined with genetic algorithm, the acceptance criteria can prevent premature convergence.

**Termination.** The algorithm terminates when the global optimal solution does not change in a continuous  $K$ -generation and when the per-set maximum number of iterations is completed.

#### 3.2.3. Steps of the Improved Genetic Algorithm

**Step 1.** The GA parameters are initialized. Population size is `pop_size`, the largest breeding algebra is `max_gen`, the number of subpopulations is  $n$ , the cross rate and variation rate of sub-population are  $F_{ic}$  and  $F_{im}$ , respectively, the global optimal solution unchanged termination of algebraic  $K$ , the subpopulation independent evolution of the  $Z$  algebra, global evolution of algebra `loop1 = 1`, general sub-population evolutionary algebra `loop2 = 1`, the initial annealing temperature is  $T$ , the minimum temperature is  $t$ , and the annealing coefficients are  $c$ .

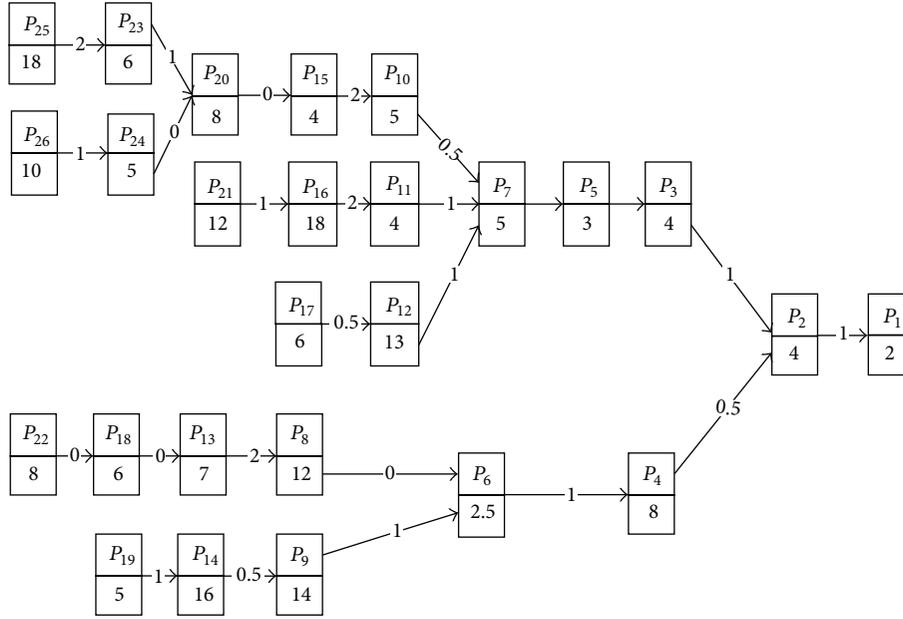


FIGURE 4: Timing constraints of tasks.

Step 2. The following operations are implemented for each general sub-population  $i$ .

- (a) The sub-populations in the fitness of each individual are evaluated.
- (b)  $x_{ij}$ ,  $x_{ik}$  are randomly selected from the sub-population. Two new individuals  $x'_{ij}$ ,  $x'_{ik}$  are crossed over and generated, and their fitness values are  $f(x'_{ij})$  and  $f(x'_{ik})$ . If  $f(x'_{ij}) \leq f(x_{ij})$ , accept  $x_{ij}$ ; if  $\min \{1, \exp(-(-f(x'_{ij}) - f(x_{ij}))/T)\} > \text{random}$  and  $f(x'_{ij}) > f(x_{ij})$ , accept  $x'_{ij}$ . The acceptance of  $x_{ik}$  and  $x'_{ik}$  are the same as before.

Step 3. Mutation. Whether the new individual is accepted according to Step 2 is determined.

Step 4. If  $\text{loop1} < Z$ ,  $\text{loop1} = \text{loop} + 1$ , Step 2 is repeated; otherwise, Step 5 is followed.

Step 5. The best individual of each of the general population (20%) is used to establish an excellent population. Crossover and mutation according to Steps 2 and 3 in the excellent population are performed.

Step 6. Form a mixed population by several general populations, and replace the individuals of the mixed population by of mixed population. Excellent populations before 20% individual replacement of the mixed population are then divided into new general populations.

Step 7. If the optimal solution is the same algebra  $K$ , this solution is obtained, and the algorithm terminates; otherwise, Step 8 is followed.

Step 8. If  $\text{loop2} < \text{max\_gen}$ ,  $T = cT$ ,  $c \in (0, 1)$ , and  $\text{loop2} = \text{loop2} + 1$ , Step 2 is repeated; otherwise, the algorithm is terminated.

## 4. Experiment

4.1. *Examples of Production Scheduling.* Enterprise  $E_1$  could not independently complete a device manufacturing task. It divides the task into 26 subtasks ( $P_1 \dots P_{26}$ ). Finally,  $E_1$  establishes a VE with four partners ( $E_2, E_3, E_4$ , and  $E_5$ ) through a tender:  $E_1$  implementation of the task set  $\{P_1, P_5, P_7, P_{14}, P_{23}, P_{26}\}$ ,  $E_2$  implementation of the task set  $\{P_4, P_9, P_{15}, P_{20}, P_{24}\}$ ,  $E_3$  implementation of the task set  $\{P_2, P_{10}, P_{12}, P_{16}, P_{19}\}$ ,  $E_4$  implementation of the task set  $\{P_6, P_8, P_{11}, P_{17}, P_{25}\}$ , and  $E_5$  implementation of the task set  $\{P_3, P_{13}, P_{18}, P_{21}, P_{22}\}$ . The task execution order constraints are shown in Figure 4. The top of the rectangular box represents the task number, the bottom represents the mission time, and the arrow line numbers indicate the transit time (when the production tasks are completed by the same enterprise, the transit time is 0).

4.2. *Scheduling Optimization.* All parameters are initialized, and MATLAB programming is used to achieve the improved multipopulation GA. The initial population size is 100, which is then divided into three general populations (with sizes of 30, 30, and 40). The crossover and mutation probabilities of the three populations are 0.8, 0.1; 0.85, 0.15; and 0.9, 0.2, respectively. The length of the chromosome is 26. The sub-population evolution algebra is 10, and the maximum evolution generation of the GA is 100. The initial annealing temperature, annealing coefficient, and lowest temperature are  $100^\circ\text{C}$ , 0.97, and  $10^\circ\text{C}$ , respectively. After implementation

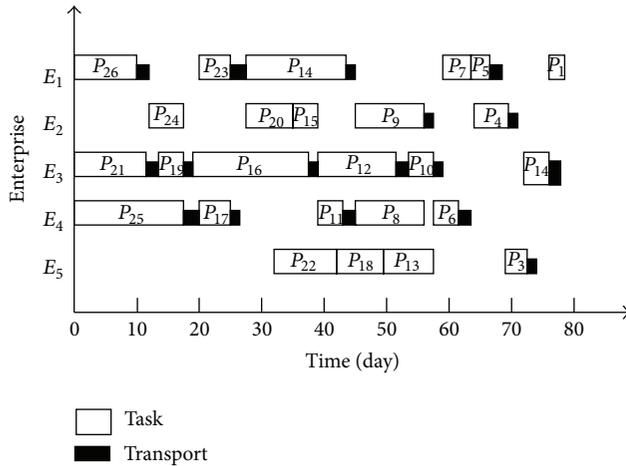


FIGURE 5: Optimal plan program by IMGSA.

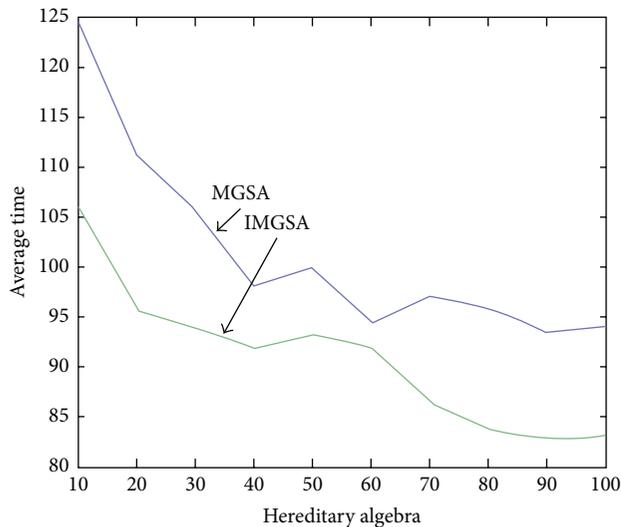


FIGURE 6: The average time comparison chart of MSGA and IMGAS.

of the algorithm, the optimal manufacturing cycle is 77.5 days. The optimal scheme using the Gantt chart is shown in Figure 5.

Using the above algorithm parameters, the unimproved multi-population (UMGSA) and improved multi-population algorithm (IMGSA) are run for 100 generations. The average completion time of the task with the convergence of genetic algebraic diagram is shown in Figure 6. The improved GA convergence rate is fast.

## 5. Conclusion

This paper presents a model for solving production planning. The improved GA is used to solve the model. It provides a good method for solving the problem of the production planning of VE. The convergence rate of the improved GA is faster than that of the unimproved GA. Thus, an optimal production plan is obtained quickly to guide the production of VE. During the production operation, the production plan

may change any time because companies are subjected to certain external factors. Therefore, the unfinished task of concentration can be based on the information of the completed task, and this algorithm can be used for adjustments.

## Acknowledgments

This work was supported by some grants from Guangdong Province Production Education and Scientific Study Programs, China (no. 2012B091100490 and no. 2011B090400056) and the Fundamental Research Funds for the Central Universities, SCUT.

## References

- [1] H. Helaakoski, P. Iskanius, and I. Peltomaa, "Agent-based architecture for virtual enterprises to support agility," *IFIP International Federation for Information Processing*, vol. 243, pp. 299–306, 2007.
- [2] A. Goel, H. Schmidt, and D. Gilbert, "Towards formalizing virtual enterprise architecture," in *Proceedings of the 13th Enterprise Distributed Object Computing Conference Workshops (EDOCW '09)*, pp. 238–242, September 2009.
- [3] N. Capuano, S. Miranda, F. Orciuoli, and S. Vassallo, "E-Learning at work in the knowledge virtual enterprise," in *Proceedings of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '08)*, pp. 507–512, March 2008.
- [4] M. H. Danesh, B. Raahemi, and M. A. Kamali, "A framework for process management in service oriented virtual organizations," in *Proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP '11)*, pp. 12–17, October 2011.
- [5] S. Esparcia, R. Centeno, R. Hermóso, and E. Argente, "Artifancing and regulating the environment of a virtual organization," in *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI '11)*, pp. 547–554, November 2011.
- [6] J. A. Crispim and J. P. De Sousa, "Partner selection in virtual enterprises," *International Journal of Production Research*, vol. 48, no. 3, pp. 683–707, 2010.
- [7] T. Jarimo and A. Salo, "Multicriteria partner selection in virtual organizations with transportation costs and other network interdependencies," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 39, no. 1, pp. 124–129, 2009.
- [8] N. Nayak, K. Prasanna, S. Datta, S. S. Mahapatra, and S. Sahu, "A novel swarm optimization technique for partner selection in virtual enterprise," in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM '10)*, pp. 1118–1122, December 2010.
- [9] D. Simona and P. Raluca, "Intellegent modeling method based on genetic algorithm for partner selection in virtual organizations," *Business and Economic Horizons*, vol. 5, no. 2, pp. 23–24, 2011.
- [10] A. M. Mohamed, "Optimal composition of virtual enterprises with interval cost parameters," in *Proceedings of the 8th International Conference on Informatics and Systems (INFOS '12)*, pp. BIO-188–BIO-194, 2012.
- [11] N. Yalaoui, L. Amodeo, F. Yalaoui, and H. Mahdi, "Particle swarm optimization under fuzzy logic controller for solving a hybrid reentrant flow shop problem," in *Proceedings of the IEEE*

- International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW '10)*, pp. 1–6, April 2010.
- [12] F. Dugardin, L. Amodeo, and F. Yalaoui, “Multiobjective scheduling of a reentrant hybrid flowshop,” in *Proceedings of the 39th International Conference on Computers and Industrial Engineering (CIE '09)*, pp. 193–198, July 2009.
- [13] Y. Gao and Z. B. Jiang, “Hybrid genetic algorithm for virtual production planning,” *Control and Decision-Making*, vol. 22, no. 8, 2007.
- [14] Y. Ding, Y. Gao, and G. Luo, “Virtual enterprise global production planning based on cost optimization with time constraints,” in *Proceedings of the International Conference on Management and Service Science (MASS '09)*, September 2009.
- [15] Q. J. Song, *The Cloud of Virtual Enterprise Production Planning and Control*, Silicon Valley, 2008.
- [16] Y. Gao and Y. S. Ding, *Virtual Enterprise Production Planning Model*, vol. 22, Enterprise Management, 2009.
- [17] Q. S. Li and J. G. Liu, “Research on VE production planning based on multi-agent,” *Research and Design*, vol. 22, no. 1, 2008.
- [18] L. M. Camarinha-Matos and H. Afsarmanesh, *VE Modeling and Support Infrastructures: Applying Multi-Agent System Approaches*, Springer, New York, NY, USA, 2001.
- [19] Q. Zhao and M. Zhou, “Virtual production tasks based on ant colony optimization scheduling,” *Wuhan University of Technology*, vol. 34, no. 3, 2011.
- [20] Z. Tao and L. Y. Xie, “Based on hybrid genetic algorithm for job shop scheduling problem,” *Computer Engineering and Applications*, vol. 18, 2005.
- [21] M. Huang, X. Wang, F.-Q. Lu, and H. Bi, “A coordination of risk management for supply chains organized as virtual enterprises,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 931690, 11 pages, 2013.
- [22] X. Y. Guo and L. Jin, “The gray relational MCDM TOPSIS,” *Technology and Management*, vol. 12, no. 5, 2010.
- [23] J. Singh and A. Madhukar, “New method for calculating non-ideal point defect induced electronic structure,” *Solid State Communications*, vol. 41, no. 12, pp. 947–950, 1982.
- [24] W. Zhou and Y. Bu, “Cultural algorithm based on particle swarm optimization for partner selection of virtual enterprise,” in *Proceedings of the 31st Chinese Control Conference*, 2012.
- [25] S.-L. Yang, H.-W. Kang, and H. Zhou, “Research on web service-based virtual enterprise integration framework,” in *Proceedings of the 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization*, 2012.
- [26] M. A. O. Pessoa, F. Junqueira, and D. J. S. Filho, “Virtual enterprise planning system using time windows and capacity constraint concepts,” in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON '12)*, pp. 2851–22856, 2012.
- [27] C. F. M. Toledo and J. M. G. Lima, “A multi-population genetic algorithm approach for PID controller auto-tuning,” in *Proceedings of the IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA '12)*, pp. 1–8, 2012.
- [28] B. Xiaojun and L. Guangxin, “The improvement of ant colony algorithm based on the inver-over operator,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07)*, pp. 2383–2387, August 2007.
- [29] Q. Xu, J. Mao, and Z. Jin, “Simulated annealing-based ant colony algorithm for tugboat scheduling optimization,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 246978, 22 pages, 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

