

## Research Article

# An Improved Hierarchical Genetic Algorithm for Sheet Cutting Scheduling with Process Constraints

Yunqing Rao,<sup>1</sup> Dezhong Qi,<sup>1</sup> and Jinling Li<sup>2</sup>

<sup>1</sup>The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science & Technology, Wuhan, Hubei 430074, China

<sup>2</sup>Shenyang Dongfon Titanium Industry Co., Ltd, Shenyang, Liaoning 110168, China

Correspondence should be addressed to Dezhong Qi; [derek@hust.edu.cn](mailto:derek@hust.edu.cn)

Received 24 September 2013; Accepted 18 November 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Copyright © 2013 Yunqing Rao 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.

For the first time, an improved hierarchical genetic algorithm for sheet cutting problem which involves  $n$  cutting patterns for  $m$  non-identical parallel machines with process constraints has been proposed in the integrated cutting stock model. The objective of the cutting scheduling problem is minimizing the weighted completed time. A mathematical model for this problem is presented, an improved hierarchical genetic algorithm (ant colony—hierarchical genetic algorithm) is developed for better solution, and a hierarchical coding method is used based on the characteristics of the problem. Furthermore, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm. The computational result and comparison prove that the presented approach is quite effective for the considered problem.

## 1. Introduction

The sheet cutting is widely used in engineering machinery, mining machinery, port machinery, and other industry machinery. The manufacturing model and management of modern enterprises have been changed greatly by the development of science and technology. Since integrated cutting stock has become a new cutting model, the integrated cutting stock brings much economic benefits to the enterprise, but it also brings some difficulty for solving the nesting and cutting scheduling problem in the meantime. Therefore, finding an advanced approach to solving the cutting scheduling problem for integrated cutting stock has an important practical and theoretical significance.

The processing capacity of different types of cutting machines is different. For example, 0.15 mm~6 mm thick steel plate can be cut on the laser cutting machine, and the cutting speed can reach 10000 mm/min. Meanwhile, 5 mm~200 mm thick steel plate can be cut on the flame cutting machine, and its cutting speed is 0~700 mm/min. Therefore, the cutting patterns of different material and thickness can be cut on the

different types of cutting machines. In other words, there are some process constraints in the process of integrated cutting stock. Furthermore, the same cutting pattern can be cut on the different types of cutting machines with different cutting speed.

Assignment of  $n$  cutting patterns to  $m$  unit different cutting machines is considered as an integrated cutting scheduling problem. Each cutting pattern can be only cut one time on a machine. The objective of integrated cutting scheduling is minimizing the weighted completed time. It belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups.

Lots of algorithms and approaches are used in cutting stock problem including conventional optimization algorithms and various metaheuristic algorithms. For instance, Blazewicz et al. [1], Gomes and Oliveira [2], Umetani et al. [3], Bennell et al. [4], Cui et al. [5], Cui and Chen [6], Xie et al. [7], and others proposed some nesting algorithms and gave some suggestions on the two-dimensional cutting stock problem. However, literatures on the cutting scheduling problem

(CSP) are not vast. There are some papers concerning the combined cutting stock and lot-sizing problem. NonÅs and Thorstenson [8] wanted to find an optimal production schedule involves the solution of a combined two-dimensional irregular cutting-stock and lot-sizing problem. In that paper, they proposed a problem formulation and suggested different solution algorithms (e.g., local search algorithms and simple tree-search algorithm) for a combined cutting-stock and lot-sizing problem. Gramani and França [9] analyzed the trade-off that arises when they solve the cutting stock problem by taking into account the production planning for various periods. In that paper, they formulated a mixed-integer mathematical model that combines the cutting stock and lot-sizing problems and developed a solution method based on an analogy with the network shortest path problem. NonÅs and Thorstenson [8] suggested a new column generating solution procedure (CGSP) for a combined cutting-stock and lot-sizing problem. Those papers are mainly concerned how to generate the cutting patterns, and minimize the trim-lost or trim-lost, setup cost and holding cost. However, how to assign those cutting patterns to different cutting machines, which belongs to cutting scheduling problem, is not concerned. As a good cutting scheduling can reduce production costs and raise the production efficiency, this problem is also important in the whole sheet cutting process. In this paper, our goal is to find a suitable cutting schedule and minimize the weighted completed time.

The cutting scheduling problem belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups. Scheduling problem has also been studied extensively, for example, Allahverdi et al. classified scheduling problems into batch and non-batch, sequence-independent and sequence-dependent setup, and categorizes the literature according to the shop environments of single machine, parallel machines, flowshops, and job shops [10]. Each scheduling problem is denoted by the standard threefield notation  $\alpha/\beta/\gamma$ . The first field  $\alpha$  describes the scheduling type, the second field  $\beta$  is reserved for the information and conditions of scheduling, while the third field  $\gamma$  contains the performance criteria. Cheng and Sin surveyed the major research results in deterministic parallel-machine scheduling [11]. And lots of the literatures of scheduling have considered parallel-machine scheduling problems. Peng and Liu, Anglani et al., Fowler et al., and other scholars have done some works on parallel machines problem [12–14]. And the vast majority of these studies have concentrated on studying the case of identical parallel machines. However, the case of nonidentical parallel-machine schedules has more practical sense than the case of identical parallel machines in real production (e.g. cutting and scheduling problem belong to nonidentical parallel-machine schedule).

Literatures on nonidentical parallel-machine schedules problem are not vast. Li and Yang gave a review of the nonidentical parallel-machine total weighted/weighted completion time problems [15]. Van Hop and Nagarur proposed a new approach to solve the PCB scheduling problem on a set of nonidentical machines. This approach model which related tasks of grouping, sequencing, and component switching as

one integrated problem, with an objective of minimizing the total makespan [16]. Balin proposed a GA approach, which minimized maximum completion time (makespan) and considered nonidentical parallel machine scheduling problem with fuzzy processing times [17]. In order to adapt GA to nonidentical parallel machine scheduling problem, he proposed a new crossover operator and a new optimality criterion. Alcan and Balişgil presented a kind of genetic algorithm based on machine code for minimizing the processing times in nonidentical machine scheduling problem [18]. Also triangular fuzzy processing times were used in order to adapt the GA to nonidentical parallel machine scheduling problem in that paper. Besides the above mentioned papers, there are a few others that have investigated the unrelated-parallel machine scheduling problems with different approach (Liaw et al. [19], Rocha et al. [20], Mehravaran and Logendran [21], Arnaout et al. [22], Charalambous and Fleszar [23]). However, even though there are some papers considered nonidentical parallel-machine scheduling, few of them deal with nonidentical parallel machine with process constraints.

Taken together, how to assign cutting patterns to different cutting machines (i.e., the sheet cutting scheduling problem) is a key point for the whole sheet cutting process, but this issue are not involved in the previous papers. Considering the practical requirement, some further work on the sheet cutting problem needs to be done. To solving this problem, a mathematical model which takes into account the weighted completed time is presented. Furthermore, an improved hierarchical genetic algorithm (ant colony—hierarchical genetic algorithm) is developed to solve this mathematical model. Based on the characteristics of the problem, a hierarchical coding method is used in this algorithm. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm.

The rest of the paper is organized as follows. In Section 2, a detailed description of the sheet cutting scheduling problem is illustrated. A mathematic model for the sheet cutting problem is presented in Section 3. In Section 4, a process for solving the sheet cutting problem with a hierarchical genetic algorithm is demonstrated. Then the results of computational experiment are described in Section 5. Finally, conclusions and directions are given in Section 6.

## 2. Statement of the Problem

The process of sheet cutting is as follows. First, a variety of cutting patterns with the combination of different parts are generated by special nesting software. Second, those cutting patterns are assigned to different cutting machines. Thirdly, those cutting patterns are cut by different cutting machines. The process of sheet cutting can be showed by Figure 1. As the processing capacity of different types of cutting machines is different, a cutting pattern can be cut on some special cutting machines under different cutting speed. The capacity of some different cutting machines is shown in Table 1.

For example, a cutting pattern, material Q235 and thickness 20 mm, can be cut on plasma cutting machine B, flame

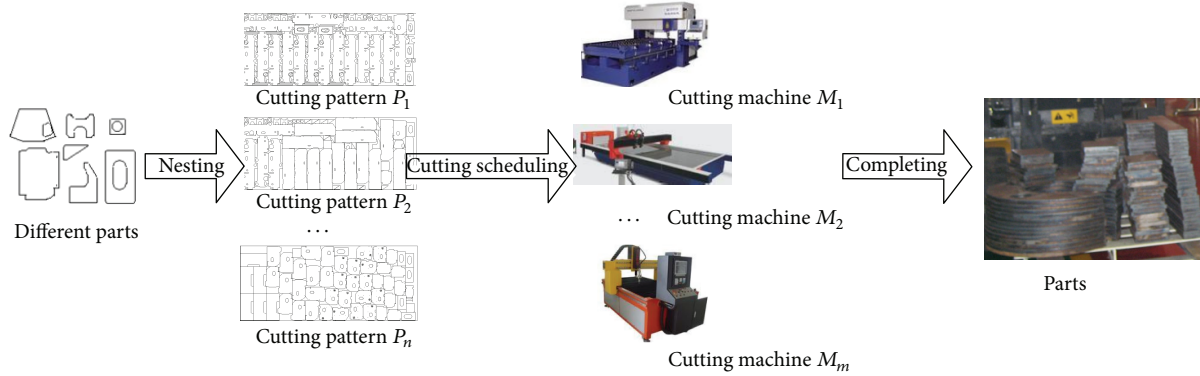


FIGURE 1: The production process of sheet cutting.

TABLE 1: The capacity of different cutting machines.

Machine no.	Cutting machine type	The range of sheet thickness can be processed	Cutting speed
$M_1$	Laser cutting machine	0.15 mm~6 mm carbon steel	10000 mm/min
$M_2$	Plasma cutting machine A	1 mm~15 mm carbon steel	1000 mm/min
$M_3$	Plasma cutting machine B	1 mm~25 mm carbon steel	800 mm/min
$M_4$	Flame cutting machine A	6 mm~200 mm carbon steel	350 mm/min
$M_5$	Flame cutting machine B	6 mm~200 mm carbon steel	450 mm/min
$M_6$	Flame cutting machine C	6 mm~200 mm carbon steel	500 mm/min

cutting machine A, flame cutting machine B and flame cutting machine C. The cutting speed is 800 mm/min on the plasma cutting machine B, but it is 500 mm/min on the flame cutting machine C. Therefore, how to assign cutting patterns to different cutting machines is needed to be solved by some method.

### 3. Mathematical Modeling

We consider the cutting scheduling problem of  $n$  cutting patterns for  $m$  nonidentical parallel machines with process constraints in detail, where  $m < n$ .  $Tp_{ij}$  represents the processing time of cutting patterns, where  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ . It means the processing time of cutting pattern  $P_i$  on the cutting machine  $M_j$ . Each cutting pattern has a corresponding weight coefficient  $\omega_i$ . We need to make an optimal scheduling to minimize the weighted completion time.

The problem considered can be summarized by the following points.

- (1)  $n$  different cutting patterns  $P_1, P_2, \dots, P_n$  need to be cut.
- (2) Each cutting pattern includes some information such as number of parts, cutting length, number of punch, and so forth.
- (3)  $m$  different cutting machines  $M_1, M_2, \dots, M_m$  can be used. Each cutting machine can cut a variety of sheets of different material and thickness.

- (4) The process time of a cutting pattern on different cutting machine may be different.
- (5) The products pass rate is 100%, in other words, there is no reprocessing.
- (6) Once starting cutting, it is not allowed to interrupt.
- (7) Each cutting pattern is independent.
- (8) Only one working procedure (i.e., cutting procedure) is included in this sheet cutting process.
- (9) The cutting process includes special process constraint, which the cutting machine set  $CP_i$  means cutting patterns  $P_i$  can be used,  $CP_i \in M$ ,  $CP_i \neq \Phi$ .
- (10) Setup time includes punch time, collection time (the time of collect parts after completing cut) and the time using to adjust steel on the cutting machine.
- (11) Empty travel time is ignored, in other words, the processing time is the actual cutting time.

*Definition.* Consider the following:

- $n$  is the number of cutting patterns,
- $m$  is the number of cutting machines,
- $i$  is cutting pattern index,  $i = 1, \dots, n$ ,
- $j$  is cutting machine index,  $j = 1, \dots, m$ ,
- $pn_i$  is number of parts in cutting pattern  $P_i$ ,
- $ph_i$  is number of punch in cutting pattern  $P_i$ ,

$$tp_{ij} \begin{cases} \text{cutting time of cutting pattern } i \text{ on machine } j \\ \text{if cutting pattern } P_i \text{ can be} \\ \text{processed on machine } j \\ \infty \text{ if cutting pattern } P_i \text{ cannot be} \\ \text{processed on machine } j, \end{cases} \quad (1)$$

$Vpm_{ij}$  is cutting speed of cutting pattern  $P_i$  on cutting machine  $M_j$ ,

$stp_{ij}$  is setup time of cutting pattern  $P_i$  on cutting machine  $M_j$ ,

$\omega_i$  is process weight of cutting pattern  $P_i$ ,

$c_{ij}$  is completed time of cutting pattern  $P_i$  on machine  $M_j$ .

**Decision Variables.** Consider the following:

$$z_{ij} \begin{cases} 1 & \text{if cutting pattern } P_i \\ & \text{can be cut on cutting machine } M_j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

**Objective Function.** Consider the following:

$$f = \min \sum_{i=1}^n \sum_{j \in CP_i} \omega_i c_{ij} z_{ij} \quad (3)$$

subject to

$$\sum_{j \in CP_i} z_{ij} = 1, \quad i = 1, 2, \dots, n \quad (4)$$

$$\sum_{i=1}^n \sum_{j \in CP_i} z_{ij} = n, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (5)$$

$$c_{ij} = (c_{xj} + tp_{ij} + stp_{ij}) \times z_{ij}, \quad (6)$$

$$stp_{ij} = ph_i \times 1 + pn_i \times 0.5 + 5, \quad (7)$$

$$tp_{ik} \times Vmp_{ik} = t_{il} \times Vmp_{il} \quad k, l = 1, 2, \dots, m, \quad (8)$$

$$CP_i \in M, \quad CP_i \notin \Phi. \quad (9)$$

The minimizing weighted completed time can be got by objective function (3). Equation (4) considers a cutting pattern can be cut only on one cutting machine. All of cutting patterns are sure to be cut by (5). Equation (6) represents the completed time of cutting pattern  $P_i$  on machine  $M_j$ . It consists of three components which involve  $c_{xj}$ ,  $t_{pj}$  and  $stp_{ij}$ , where  $c_{xj}$  represents the completed time of cutting pattern  $P_x$

which is cut on the cutting machine  $M_j$  before cutting pattern  $P_i$ . There are three components for (7). The first component represents punch time (punch time of each drilling is 1 minute). The second component represents collection time (the time of collect each part which a cutting pattern includes cut is 0.5 minute). The last component relates to the time using to adjust steel on the cutting machine. Processing time of a cutting pattern  $P_i$  on different machines can be expressed by (8). Formula (9) considers the cutting process contains a special process constraint. In other words, a cutting pattern is assigned which cutting machine depends on the sheet characteristics.

With the increase of the problem size, the solution of the problem becomes very complicated, or it is even impossible to be solved with conventional optimization methods. And it has been proved it is a NP-problem. In this paper, an ant colony—hierarchical genetic algorithm is considered to solve this problem. A set of optimized initial solution is generated by ant colony algorithm which has some merits, such as simple and universal, robustness. However, it is easy to fall into local optimum. Then, the hierarchical genetic algorithm which has strongly global search capability is used to further optimize the initial solution. Fast and efficient global optimizing can come true by using the ant colony—hierarchical genetic algorithm.

#### 4. Ant Colony—Hierarchical Genetic Algorithm

A set of optimized initial solution is generated by using the ant colony algorithm for its simple and universal [24]. Then the solution is furthermore optimized by hierarchical genetic algorithm for its strongly global search capability. A hierarchical structure, which is consisted of parameter genes and control genes, is used by the hierarchical genetic algorithm [25]. The parameter genes are the lowest level, and control genes are in the higher levels of the parameter genes. The parameter genes are controlled by the control genes. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm.

**4.1. Code.** If the binary encoding is used for solving the cutting scheduling problem, the chromosome will be very complex and difficult for the crossover operator and decoding. So, to overcome these difficulties, the natural number encoding couple with binary encoding is used.

For example, six cutting patterns are assigned to three different cutting machines. The cutting sequence is showed by control genes and the cutting machine of select is showed by parameter genes. The value of parameter gene is 1 means the selected cutting machine is  $M_1$ . As is shown in Figure 2, the sequence of cutting patterns is  $P_4, P_5, P_1, P_6, P_2$ , and  $P_3$  and the cutting pattern  $P_1, P_2, P_3, P_4, P_5$ , and  $P_6$  are, respectively, assigned to  $M_1, M_2, M_1, M_1, M_3$ , and  $M_2$ . Each parameter gene can be further expressed by 0-1 variables  $z_{ij}$ .

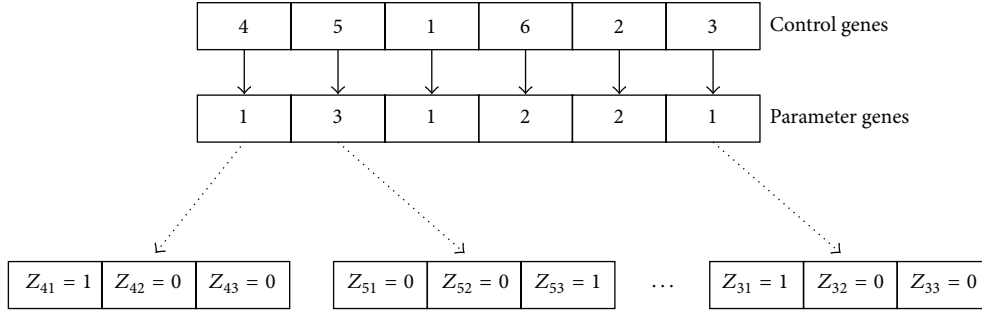


FIGURE 2: The process of natural number encoding.

**4.2. Initial Population.** The initial population is generated by ant colony algorithm. The following gives the concrete solving process of ant colony algorithm.

**Selecting the Ants' Path.** It supposes that the ants movement between nodes which represent cutting patterns on different cutting machines and different amount of pheromone left on the nodes at the same time. Then the pheromone impacts the path of the next lot size ants moving [26].  $\tau_{z_{ij}}(t)$  represents the pheromone values at  $t$  ( $t = 0, 1, 2, \dots$ ) moment on the different nodes. The initial of the pheromone value  $\tau_{z_{ij}}(0) = \varepsilon$  (the  $\varepsilon$  is a minimal number). There are  $N_{\text{ant}}$  ants distributed on the different nodes. Then each ant according to the pheromone value of next node and the heuristic factor independently chooses the next node.  $p_{z_{ij}z_{(i+1)x}}^k(t)$  represents that the transition probability of ant  $k$  ( $k = 1, 2, \dots, N_{\text{ant}}$ ) moves from the node  $z_{ij}$  to the next node  $z_{(i+1)x}$  at time  $t$ . The formulation of  $p_{z_{ij}z_{(i+1)x}}^k(t)$  can be described as follows:

$$p_{z_{ij}z_{(i+1)x}}^k(t) = \begin{cases} \frac{[\tau_{[z_{ij}][z_{(i+1)x}]}(t)]^\alpha [\eta_{[z_{ij}][z_{(i+1)x}]}(t)]^\beta}{\sum_{x, x \in \text{CP}_{(i+1)}} [\tau_{[z_{ij}][z_{(i+1)x}]}(t)]^\alpha [\eta_{[z_{ij}][z_{(i+1)x}]}(t)]^\beta} & z_{(i+1)x} \notin \text{tabu}_k, x \in \text{CP}_{(i+1)} \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

where  $\text{tabu}_k$  represents the next node set which ant  $k$  cannot go to.  $\alpha$  is a positive parameter, whose value represents the relative influence of pheromone trail. It shows the relative importance of ant track. The bigger the value of  $\alpha$  is, the ant more inclines to choose the path which others have passed.  $\beta$  is a positive parameter, whose value represents the relative influence of heuristic information. The value of  $\beta$  is bigger, the state transition probability will be close to the greed rules.  $\eta_{[z_{ij}][z_{(i+1)x}]}(t)$  is a heuristic function, whose value represents the expected next node ( $z_{(i+1)x}$ ) what the ant expects to choose. The value of  $\eta_{[z_{ij}][z_{(i+1)x}]}(t)$  can be gained by (11), where  $c_{(i+1)x}$  represents the completion time of next node:

$$\eta_{[z_{ij}][z_{(i+1)x}]}(t) = 1 - \frac{c_{(i+1)x}}{\sum_{x, x \in \text{CP}_{(i+1)}} c_{(i+1)x}}. \quad (11)$$

**Pheromone Updating.** Pheromone evaporation is inevitable; meanwhile, the ants deposit pheromone in each iteration. So the pheromone value  $\tau_{z_{ij}}(t)$  is changing in each iteration. The pheromone value  $\tau_{z_{ij}}(t)$  can be updated as following equations:

$$\tau_{[z_{ij}][z_{(i+1)x}]}(t+1) = (1 - \rho) \tau_{[z_{ij}][z_{(i+1)x}]}(t) + \Delta \tau_{[z_{ij}][z_{(i+1)x}]}(t) \quad (12)$$

$$\Delta \tau_{[z_{ij}][z_{(i+1)x}]}(t) = \sum_{k=1}^{N_{\text{ant}}} \Delta \tau_{[z_{ij}][z_{(i+1)x}]}^k, \quad (13)$$

$$\Delta \tau_{[z_{ij}][z_{(i+1)x}]}^k = \begin{cases} \frac{Q}{Z_k} & \text{if ant } k \text{ travels on edge } (z_{ij}, z_{(i+1)x}) \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where  $\rho$  ( $0 < \rho < 1$ ) is the rate of pheromone evaporation, and  $1 - \rho$  is the rate of pheromone retention. Reducing the pheromone values enables the algorithm to forget bad decisions made in previous iterations [27]. Thus, the pheromone updating can help ants to explore new area in the search space.  $\Delta \tau_{[z_{ij}][z_{(i+1)x}]}^k$  represents the pheromone value deposited on edge ( $z_{ij}, z_{(i+1)x}$ ) by ant  $k$  at  $t$  iteration;  $\Delta \tau_{[z_{ij}][z_{(i+1)x}]}$  is the sum of the pheromone value deposited on edge ( $z_{ij}, z_{(i+1)x}$ ) by all of ants;  $Q$  represents pheromone strength that affects the convergence speed of the algorithm.  $Z_k$  represents the objective function value of ant  $k$  in this iteration.

The step by step details are given as shown in Algorithm 1.

**4.3. Fit Function.** Minimizing weighted completion time is objective function, and the fitness function can be obtained by the exponential transform of objective function:

$$f = a \exp\left(-b \sum_{i=1}^n w_{ij} c_{ij} z_{ij}\right), \quad (15)$$



```

Step1. Initialization
    Set ACO parameters  $\alpha, \beta, \rho, \tau_{y_{jm}}(0), NC_{max}$  (Maximum iterations),  $Q, N_{ant}$  (number of ant);
Step2. Generate  $N_{ant}$  ants, and place it on different nodes
Step3. for  $NC = 1:NC_{max}$ 
    for  $k = 1:N_{ant}$ 
        for  $i = 2:n$ 
            Cumulating the transition probability of each ant will go to according (11), ants select the next node of max
            transition probability
        End;
    End;
Step4. Cumulating the best solution for the objective function (3) in this iteration. If the current solution is better than the
    former best solution, then update the former best solution.
Step5. updating the pheromone value according to (12), (13) and (14).
Step6. if  $NC < NC_{max}$ 
     $NC = NC + 1$ ;
    Go to Step 2;
    Else
    Output the optimal solution;
    End;
End;

```

ALGORITHM 1

where  $a$  is a positive real number,  $b$  is obtained by formula (18)

$$b = \begin{cases} N \times M \times \frac{f_{\min}}{f_{\min} + f_{\max}} & f \geq f_{\text{avg}} \\ N \times M \times \frac{f_{\max}}{f_{\min} + f_{\max}} & f \leq f_{\text{avg}} \end{cases} \quad (16)$$

where  $N$  is evolution generation,  $M$  is the number of individuals in the population,  $f_{\text{ave}}$  is the average fitness,  $f_{\min}$  is the minimum fitness of the individual,  $f_{\max}$  is the maximum fitness of the individual, and  $f$  is the individual fitness.

**4.4. Selection.** The hierarchical genetic algorithm allows the population to progress from one generation into the next. The selection process is based on the fitness of the individuals, higher fitness results in more frequent selection. There are different selection rules such as the roulette wheel implementation, tournament selection, and elitism. Roulette wheel selection method is used.

Firstly, the fitness of individual  $i(f_i)$  is calculated by formula (15), then the selected probability of individual  $i$  can be calculated by the following formula:

$$P_i = \frac{f_i}{\sum_{k=1}^n f_k}. \quad (17)$$

Secondly, the cumulative probability of each chromosome is calculated by formula (20):

$$q_i = \sum_{i=1}^l P_i, \quad (18)$$

where  $l$  represents the iteration times.

Final, using roulette selection method selects the individual.

**4.5. Crossover.** Some of the genetic material of two individuals are swapped (i.e., crossover operator), creating new individuals (children), who are possibly better than their parents. There are different crossover operator such as mapping crossover, different location crossover, same location crossover and leading crossover. For control gene of chromosome, partially mapping crossover method is used. The process of partially mapping crossover is that firstly, selecting two crossover points from parents' chromosome; secondly, the fragment of parents chromosome between the two crossover points is exchanged; thirdly, for the other genes, if the genes do not belong to the exchanged fragment of parents chromosome, the genes retain their value, otherwise, the value of those genes can be got using partially mapping method. For example, two parent individuals is  $p_1 = [6 \ 5 \ 3 \ 1 \ 2 \ 4]$  and  $p_2 = [5 \ 1 \ 2 \ 6 \ 4 \ 3]$ , if the crossover point is 2 and 4, the offspring of individuals is  $q_1 = [5 \ 1 \ 2 \ 6 \ 3 \ 4]$  and  $q_2 = [2 \ 5 \ 3 \ 1 \ 6 \ 4]$ . The process of crossover operator is shown by Figure 3. The illegal individuals can be avoided with this method.

As the cutting scheduling problem involves process constraints, the crossover operator cannot be used for the parameter genes of chromosome. For example, the cutting pattern  $P_1$  and  $P_2$  are respectively assigned to the cutting machine  $M_5$  and  $M_1$ , after some crossover operator, the cutting pattern  $P_1$  and  $P_2$  may be, respectively, assigned to the cutting machine  $M_1$  and  $M_5$ , but the cutting pattern  $P_1$  cannot be cut on cutting machine  $M_1$ . So some illegal individuals can be generated by using crossover operator for the parameter genes of chromosome.

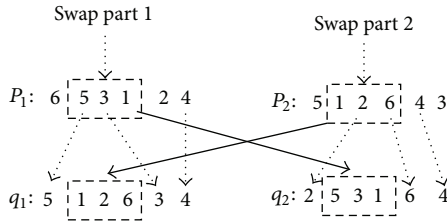


FIGURE 3: The process of crossover operator for control gene.

An adaptive process dynamically adjusts the operator’s probabilities during the process of evolving a solution [28]. In order to accelerate evolutionary speed and enlarge searching scope, an adaptive crossover probability  $p_{cross}$  is designed:

$$p_{cross} = \begin{cases} p_{cross} \times \frac{1}{1 - f_{min}/f_{max}} & \text{if } \frac{f_{min}}{f_{max}} > e \\ p_{cross} & \text{others,} \end{cases} \quad (19)$$

where  $f_{ave}$  is the individual’s average fitness of each generation,  $f_{min}$  is the individual’s minimum fitness of each generation,  $f_{max}$  is the individual’s maximum fitness of each generation.  $f_{min}/f_{max}$  is a positive parameter, whose value reflects the concentration of the whole generation. The bigger the value of  $f_{min}/f_{max}$  is, this algorithm is more likely to fall into local optimal solution. If the the value of  $f_{min}/f_{max}$  exceed a previously set threshold value  $e$  ( $0 < e < 1$ ), the individuals tend to concentration.

**4.6. Mutation.** In order to explore new areas of the search space, the mutation with introducing a variation in the population and avoid premature convergence is needed. There are two types of mutation operators: control genes mutation and parameter genes mutation. For control genes of chromosome, changing sequence variation is used. In other words, randomly two points of parent chromosome is selected, then the value of the selected point of parents chromosome is exchanged each other. For example, the parent chromosome is  $p = [6\ 5\ 3\ 1\ 2\ 4]$ , selected the variation point is 2 and 5, the offspring chromosome is  $p = [6\ 2\ 3\ 1\ 5\ 4]$  after variation operator. For parameter genes of chromosome, the integer variation is used, that is, the parent parameter gene is replaced by an integer  $k$  ( $k \in CP_i$ ) with a certain probability.

An adaptive mutation probability  $p_{muta}$  is designed as follows:

$$p_{muta} = \begin{cases} p_{muta} \times \frac{1}{1 - f_{min}/f_{max}} & \text{if } \frac{f_{min}}{f_{max}} > r \\ p_{muta} & \text{others.} \end{cases} \quad (20)$$

**4.7. Stop Criteria.** In this paper, setting a maximum iteration number has been used, and the algorithm will stop when the iteration reaches the setting maximum iteration.

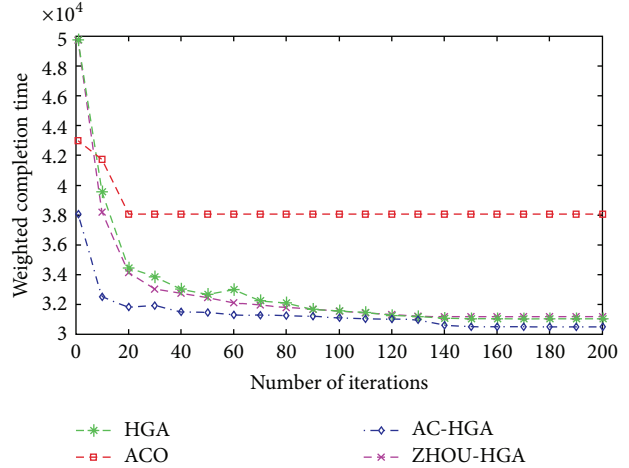


FIGURE 4: Average curve evolution.

### 5. Computational Experiments

In this section, the proposed optimization approach is proven to be available, and the performance of the solution strategy is evaluated by describing an experiment. First of all, a real set of cutting patterns, given by the metal forming factory, is tested by this approach. Later, the test results are reported and analyzed. The date of cutting patterns is given in Table 2, and the cutting machine date has been given in Table 1 in Section 2.

In this paper, the hierarchical genetic algorithm, ant colony algorithm and ant colony—hierarchical genetic algorithm are used for this experiment. Where, the initial population size of hierarchical genetic algorithm is 100; the maximum iterations number of hierarchical genetic algorithm is 200; the threshold value  $e$  is 0.7; the threshold value  $r$  is 0.8; the initial probability of control genes crossover  $p_{cross}$  is 0.7, the initial probability of control genes mutation  $p_{muta}$  is 0.6, the initial probability of parameter genes mutation is  $p_{muta}$  0.4; the information heuristic factor  $\alpha$  of ant colony algorithm is 0.9; the expect heuristic factor  $\beta$  of ant colony algorithm is 6; the pheromone evaporation rate  $\rho$  is 0.3, the maximum iterations number of ant colony algorithm is 200, the initial population size of of ant colony algorithm is 30; the intensity of pheromone  $Q$  is 1000.

The result of the example cutting scheduling problem with process constraints used different algorithms is given by Table 3. The evolution curve of different algorithms is also shown in Figure 4. Learned from Table 3, the result of using ant colony—hierarchical genetic algorithm is best. The minimizing weighted completion time is 30510 min. The Gantt chart of each cutting machine is shown in Figure 5.

### 6. Conclusion and Deductions

This paper discusses the scheduling problem of integrated cutting stock with process constraints. The objective of minimizing weighted completion time of  $n$  cutting patterns for  $m$  nonidentical parallel machines is considered. There are three

TABLE 2: The data of cutting patterns.

Cutting pattern	Weight	Cutting length	Number of parts	Number of punch	Material	Thickness	Available cutting machine
$P_1$	5	43256	29	34	Q235	8	$M_2, M_3, M_4, M_5, M_6$
$P_2$	2	16656	13	16	Q235	8	$M_2, M_3, M_4, M_5, M_6$
$P_3$	4	12533	26	36	Q235	8	$M_2, M_3, M_4, M_5, M_6$
$P_4$	3	28768	22	22	Q235	10	$M_2, M_3, M_4, M_5, M_6$
$P_5$	5	11465	17	20	Q235	10	$M_2, M_3, M_4, M_5, M_6$
$P_6$	4	11909	22	29	Q235	10	$M_2, M_3, M_4, M_5, M_6$
$P_7$	3	118920	37	44	Q235	10	$M_2, M_3, M_4, M_5, M_6$
$P_8$	1	3763	5	8	Q235	10	$M_2, M_3, M_4, M_5, M_6$
$P_9$	5	32729	34	38	Q235	12	$M_2, M_3, M_4, M_5, M_6$
$P_{10}$	3	17464	21	29	Q235	12	$M_2, M_3, M_4, M_5, M_6$
$P_{11}$	4	18671	27	35	Q235	12	$M_2, M_3, M_4, M_5, M_6$
$P_{12}$	4	72579	33	39	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{13}$	5	58832	38	48	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{14}$	4	218320	31	43	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{15}$	1	13640	11	19	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{16}$	3	11393	19	27	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{17}$	5	106680	38	50	Q345	15	$M_2, M_3, M_4, M_5, M_6$
$P_{18}$	5	5928	8	15	Q345	20	$M_3, M_4, M_5, M_6$
$P_{19}$	3	9564	9	13	Q345	20	$M_3, M_4, M_5, M_6$
$P_{20}$	5	83235	17	23	Q345	20	$M_3, M_4, M_5, M_6$
$P_{21}$	1	29441	7	17	Q345	20	$M_3, M_4, M_5, M_6$
$P_{22}$	2	133620	27	36	Q345	20	$M_3, M_4, M_5, M_6$
$P_{23}$	5	71432	14	18	Q345	24	$M_3, M_4, M_5, M_6$
$P_{24}$	1	162360	35	42	Q345	24	$M_3, M_4, M_5, M_6$
$P_{25}$	1	69178	10	16	Q345	24	$M_3, M_4, M_5, M_6$
$P_{26}$	2	85800	12	16	Q345	24	$M_3, M_4, M_5, M_6$
$P_{27}$	1	125400	32	39	Q345	30	$M_4, M_5, M_6$
$P_{28}$	4	159370	26	34	Q345	30	$M_4, M_5, M_6$
$P_{29}$	2	144100	34	46	Q345	32	$M_4, M_5, M_6$
$P_{30}$	1	57115	12	19	Q345	32	$M_4, M_5, M_6$

TABLE 3: The comparison of different algorithms.

Optimization method	Optimum solution	Average value (20)	Iterative times
ACO	38063	38063	200
HGA	31063	31078	200
AC-HGA	30510	30537	200
ZHOU-HGA [25]	31198	32007	200

tasks contributing to solving the cutting scheduling problem. Firstly, a mathematical model for the cutting scheduling problem with process constraints is proposed. Secondly, an ant colony—hierarchical genetic algorithm is designed to solve the mathematical model. In the proposed technique, a set of initial solution is generated by ant colony algorithm, and then an optimum solution is obtained by using selection, crossover, and mutation of the hierarchical genetic algorithm. Where, the control genes are used to determine the sequence of the cutting pattern, and the parameter genes are employed

to identify the cutting pattern assigned on cutting machine. Finally, computational experiment is performed to evaluate the performance of the proposed algorithm. It is verified that the proposed method on cutting scheduling problem with process constraints is effective. In addition, as Figure 4 shown, there reaches a conclusion that the proposed method overweighs other methods. The result of cutting scheduling problem is given by using Gantt chart.

The cutting scheduling plays an important role in sheet cutting process. The existing literatures give a deep research



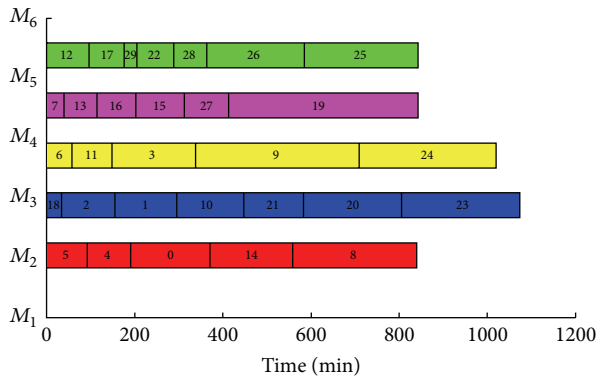


FIGURE 5: The Gantt chart of each cutting machine.

on cutting-stock problem and the combined cutting-stock and lot-sizing problem, but the study of the cutting scheduling problem has been ignored. To the best of our knowledge, the present paper is an effort to consider the cutting scheduling problem with process constraints for the first time, which involve  $n$  cutting patterns for  $m$  nonidentical parallel machines with process constraints. It belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups.

To solve this problem better, an improving hierarchical genetic algorithm is applied. A new encoding mode, that is, the natural number encoding couple with binary encoding, is adopted in this algorithm. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm. Our approach for solving cutting scheduling problem has been applied in practical production process and has been accepted by sheet cutting manufacturers of china. This approach is novel and gives a different perspective for the manufacturing management of sheet cutting.

In this paper, the goal is focus on the minimizing weighted completion time. To achieve the target for solving the cutting scheduling problem, the minimum tardiness or other objectives can be considered. However, some products which needs sheet cutting do not merely includes cutting process (such as structural parts, after cutting process, its need to be bended, or other machined). In the future work, we will focus on the cutting scheduling problem with multi-processes.

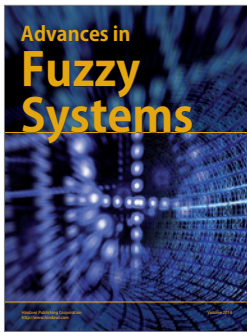
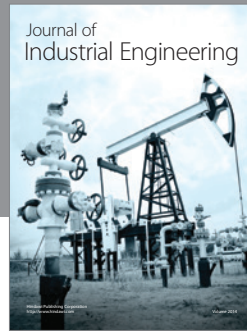
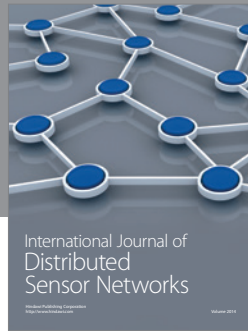
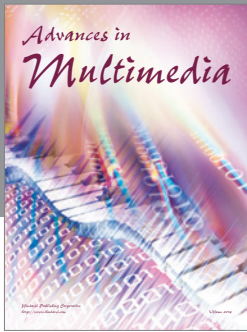
## Acknowledgments

This research work is supported by the Science and Technology Major Project of China [Grant no. 2011ZX04015-011-07].

## References

- [1] J. Blazewicz, V. Boljunčić, S. Martello, and J. Skorin-Kapov, "Combinatorial optimization issues in scheduling," *Journal of Scheduling*, vol. 14, no. 3, pp. 221–223, 2011.
- [2] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 359–370, 2002.
- [3] S. Umetani, M. Yagiura, and T. Ibaraki, "An LP-based local search to the one dimensional cutting stock problem using a given number of cutting patterns," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 5, pp. 1093–1102, 2003.
- [4] J. A. Bennell, K. A. Dowsland, and W. B. Dowsland, "The irregular cutting-stock problem: a new procedure for deriving the no-fit polygon," *Computers and Operations Research*, vol. 28, no. 3, pp. 271–287, 2000.
- [5] Y.D. Cui, J. L. Wu, and H. C. Chen, "Generating multi-section silicon steel sheet cutting patterns in the manufacturing industry of electric generators," *International Journal of Advanced Manufacturing Technology*, vol. 32, no. 3-4, pp. 310–314, 2007.
- [6] Y. Cui and Q. Chen, "Simple heuristic for the constrained two-dimensional cutting problem," *Proceedings of the Institution of Mechanical Engineers B*, vol. 226, no. 3, pp. 565–572, 2012.
- [7] S. Q. Xie, G. G. Wang, and Y. Liu, "Nesting of two-dimensional irregular parts: an integrated approach," *International Journal of Computer Integrated Manufacturing*, vol. 20, no. 8, pp. 741–756, 2007.
- [8] S. L. Nonås and A. Thorstenson, "Solving a combined cutting-stock and lot-sizing problem with a column generating procedure," *Computers and Operations Research*, vol. 35, no. 10, pp. 3371–3392, 2008.
- [9] M. C. N. Gramani and P. M. França, "The combined cutting stock and lot-sizing problem in industrial processes," *European Journal of Operational Research*, vol. 174, no. 1, pp. 509–521, 2006.
- [10] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega*, vol. 27, no. 2, pp. 219–239, 1999.
- [11] T. C. E. Cheng and C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research," *European Journal of Operational Research*, vol. 47, no. 3, pp. 271–292, 1990.
- [12] J. Peng and B. Liu, "Parallel machine scheduling models with fuzzy processing times," *Information Sciences*, vol. 166, no. 1–4, pp. 49–66, 2004.
- [13] A. Anglani, A. Grieco, E. Guerriero, and R. Musmanno, "Robust scheduling of parallel machines with sequence-dependent setup costs," *European Journal of Operational Research*, vol. 161, no. 3, pp. 704–720, 2005.
- [14] J. W. Fowler, S.-M. Horng, and J. K. Cochran, "A hybridized genetic algorithm to solve parallel machine scheduling problems with sequence dependent setups," *International Journal of Industrial Engineering*, vol. 10, no. 3, pp. 232–243, 2003.
- [15] K. Li and S.-L. Yang, "Non-identical parallel-machine scheduling research with minimizing total weighted completion times: models, relaxations and algorithms," *Applied Mathematical Modelling*, vol. 33, no. 4, pp. 2145–2158, 2009.
- [16] N. Van Hop and N. N. Nagarur, "The scheduling problem of PCBs for multiple non-identical parallel machines," *European Journal of Operational Research*, vol. 158, no. 3, pp. 577–594, 2004.
- [17] S. Balin, "Non-identical parallel machine scheduling with fuzzy processing times using genetic algorithm and simulation," *International Journal of Advanced Manufacturing Technology*, vol. 61, no. 9–12, pp. 1115–1117, 2012.
- [18] P. Alcan and H. Balişgil, "A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem," *Advances in Engineering Software*, vol. 45, no. 1, pp. 272–280, 2012.

- [19] C.-F. Liaw, Y.-K. Lin, C.-Y. Cheng, and M. Chen, "Scheduling unrelated parallel machines to minimize total weighted tardiness," *Computers and Operations Research*, vol. 30, no. 12, pp. 1777–1789, 2003.
- [20] P. L. Rocha, M. G. Ravetti, G. R. Mateus, and P. M. Pardalos, "Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times," *Computers and Operations Research*, vol. 35, no. 4, pp. 1250–1264, 2008.
- [21] Y. Mehravaran and R. Logendran, "Bicriteria supply chain scheduling on unrelated-parallel machines," *Journal of the Chinese Institute of Industrial Engineers*, vol. 28, no. 2, pp. 91–101, 2011.
- [22] J. Arnaout, G. Rabadi, and H. Mun, "A dynamic heuristic for stochastic unrelated parallel machine scheduling problem," *International Journal of Operations Research*, vol. 3, no. 2, pp. 136–143, 2006.
- [23] C. Charalambous and K. Fleszar, "Variable neighborhood descent for the unrelated parallel machine scheduling problem," *International Journal on Artificial Intelligence Tools*, vol. 21, no. 4, Article ID 1240019, 18 pages, 2012.
- [24] X. W. Luo, "Study on the Ant colony optimization," *Key Engineering Materials*, vol. 467–469, pp. 300–305, 2011.
- [25] H. Zhou, W. Tang, and B. Niu, "Optimization of multiple traveling salesman problem based on hierarchical genetic algorithm," *Application Research of Computers*, vol. 26, no. 10, pp. 3754–3757, 2009.
- [26] H. Zhang, A. Li, and X. Liu, "Hybrid ant colony algorithm for TSP," *Computer Engineering*, vol. 35, no. 8, pp. 34–37, 2009.
- [27] M. Mavrouniotis and S. Yang, "Ant colony optimization with immigrants schemes in dynamic environments," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN XI '10)*, pp. 371–380, 2010.
- [28] A. Tuson and P. Ross, "Adapting operator settings in genetic algorithms," *Evolutionary Computation*, vol. 6, no. 2, pp. 161–184, 1998.



**Hindawi**

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

