

## Research Article

# A Hybrid IP/GA Approach to the Parallel Production Lines Scheduling Problem

Huizhi Ren<sup>1</sup> and Shenshen Sun<sup>2</sup>

<sup>1</sup>School of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China

<sup>2</sup>School of Information Engineering, Shenyang University, Shenyang 110044, China

Correspondence should be addressed to Huizhi Ren; [renhuizhi@126.com](mailto:renhuizhi@126.com)

Received 11 December 2015; Revised 12 March 2016; Accepted 14 April 2016

Academic Editor: Yourim Yoon

Copyright © 2016 H. Ren and S. Sun. 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.

A special parallel production lines scheduling problem is studied in this paper. Considering the time window and technical constraints, a mixed integer linear programming (MILP) model is formulated for the problem. A few valid inequalities are deduced and a hybrid mixed integer linear programming/constraint programming (MILP/CP) decomposition strategy is introduced. Based on them, a hybrid integer programming/genetic algorithm (IP/GA) approach is proposed to solve the problem. At last, the numerical experiments demonstrate that the proposed solution approach is effective and efficient.

## 1. Introduction

Electric wire and cables are usually continuously produced and can be in coil packaging (or circle packing) [1]. In the large cable manufacturing enterprise, there is usually more than one continuous cable production line and cables can be produced on most of the lines. For the same cable product, the different lines always take different production cost and different production time. It raises a parallel scheduling problem over these production lines. The objective is to minimize the total production cost. The due time of each cable product and technological requirements of each cable production line need to be satisfied. The problem can be viewed as a parallel machine scheduling problem with complicated constraints of continuous cable production lines. It includes two kinds of scheduling decisions: a job allocation decision over multiple machines (cable production lines) and a job sequencing decision on each machine (cable production line).

The parallel machine scheduling problem is a common combinatorial optimization problem and there have been a lot of related researches published. In the 1970s, a series of researches have ever focused on problem complexity analysis, as shown in Table 1, where the symbol description system in Graham et al. [2] was adopted and the identical parallel machine problems with various objectives in literature [3–6]

were proven to be NP-hard while the heterogeneous parallel machine scheduling problem with total complete times was provided an algorithm with  $O(n \log n)$  complexity. Later, Dessouky et al. [7] also proved that heterogeneous parallel machine problems, including  $Q/r_i$ ,  $p_i = 1/C_{\max}$ ,  $Q/p_i = 1/\sum \omega_i c_i$ ,  $Q/p_i = 1/L_{\max}$ , and  $Q/p_i = 1/\sum T_i$ , have  $O(n \log n)$  complexity. Various kinds of algorithms have been developed to solve the parallel machine scheduling problems of NP-hardness, including the intelligent optimization algorithms, such as Tabu Search [8–10], Simulated Annealing [11], and genetic algorithm [12, 13] and optimization algorithms, such as Branch and Bound [14–17], Dynamic Programming [18–20], and Column Generation [21–23].

It is worth noting that a hybrid mixed integer linear programming/constraint programming (MILP/CP) approach has been proposed to solve a class of parallel machine scheduling problems (Harjunkoski et al. [24], Jain and Grossmann [25], Hooker [26], and Maravelias and Grossmann [27]) which are similar to the problem studied in this paper. The approach divides the parallel scheduling problem into a job allocation master problem and a series of single machine scheduling subproblems. It applies the mixed integer linear programming (MILP) to solve the former master problem and constraint programming (CP) to the latter subproblems. Typically, the process needs to be iterated because

TABLE 1: The complexity research of various parallel machine problems in the 1970s.

Literature	Year	Problem variants	Complexity
Karp [3]	1972	$P2//C_{\max}$	NP-hard
Bruno et al. [4]	1974	$P2//\sum \omega_i c_i$	NP-hard
Lenstra [5]	1977	$I/r_i/\sum c_i$	NP-hard
Garey and Johnson [6]	1978	$P//C_{\max}$	NP-complete
Horowitz and Sahni [29]	1976	$Q//\sum c_i$	$O(n \log n)$

the CP always encounters an infeasibility in solving the single machine scheduling subproblem. The hybrid MILP/CP approach can obtain an optimal solution after a number of iterations. Its limitation is the computational inefficiency. It can deal with only small-scale problems in an acceptable CPU time, although some improvements have been developed [27, 28] to reduce the number of iterations.

The hybrid MILP/CP approach can also be applied to the parallel cable production line scheduling problem after revising its CP model according to the particular scheduling requirement of the cable production line. The hybrid MILP/CP approach, though dealing with only some small-scale problems, motivates us to combine the integer programming (IP) and genetic algorithm (GA) together to develop the hybrid IP/GA approach to the parallel cable production lines scheduling problem. In the hybrid IP/GA approach, the traditional exact optimization method (IP) and the intelligent optimization method (GA) can complement each other to strengthen the solving process, similar to what the MILP and CP perform in the hybrid MILP/CP approach. This is our main contribution in the paper.

The outline of this paper is as follows. Section 2 describes constraints of the problem and provides its integer programming model. In Section 3, a few valid inequalities are deduced and a hybrid MILP/CP model is presented. In Section 4, a hybrid IP/GA approach is proposed to resolve the problem. Section 5 tests the proposed solution approach. Section 6 concludes the paper.

## 2. Problem Description and Mathematical Model

**2.1. The Problem Description.** In the cable manufacturing enterprise of our own interest, cables are produced coil by coil through the equipped production lines and thus a cable coil, hereinafter referred to as the “cable,” will be viewed as a job to be scheduled. The cable requires different processing time and cost on different production lines because the lines are configured with different product preferences. The processing time, including the setup time, is assumed to be independent of processing sequence and thus the setup time will be ignored (or included in the processing time) in this problem. The scheduling objective is to minimize the total processing cost.

There are always some cables requiring more cable rubber and the production line needs to provide more heated rubber for these cables, which will form a technical constraint for

the scheduling problem. To facilitate the problem description, the cables that need more rubber will be classified into “type B” and others into “type A.” The cables of type B cannot be processed consecutively in a production line and several cables of type A are needed between processing two successive cables of type B; otherwise the cable rubber quality could be impaired.

Another constraint comes from the time window of each cable. Since the cable cores are handed off from the upstream process, each coil has a release time in the cable production line. In addition, most cables have the contracted delivery time. These two times constitute the time window constraint.

A cable production scheduling problem instance is illustrated in Figure 1. It needs to schedule seven cables (or jobs, labeled as  $J_1, J_2, \dots, J_7$  in Figure 1) over two production lines (labeled as  $L_1$  and  $L_2$ ). Cables  $J_1$  and  $J_2$  belong to type B and others belong to type A. As shown in the top part of Figure 1, each cable has its own time window  $[r_i, d_i]$  and two different processing times,  $p_1^1$  and  $p_1^2$ , corresponding to the two production lines, respectively. A schedule arrangement is shown in the bottom part of Figure 1. For the time window constraint and technical requirement (forbidding cables  $J_1$  and  $J_2$  to be processed continuously on line  $L_1$  or  $L_2$ ), no production line processes cable  $J_2$  and thus the schedule is infeasible.

**2.2. The Mixed Integer Linear Programming Model.** To define and formulate the problem explicitly, the following notations need to be introduced to represent all coils, production lines, coil type parameters, and so on:

$N$ : set of cables and  $N = \{0, 1, 2, \dots, n\}$ , where 0 is a dummy cable.

$M$ : set of cable production lines,  $M = \{1, 2, \dots, m\}$ .

$B$ : set of cables belonging to type B.

$A$ : set of cables belonging to type A,  $A \cap B = \emptyset$ ,  $A \cup B = N$ .

$Q$ : number of necessary cables of type A between successive cables of type B in a production line.

$r_i$ : release time of cable  $i$ ,  $i \in N$ .

$d_i$ : due time of cable  $i$ ,  $i \in N$ .

$c_{ij}$ : production cost of cable  $i$  on line  $j$ .

$p_{ij}$ : production time of cable  $i$  produced on line  $j$ .

$Z$ : a sufficiently large constant.

To represent cable assignment decisions over the production lines and scheduling decisions on each line, we introduce the following decision variables:

$$y_{ij} = \begin{cases} 0, & \text{if cable } i \text{ is produced on line } j; \\ 1, & \text{otherwise.} \end{cases}$$

$$x_{ii'j} = \begin{cases} 0, & \text{if cable } i' \text{ is produced immediately after } i \text{ on line } j; \\ 1, & \text{otherwise.} \end{cases}$$

$$S_i: \text{start time of producing cable } i, i \in N.$$

$$C_i: \text{complete time of cable } i, i \in N.$$

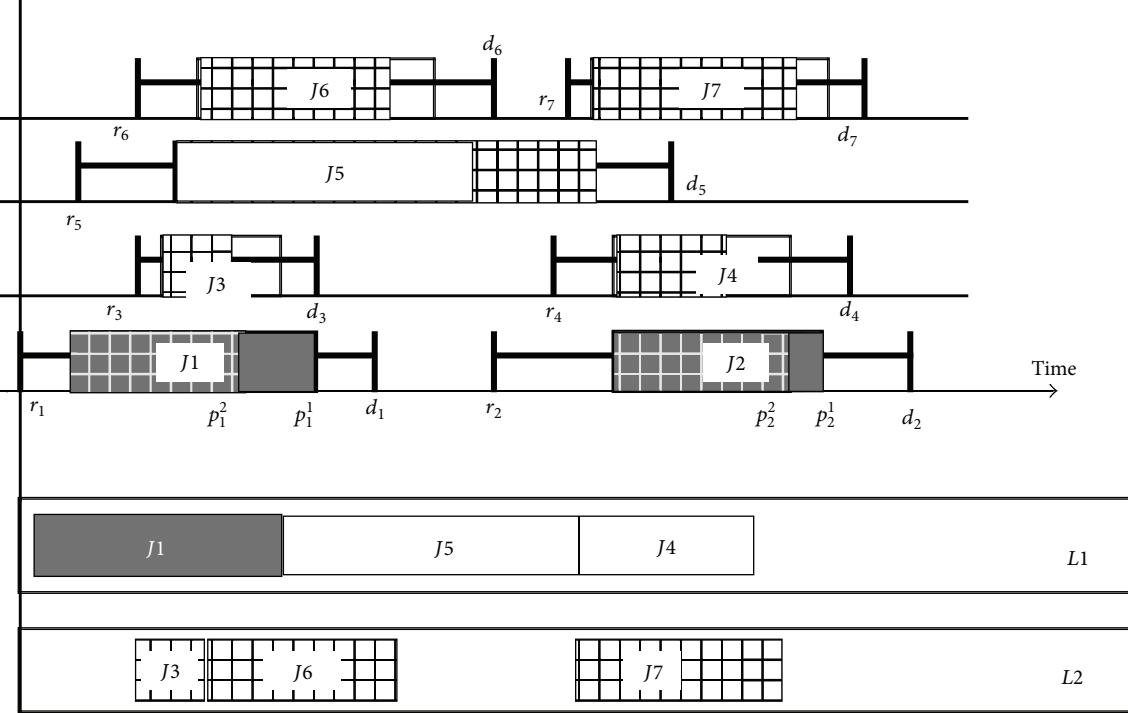


FIGURE 1: An illustration of the cable parallel production lines scheduling problem.

The following auxiliary variables are also introduced to represent the number of consecutive cables of type A on a production line, which is necessary to represent the technological requirement of production lines:

$a_i$ : number of consecutive cables of type A before cable  $i$ ,  $i \in A$ .

Based on the above notations, the parallel cable production lines scheduling problem can be formulated as the following mixed integer linear programming model:

$$\text{Min} \quad \sum_{i \in N \setminus \{0\}} \sum_{j \in M} c_{ij} y_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in M} y_{ij} = 1, \quad i \in N \setminus \{0\} \quad (2)$$

$$\sum_{i \in N, i \neq i'} x_{ii'j} = y_{i'j}, \quad i' \in N \setminus \{0\}, \quad j \in M \quad (3)$$

$$\sum_{i' \in N, i' \neq i} x_{ii'j} = y_{ij}, \quad i \in N \setminus \{0\}, \quad j \in M \quad (4)$$

$$\sum_{i' \in N \setminus \{0\}} x_{0i'j} \leq 1, \quad j \in M \quad (5)$$

$$a_{i'} \leq Z \left( 1 - \sum_{j \in M} \sum_{i \in B} x_{ii'j} \right), \quad i' \in A \quad (6)$$

$$a_{i'} \leq a_i + 1 + Z \left( 1 - \sum_{j \in M} x_{ii'j} \right), \quad i, i' \in A \quad (7)$$

$$\sum_{j \in M} x_{ii'j} \leq 0, \quad i, i' \in B \quad (8)$$

$$a_i \geq Q - 1 - Z \left( 1 - \sum_{j \in M} x_{ii'j} \right), \quad i \in A, \quad i' \in B \quad (9)$$

$$S_i \geq r_i, \quad i \in N \setminus \{0\} \quad (10)$$

$$C_i \leq d_i, \quad i \in N \setminus \{0\} \quad (11)$$

$$S_{i'} - C_i \geq Z \left( \sum_{j \in M} x_{ii'j} - 1 \right), \quad i, i' \in N \setminus \{0\} \quad (12)$$

$$C_i - S_i \geq \sum_{j \in M} p_{ij} y_{ij}, \quad i \in N \setminus \{0\} \quad (13)$$

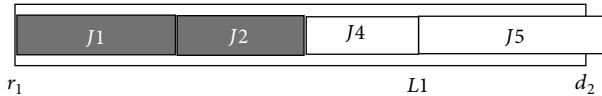
$$x_{ii'j}, y_{ij} \in \{0, 1\}, \quad i, i' \in N, \quad j \in M \quad (14)$$

$$S_i, C_i \geq 0, \quad i \in N \quad (15)$$

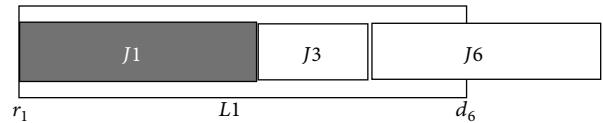
$$a_i \in \{0, 1, 2, \dots, n\}, \quad i \in A. \quad (16)$$

The objective function (1) of the model is to minimize the total production cost of the adopted cable production tasks assignments. Constraints (2)–(5) state the logic relations among the decision variables. Constraint (2) ensures that a production line is assigned to each cable. Constraints (3) and (4) ensure that each cable has one preceding cable and one succeeding cable, respectively. Constraint (5) states that each line has at most one starting cable, which implies that there are at most  $m$  available lines.

Constraints (6)–(9) aim to the technical requirement that there are enough cables of type A between any two successive cables of type B on the same line. Among them, constraint (6) states that  $a_{i'} \leq 0$  if its preceding cable belongs to type B.



(a) A cable assignment against the production capacity inequality



(b) An assignment against the production time equilibrium inequality

FIGURE 2: Cable assignments excluded by the valid inequalities.

Constraint (7) calculates the number of consecutive cables of type A. Constraint (8) prevents the consecutive cables of type B while constraint (9) guarantees no less than Q cables of type A between any two cables on the same line.

Constraints (10) and (11) are the time window constraints. Constraint (12) ensures the cable production sequence on the same line and constraint (13) provides enough production time for each assignment. Constraints (14)–(16) provide value ranges for the variables.

### 3. Valid Inequalities, a Hybrid MILP/CP Solution Method, and Lower Bound for the Problem

**3.1. Valid Inequalities.** In this section, a few valid inequalities will be introduced. To the problem definition, MILP model (1)–(16) is already appropriate and all the valid inequalities are redundant, but they can reduce the computing time if we attempt to directly solve the model through optimization software. It is more important that the valid inequalities can reflect the problem features and work for the heuristic approach.

*Production Capacity Inequality.* Consider the following:

$$\sum_{i \in N} y_{ij} p_{ij} \leq \max \{d_i \mid i \in N\} - \min \{r_i \mid i \in N\}, \quad (17)$$

$j \in M.$

Inequality (17) prevents total production times of assignments to any line more than its capacity in the scheduling horizon. For the case in Figure 1, cables J1, J2, J4, and J5 cannot be assigned to line L1 concurrently because inequality  $p_{11} + p_{21} + p_{41} + p_{51} > \max \{d_i \mid i \in N\} - \min \{r_i \mid i \in N\} = d_2 - r_1$  stands, which conflicts with inequality (17), as shown in Figure 2(a).

There is still a tighter inequality which takes into consideration the production tasks balancing in time.

*Production Time Equilibrium Inequality.* Consider the following:

$$\sum_{i \in \Psi(r_{i'}, d_{i''})} y_{ij} p_{ij} \leq d_{i''} - r_{i'}, \quad (18)$$

$i, i' \in N, d_{i''} > r_{i'}, j \in M,$

where  $\Psi(r_{i'}, d_{i''}) = \{i \mid r_i \geq r_{i'} \text{ and } d_i \leq d_{i''}, i \in N\}$  includes all cables that need to be produced in time window  $[r_{i'}, d_{i''}]$ .

Inequality (17) can be viewed as a special case of inequality (18) and inequality (18) is a tighter inequality. For the case illustrated in Figure 1,  $\Psi(r_1, d_6) = \{1, 3, 6\}$  and cable combination J1, J3, and J6 can be excluded from L1 through inequality (18), although the cable combination meets inequality (17), as shown in Figure 2(b).

It is worth noting that inequality (18) is not adequate to ensure the production equilibrium absolutely in time because the real production duration is less than the given time window and permitted to slide in the time windows. Therefore, the assignment scheme satisfying inequality (18) can still be infeasible for one or more productions even if ignoring the technical requirements (see [24, 28] for more details). In addition, inequality (18) can be significantly tightened again as in Ren and Liu [28].

*Limit of Cable Rate of Type B in a Production Line.* In effect, the technical constraint (9) limits the ratio of the cable numbers of type B and type A in a production line:

$$\sum_{i \in A} y_{ij} \geq (Q - 1) \sum_{i \in B} y_{ij}, \quad j \in M. \quad (19)$$

Inequality (19) provides a filtering mechanism for the cable assignment over the production lines. For the assignment in Figure 2(a), cable combination J1, J2, and J4 can be filtered out from any line when  $Q \geq 2$ , because at least two cables of type A are necessary between cables J1 and J2.

**3.2. Hybrid MILP/CP Approach and Lower Bound for the Problem.** In the hybrid MILP/CP approach [24, 25], the parallel cable production lines scheduling problem should be decomposed into a relaxed master problem, focusing on the cable assignment decision, and a single production line scheduling subproblem. The relaxed master problem can be formulated into an IP model by formulae (1), (2), and (18). The subproblem for a single production line is formulated by the following CP model:

$$\text{All-different}(W); \quad (20)$$

$$\text{if } (W_k \in B) \quad (b_{W_{k+1}} + b_{W_{k+2}} + \dots + b_{W_{k+Q}} = Q) \vee (L < (k + Q)) \quad (21)$$

$$\text{The required production time and time window constraints.} \quad (22)$$

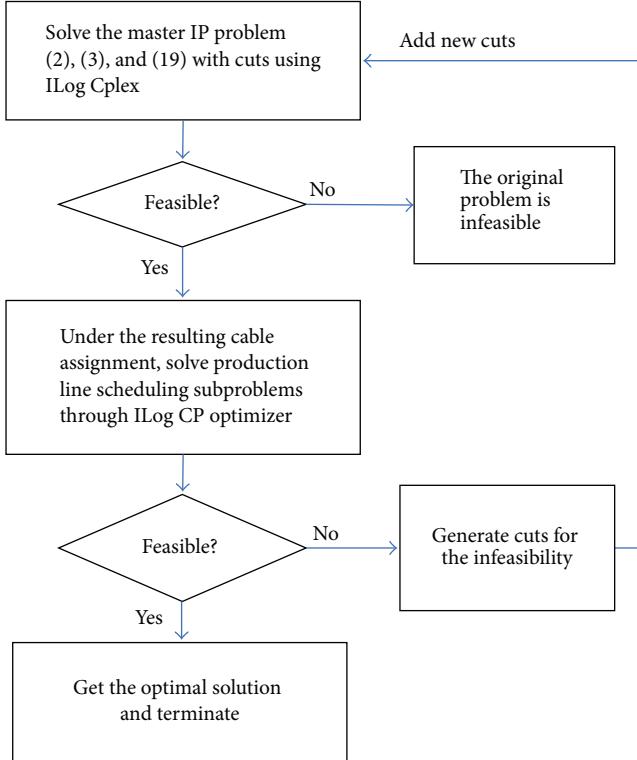


FIGURE 3: Illustration of the hybrid MILP/CP approach.

In the above CP model,  $W_k$  denotes the  $i$ th cable on the considered production line and  $L$  is the number of cables assigned to the line. Formula (20) means that the  $L$  cables are all different from each other. Formula (21) is the technical constraint in the line which requires at least  $Q$  cables of type A immediately after a cable of type B.

As illustrated in Figure 3, the hybrid MILP/CP approach begins with solving the relaxed master problem through solving the corresponding IP model optimally. Here, the IP model is solved by ILOG Cplex, a commercial optimization software package. Based on the optimal solution to the master problem, the cable set for each production line can be determined temporarily. The CP model (20)–(22), solved by ILOG Scheduler (an ILOG CP Optimizer), can deal with the  $m$  single line scheduling subproblems. If the  $m$  subproblems are all feasible and  $m$  single line schedules are obtained, the computing process terminates and the optimal solution is obtained. Otherwise, a “cut,” a kind of constraint to eliminate the current infeasible cable assignments (as formulated by inequality (23)), will be constructed and appended to the master problem model which will trigger a new repeat. The iterated process will repeat until all the  $m$  feasible single line schedules are obtained.

The “cut” in the iterating process can be formulated as

$$\sum_{i \in \Omega} y_{ij^*} \leq \|\Omega\| - 1, \quad (23)$$

where  $j^*$  is the production line encountering infeasibility and  $\Omega$  is a set of cables that contribute to the infeasibility.

The hybrid approach, with improvement in Ren and Liu [28], can solve only small-scale problems, even if ignoring the special technical constraints. For the problem instances of larger size in this paper, it can provide a lower bound and the lower bound can be increased with the more cuts in iterations. But the iterating process can become very time consuming and the lower bound always keeps fixed in many iterations for large problem sizes. Another significance of the hybrid approach is to give an initial cable assignment scheme. Although it is typically infeasible, our heuristic solution can view it as a starting point of the search process.

#### 4. Hybrid IP/GA Approach to the Problem

Genetic algorithm (GA) is a metaheuristic based on evolution population and explores multiple solutions concurrently and thus performs well at global search which is complementary for the optimality (with infeasibility) and solution directivity of the integer programming (IP) method. More GA details can be found in Goldberg [30], Grefenstette [31], and Oliveto and Witt [32]. There are some published studies of genetic algorithm applications in kinds of scheduling problems [33–37]. Through evolution, they find high-quality solutions within reasonable computing time.

Generally, a desirable GA can converge quickly at the same time avoiding premature convergence to low-quality solutions. In practice, it needs a good balance between computational time and the solution quality [32]. To improve the GA searching process, an optimization approach, the integer programming (IP) method based on the IP model (1), (2), and (18), is introduced to the iterating steps. It can not only improve the GA searching process but also provide a better lower bound to evaluate the final solution quality.

**4.1. Framework of the Hybrid IP/GA.** The hybrid IP/GA approach is illustrated in Figure 4. It is characterized by the interaction between IP and GA in the iterating process. The optimization method with IP model provides cable assignment schemes to the GA and the GA feeds back to the IP model some “cuts.” These “cuts” can cut off some cable assignment schemes and tighten the IP model in turn. The tighter model can possibly produce better solution (cable assignment scheme) and better lower bound.

**4.1.1. Generating Cable Assignment Schemes through IP.** As mentioned in Section 3.2, it can provide a cable assignment scheme to solve IP model (1), (2), and (18) through the optimization software (such as ILOG Cplex). To satisfy the technical constraint, inequality (19) can also be inserted into the IP model. The cable assignment scheme can be extended to a solution through applying the earliest due date (EDD) rule to each production line. The solution, which violates one or more of the time window constraints in most cases but is “optimal” for the objective function, can be taken as a seed to generate the initial chromosomes for GA. The infeasibility can be accepted for chromosome but causes a great penalty in the fitness function.

In the evolution process of GA, the IP model can be attached with a series of “cuts,” inequalities as formula (23),

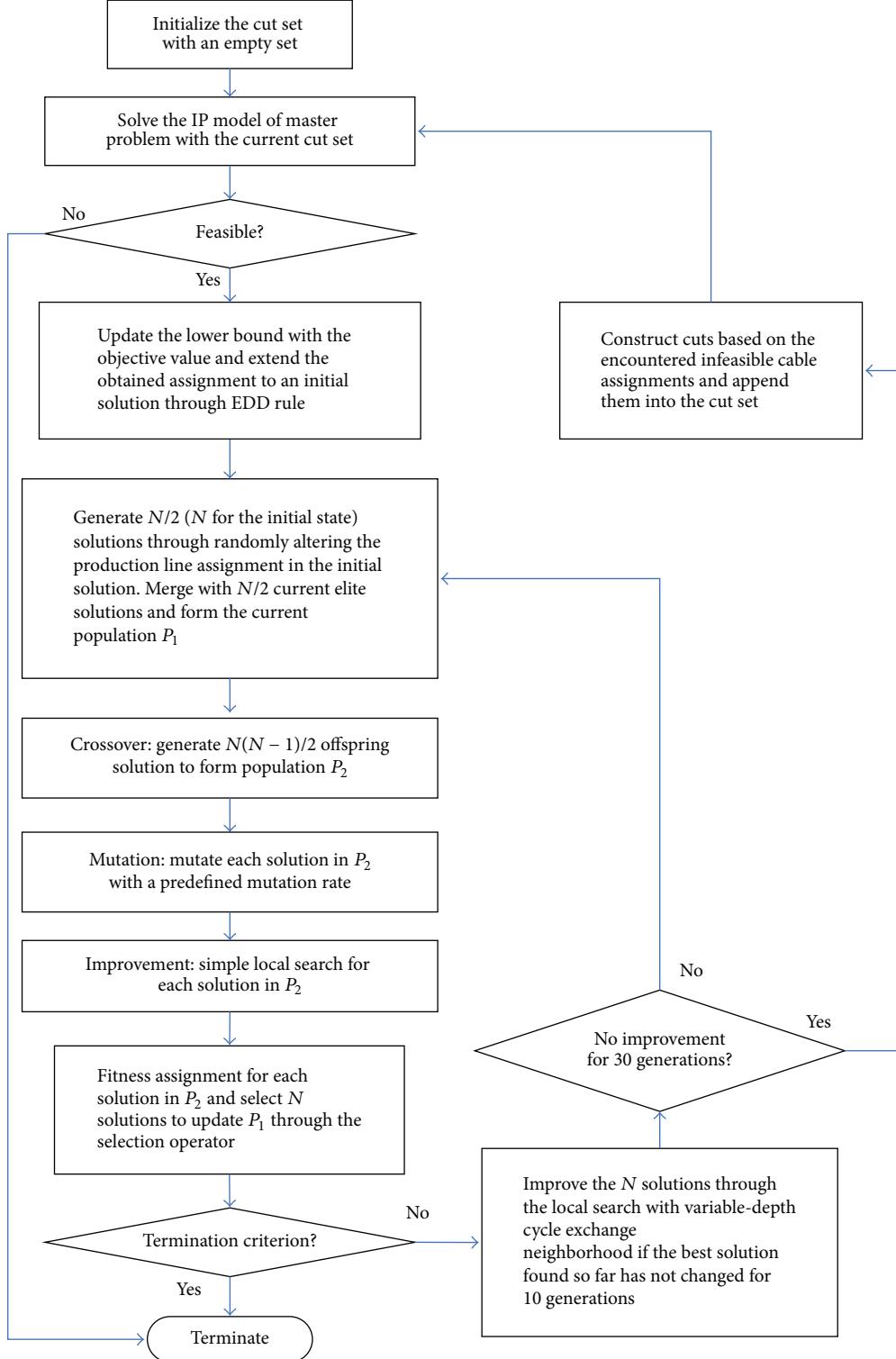


FIGURE 4: The algorithm framework of the hybrid IP/GA approach.

and generate more diverse cable assignment schemes. The cable assignment schemes can also be extended to solutions and provide diverse chromosomes to the GA. In our hybrid algorithm, the IP model solver is called for a certain number of generations.

**4.1.2. Generating Cuts for IP in the GA Generations.** Some of the chromosomes and their infeasibility information in the GA generations need to be retained and form “cuts” to be appended to the IP model. To generate new cable assignment schemes, the cut set of IP model needs to be updated each

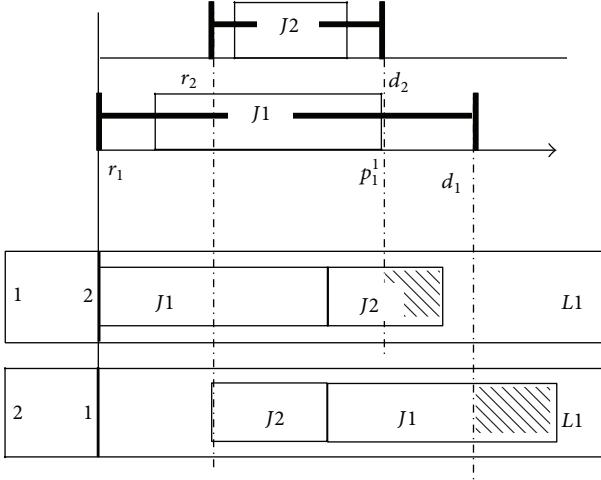


FIGURE 5: The “cut” for an infeasible cable assignment.

time the IP solver is called; otherwise, it will generate the same scheme as the last.

An infeasibility and the “cut” are illustrated in Figure 5. It is infeasible to assign cables  $J_1$  and  $J_2$  to line  $L_1$ , irrespective of the processing sequence, due to the nonpreemption and time window constraint, although it satisfies inequality (18). Based on formula (23), a “cut,” formulated as inequality  $y_{11} + y_{21} \leq 1$ , can be appended into the IP model of the master problem to eliminate the infeasible cable assignment.

It is worth noting that the infeasibility needs to be verified through CP model in Section 3.2 because our single machine scheduling process is not an exact algorithm and cannot guarantee a feasible schedule for each feasible cable combination on a single production line although it does in most cases. If the infeasibility is denied by the CP solver, that is, a feasible schedule is outputted, the cable combination assigned to the line should be accepted and the related solution should be a feasible solution. In the numeric tests, the presumptive case has never happened. On the other hand, some feasible cable assignments can also be used to form “cuts” to make the initial cable assignments more diverse.

Because the CP solver can waste more computing time and too many cuts can decrease the solving speed of the IP model, the infeasible cable combinations (or cuts) need to be selected carefully. In the hybrid approach, only cuts from the assignment schemes with currently optimal objectives can be selected to the CP model.

Typically, IP model (i.e., the relaxed master problem) with more cuts has more chance to improve the lower bound for the problem.

**4.2. Genetic Coding for the Problem.** To facilitate the crossover and mutation operators, the vector-group coding method [38] is adopted. In the coding method, a chromosome is constructed using a sequence of two-dimensional vectors as follows:

$$\text{Ch} = [g_1 \ g_2 \ \dots \ g_n] = \begin{bmatrix} h_1 & h_2 & \dots & h_n \\ k_1 & k_2 & \dots & k_n \end{bmatrix}, \quad (24)$$

where  $g_i = \begin{bmatrix} h_i \\ k_i \end{bmatrix}$  represents the  $i$ th gene in chromosome Ch,  $h_i \in N$  is a cable,  $k_i \in M$  is production line to process cable  $h_i$ , and  $\{h_1, h_2, \dots, h_k\} = N \setminus \{0\}$ . The cables assigned to the same line will be processed in gene order of the chromosome.

The start time of each cable production is equal to the maximum of the complete time of the previous cable and its own release time.

**4.3. Fitness Function and Selection Operator.** Since the objective function needs to be minimized and some chromosome can be infeasible, we define a new referred function by

$$f'(s) = f(s) + N_s P, \quad (25)$$

where  $f(s)$  is the objective function value of chromosome  $s$ ,  $N_s$  is the number of cables that violate their respective time windows or technical constraint (9), and  $P$  is a predefined constant larger than the expected value of  $f(s)$ .

We use a common selection operator: roulette-wheel selection. A dynamic linear fitness function [30]  $U_t(s) = 1.1 \max \{f'(s) \mid s \in P(t)\} - f'(s)$ , where  $t$  is the generation number of the current population  $P(t)$ .

**4.4. Crossover Operator.** Based on the vector-group coded chromosomes, we adopt one-point and two-point extended order crossovers. Let two chromosomes,  $\text{Ch} = [g_1 \ g_2 \ \dots \ g_n]$  and  $\text{Ch}' = [g'_1 \ g'_2 \ \dots \ g'_n]$ , serve as the present chromosomes. The following describes how to cross over to produce new chromosome.

In the one-point extended order crossover, a random integer  $\lambda$  with  $1 \leq \lambda < m$  is first generated to decide the crossover position. New chromosome  $\text{Ch}'' = [g_1 \ g_2 \ \dots \ g_\lambda \ g'_{\lambda+1} \ \dots \ g'_n]$  if  $\{g_1, g_2, \dots, g_\lambda\}$  and  $\{g'_{\lambda+1}, \dots, g'_n\}$  do not include a common cable; otherwise, discard genes with common cable in  $\text{Ch}'$  and insert the genes corresponding to the lost cables in  $\text{Ch}'$ , since the number of common cables must be equal to the number of lost ones. It can generate different chromosome to reverse the  $\text{Ch}$  and  $\text{Ch}'$ .

In the two-point extended order crossover, two random integers  $\lambda_1$  and  $\lambda_2$  with  $1 \leq \lambda_1 < \lambda_2 < m$  are first generated to decide the crossover positions. New chromosome  $\text{Ch}'' = [g'_1 \ g'_2 \ \dots \ g_{\lambda_1} \ g_{\lambda_1+1} \ g_{\lambda_2} \ \dots \ g'_{\lambda_2+1} \ \dots \ g'_n]$  if no replicate cable is involved in the crossover. The treatment for the replicate cable is the same as that in the one-point crossover.

**4.5. Mutation Operator.** The bit-mutation and swap-mutation are adopted. In view of the signification of line assignment decisions for cables, the bit-mutation mutates only the second row of the chromosome genes, replacing it with a random number from  $\{1, 2, \dots, m\}$ . It changes the production line to process the cable.

The swap-mutation swaps randomly selected gene values in a chromosome. It can change only the production sequence of one or two production lines. The technical and time window constraints can be involved in the mutation.

**4.6. Local Search.** To improve the evolution process, a simple local search, based on the swap and insert neighborhoods,

is applied to all chromosomes in the population for all generations while a local search, based on a variable-depth cycle exchange neighborhood (see Luo and Tang [39] for more details), is applied to chromosomes when the best solution found so far has not changed for 10 generations.

The swap neighborhood is obtained through exchanging the assigned production lines of each pair of genes in the given chromosome. The production orders should be exchanged if the pair of genes is corresponding to the same line. The processing sequence should be changed locally according to the technical constraint (formula (9)) and EDD rule which can improve the solution, if the production lines are exchanged. Similar to the swap neighborhood, the insert neighborhood involves inserting a cable to another line or a different position in the same line. The technical constraint has a higher priority than the EDD ruler when they conflict.

## 5. Computational Experiments

To test the performance of the formulated mathematical model and the proposed heuristic approach with improvement strategies, we complemented all the involved programs under the development environment of VC++ 2010 and solved the IP model and CP with *ILog Cplex* and *ILog Scheduler*, respectively. The experiments were all performed on a computer with Win 7 operating system, 2.8 Ghz Intel 2 Core CPU, and 4 GB RAM.

**5.1. Configuration of the Hybrid IP/GA Approach.** GA parameters for all problems were set to be the same: population size = 100, crossover probability = 1, mutation probability = 0.1, and the algorithm stops when gap between the lower bound and the best solution found so far has not been improved for 60 generations, the gap is lower than 0.5%, or the total number of generations reaches 600 (which comes from the computing time limit). The IP master problem was triggered when the best solution found so far has not been improved for 30 generations.

**5.2. Test Problem Instances.** We tested 12 problem instances in the numerical experiments from a practical cable production enterprise, which had five production lines but only four of them were available in these instances. The production cost and times were both estimated based on the empirical data. The number of necessary cables of type A between successive cables of type B in a production line was three and thus  $Q = 3$  in all test problems and the proportion of type B was less than 20% in practice.

To test the developed hybrid IP/GA approach thoroughly, a series of problem instances of six different sizes were generated through imitating the practical data. Each problem size has 12 instances. The processing costs ( $c_{ij}$  for cable  $i$  and production line  $j$ ) were random integers from uniform distribution of  $U[20, 100]$ , which magnified the practical figures about 10 times for more comparability. The processing times were from  $U[30, 60]$ . The release times ( $r_i$ ) were from  $U[0, T_{\max}]$  and due times  $d_i = r_i + \delta p_i$ , where  $T_{\max}$  should be large enough to ensure the feasibility,  $p_i = \text{Max} \{p_{ij} \mid j \in M\}$

TABLE 2: Experimental result for the small sized problems.

Problem	$n \times m$	Objective value		CPU time (seconds)	
		MILP/CP	IP/GA	MILP/CP	IP/GA
1	$12 \times 2$	56	56	3.41	<1
2	$12 \times 2$	58	58	0.86	<1
3	$16 \times 2$	61	61	1.14	1.13
4	$16 \times 2$	79	79	14.87	1.21
5	$15 \times 3$	81	81	3.48	1.79
6	$15 \times 3$	67	67*	143.34	2.76
7	$18 \times 3$	94	94	23.55	4.84
8	$18 \times 3$	87	87	6.56	4.16
9	$16 \times 4$	67	67	31.42	2.89
10	$16 \times 4$	53	53	39.38	4.54
11	$20 \times 4$	102	102*	207.51	5.30
12	$20 \times 4$	113	113	71.35	6.13

\*The optimal solution is obtained but gap is not equal to 0.

is the maximal processing time of the cable  $i$ , and  $\delta$  is a magnification factor randomly from  $U[1.5, 3.5]$ .

**5.3. Computational Results.** The small sized problems were first solved to optimality using the hybrid MILP/CP algorithm and then solved by the proposed hybrid IP/GA. The small sized problem instances were cut out from the 12 practical instances.  $n \times m$  represented the problem size, where  $n$  and  $m$  denote the number of cables and production lines, respectively. Table 2 shows the experimental results with the different problem sizes. For the 12 problems of small size, the hybrid IP/GA also obtained almost all the optimal solutions but did not verify the optimality for two of them.

The experimental results for the 12 practical problems are listed in Table 3. Each problem has four production lines and the problem size is represented by the number of cables. In the table, Gap = (Objective value – Lower bound)/Lower bound is used to evaluate the solution quality. The average relative gap was 2.78% and the average CPU-s was 83.

For the six sets of randomly generated problem instances, Table 4 lists the average solution gap and CPU-seconds. The average relative gap was 2.51% and the average computing time was 121 CPU-seconds. In addition, the number of iterations reaches up to the limit (600 generations) for most of the instances.

To further test the performance of the hybrid IP/GA approach, we also solved the problems with a GA by Vallada and Ruiz [40], which was proposed to solve unrelated parallel machine scheduling problem sharing similarities with our problem. Figure 6 shows the comparison results of the average relative objective value (ARD) of the group of practical problem instances and six sets of randomly generated instances. From the results, the efficiency of our algorithm in solving our problem has been verified. In addition, our proposed hybrid IP/GA approach can provide a better lower bound for the problem.

**5.4. Computational Analysis.** From the computational results, all the problem instances can be resolved in

TABLE 3: Experimental results for the practical problems.

Problem	Size	Objective value IP/GA	LB	Gap (%)	CPU time (seconds)
1	41	172	170	1.32	34
2	46	229	226	1.37	58
3	51	231	222	3.87	117
4	51	244	240	1.52	13
5	55	283	275	2.95	47
6	59	375	367	1.93	143
7	64	396	388	1.98	13
8	67	424	409	3.74	16
9	72	446	421	5.97	231
10	78	487	475	2.47	39
11	86	400	394	1.49	208
12	88	503	480	4.71	71
Average				2.78	83

TABLE 4: The average experiment results for the generated 6 problem sizes.

Group number	Size	Gap (%)	CPU time (seconds)
1	40 × 3	1.32	34
2	80 × 3	3.87	118
3	120 × 3	2.61	156
4	50 × 4	1.57	29
5	100 × 4	2.17	126
6	150 × 4	3.54	265
Average		2.51	121

acceptable computing times (no more than 5 min) by these algorithms. In those experiments, the IP model can provide the cable assignment schemes to the GA as well as the lower bound to evaluate the solution quality. Moreover, we have the following observations:

- (1) In Table 2, the proposed hybrid IP/GA approach performs much better than the hybrid MILP/CP approach in solution efficiency particularly to the instances with larger sizes. For the solution quality, they reached the same objective function value for all the instances but the former failed to reduce the gap to 0, meaning that the solution optimality cannot be validated by its own lower bound.
- (2) In Tables 3 and 4, the computation times increase with the problem size but the solution quality is relatively stable. It implies that the solution quality is seldom affected by the problem size but depends mainly on the permitted number of iterations, since most of iterations reach up to the limit.
- (3) The IP model contributes to improving the solution as well as providing the lower bound. The cable assignment schemes obtained through resolving the IP model, aiming at the optimal objective, can lead the GA to search around the optimum point.

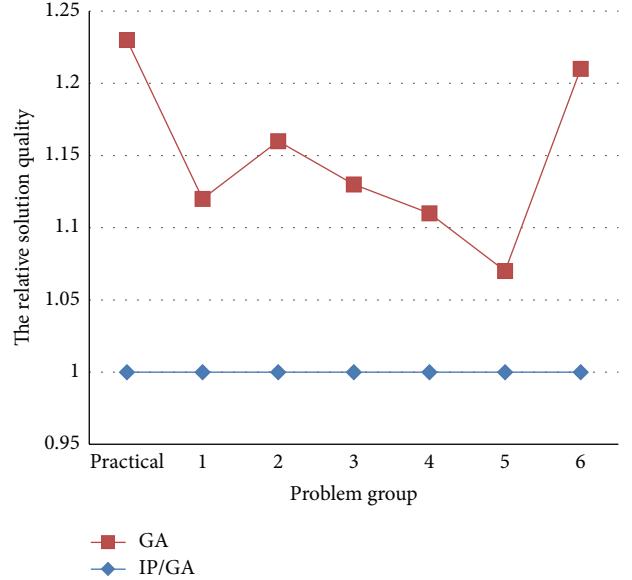


FIGURE 6: The ARD comparison of the IP/GA from an existing GA.

## 6. Conclusion

In this paper, a special parallel production lines scheduling problem was studied. A mixed integer programming model considering time window and technical constraints was first formulated. A few valid inequalities were deduced and a hybrid MILP/CP approach was introduced. Based on them, a hybrid IP/GA algorithm was proposed to solve the problem. The numerical experiments were carried out which demonstrated that the proposed solution approach was effective and efficient.

This study highlighted the advantage of combining the traditional exact optimization (IP) with intelligent optimization (GA) method. Other intelligent optimization methods can also be able to hybridize to the IP (or MILP) method. It is more interesting and challenging to hybridize the IP (or MILP) to more elements of the GA (or other heuristics) and deepen the hybridization for the purpose of improving the solution approach.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This research is partly supported by National Natural Science Foundation of China (71201105), China Postdoctoral Science Foundation (2013M530947), and Liaoning Province Universities Scientific Research Project (L2014480).

## References

- [1] S. Niroomand and B. Vizvari, "Exact mathematical formulations and metaheuristic algorithms for production cost minimization: a case study of the cable industry," *International*

- Transactions in Operational Research*, vol. 22, no. 3, pp. 519–544, 2015.
- [2] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. Rinnooy Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.
  - [3] R. M. Karp, “Reducibility among combinatorial problems,” in *Proceedings of the Symposium on the Complexity of Computer Computations*, New York, NY, USA, March 1972.
  - [4] J. Bruno, J. Coffman, and R. Sethi, “Scheduling independent tasks to reduce mean finishing time,” *Communications of the ACM*, vol. 17, pp. 382–387, 1974.
  - [5] J. K. Lenstra, *Sequencing by Enumerative Methods*, Matematisch Centrum, Amsterdam, Netherlands, 1977.
  - [6] M. R. Garey and D. S. Johnson, “‘Strong’ NP-completeness results: motivation, examples, and implications,” *Journal of the Association for Computing Machinery*, vol. 25, no. 3, pp. 499–508, 1978.
  - [7] M. I. Dessouky, B. J. Lageweg, J. K. Lenstra, and S. L. van de Velde, “Scheduling identical jobs on uniform parallel machines,” *Statistica Neerlandica*, vol. 44, no. 3, pp. 115–123, 1990.
  - [8] R. Hübscher and F. Glover, “Applying tabu search with influential diversification to multiprocessor scheduling,” *Computers and Operations Research*, vol. 21, no. 8, pp. 877–884, 1994.
  - [9] B. Srivastava, “An effective heuristic for minimizing makespan on unrelated parallel machines,” *Journal of the Operational Research Society*, vol. 49, no. 8, pp. 886–894, 1998.
  - [10] C. Smutnicki, “On-time scheduling jobs on parallel machines,” in *Proceedings of the 16th European Conference on Operational Research*, Brussels, Belgium, 1998.
  - [11] T. Kämpke, “Simulated annealing: use of a new tool in bin packing,” *Annals of Operations Research*, vol. 16, no. 1, pp. 327–332, 1988.
  - [12] J. C. Bean, “Genetic algorithms and random keys for sequencing and optimization,” *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.
  - [13] F. Sivrikaya-Serifoglu and G. Ulusoy, “Parallel machine scheduling with earliness and tardiness penalties,” *Computers and Operations Research*, vol. 26, no. 8, pp. 773–787, 1999.
  - [14] H. I. Stern, *Minimizing Makespan for Independent Jobs on Non-Identical Parallel Machine—An Optimal Procedure*, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beersheba, Israel, 1976.
  - [15] H. Belouadah and C. N. Potts, “Scheduling identical parallel machines to minimize total weighted completion time,” *Discrete Applied Mathematics*, vol. 48, no. 3, pp. 201–218, 1994.
  - [16] M. Dell’Amico and S. Martello, “Optimal scheduling of tasks on identical parallel processors,” *ORSA Journal on Computing*, vol. 7, no. 2, pp. 191–200, 1995.
  - [17] S. Martello, F. Soumis, and P. Toth, “Exact and approximation algorithms for makespan minimization on unrelated parallel machines,” *Discrete Applied Mathematics*, vol. 75, no. 2, pp. 169–188, 1997.
  - [18] M. H. Rothkopf, “Scheduling independent tasks on parallel processors,” *Management Science*, vol. 12, no. 5, pp. 437–447, 1966.
  - [19] E. L. Lawler and J. M. Moore, “A functional equation and its application to resource allocation and sequencing problems,” *Mgmt Science*, vol. 16, no. 1, pp. 77–84, 1969.
  - [20] S. K. Sahni, “Algorithms for scheduling independent tasks,” *Journal of the Association for Computing Machinery*, vol. 23, no. 1, pp. 116–127, 1976.
  - [21] Z.-L. Chen and W. B. Powell, “Solving parallel machine scheduling problems by column generation,” *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 78–94, 1999.
  - [22] J. M. van den Akker, J. A. Hoogeveen, and S. L. van de Velde, “Parallel machine scheduling by column generation,” *Operations Research*, vol. 47, no. 6, pp. 862–872, 1999.
  - [23] Z.-L. Chen and W. B. Powell, “A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem,” *European Journal of Operational Research*, vol. 116, no. 1, pp. 220–232, 1999.
  - [24] I. Harjunkoski, V. Jain, and I. E. Grossmann, “Hybrid mixed-integer/constraint logic programming strategies for solving scheduling and combinatorial optimization problems,” *Computers & Chemical Engineering*, vol. 24, no. 2–7, pp. 337–343, 2000.
  - [25] V. Jain and I. E. Grossmann, “Algorithms for hybrid MILP/CP models for a class of optimization problems,” *INFORMS Journal on Computing*, vol. 13, no. 4, pp. 258–276, 2001.
  - [26] J. N. Hooker, “Logic, optimization, and constraint programming,” *INFORMS Journal on Computing*, vol. 14, no. 4, pp. 295–321, 2002.
  - [27] C. T. Maravelias and I. E. Grossmann, “A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants,” *Computers & Chemical Engineering*, vol. 28, no. 10, pp. 1921–1949, 2004.
  - [28] H. Ren and S. Liu, “Study on the hybrid IP/CP approach for a multi-machine assignment scheduling problem,” in *Proceedings of the International Conference on Management and Engineering (CME ’14)*, Shanghai, China, 2014.
  - [29] E. Horowitz and S. Sahni, “Exact and approximate algorithms for scheduling non-identical processors,” *Journal of the Association for Computing Machinery*, vol. 23, no. 2, pp. 317–327, 1976.
  - [30] D. E. Goldberg, *Genetic Algorithms*, Addison-Wesley, Reading, Mass, USA, 1989.
  - [31] J. J. Grefenstette, “Optimization of control parameters for genetic algorithms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.
  - [32] P. S. Oliveto and C. Witt, “On the runtime analysis of the simple genetic algorithm,” *Theoretical Computer Science*, vol. 545, no. 6, pp. 2–19, 2014.
  - [33] S. M. Lee and A. A. Asllani, “Job scheduling with dual criteria and sequence-dependent setups: mathematical versus genetic programming,” *OMEGA, The International Journal of Management Science*, vol. 32, no. 2, pp. 145–153, 2004.
  - [34] R. Ruiz, C. Maroto, and J. Alcaraz, “Two new robust genetic algorithms for the flowshop scheduling problem,” *Omega*, vol. 34, no. 5, pp. 461–476, 2006.
  - [35] M. Kapanoglu and I. O. Koc, “A multi-population parallel genetic algorithm for highly constrained continuous galvanizing line scheduling,” in *Hybrid Metaheuristics*, vol. 4030 of *Lecture Note in Computer Science*, pp. 28–41, Springer, Berlin, Germany, 2006.
  - [36] C. H. Martin, “A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming,” *Omega*, vol. 37, no. 1, pp. 126–137, 2009.
  - [37] J.-W. Chung, S.-M. Oh, and I.-C. Choi, “A hybrid genetic algorithm for train sequencing in the Korean railway,” *Omega*, vol. 37, no. 3, pp. 555–565, 2009.

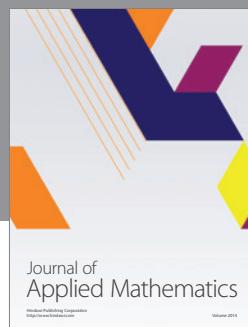
- [38] W. Yin, M. Liu, and C. Wu, "A new genetic algorithm for parallel machine scheduling with process constraint," *Acta Electronica Sinica*, vol. 29, no. 11, pp. 1482–1485, 2001.
- [39] J. X. Luo and L. X. Tang, "A new ILS & SS algorithm for parallel-machine scheduling problem," *Acta Electronica Sinica*, vol. 31, no. 6, pp. 917–924, 2005.
- [40] E. Vallada and R. Ruiz, "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times," *European Journal of Operational Research*, vol. 211, no. 3, pp. 612–622, 2011.



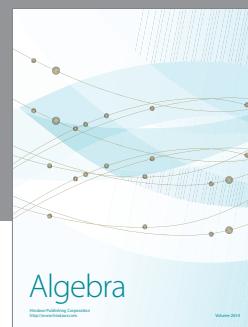
Advances in  
Operations Research



Advances in  
Decision Sciences



Journal of  
Applied Mathematics



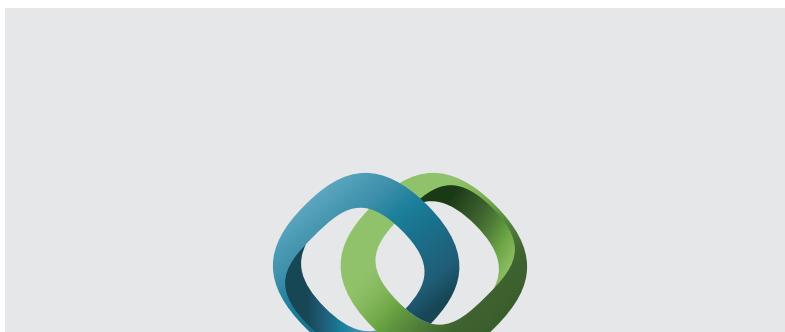
Algebra



Journal of  
Probability and Statistics



The Scientific  
World Journal

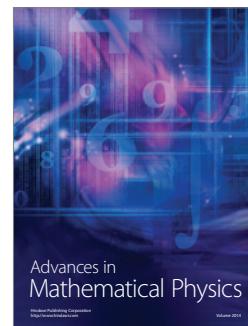


Hindawi

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



International Journal of  
Combinatorics



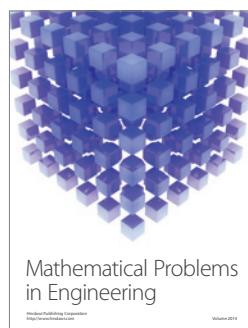
Advances in  
Mathematical Physics



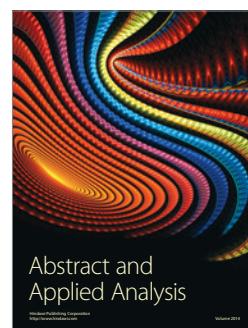
Journal of  
Complex Analysis



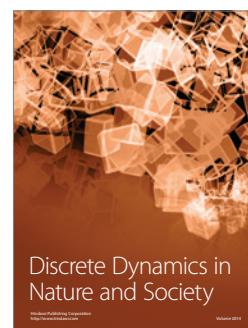
Journal of  
Mathematics



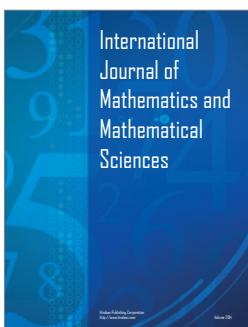
Mathematical Problems  
in Engineering



Abstract and  
Applied Analysis



Discrete Dynamics in  
Nature and Society



International  
Journal of  
Mathematics and  
Mathematical  
Sciences



Journal of  
Discrete Mathematics



Journal of  
Function Spaces



International Journal of  
Stochastic Analysis



Journal of  
Optimization