Models and Algorithms of Production Scheduling in Tandem Cold Rolling

ZHAO Jun[1] LIU Quan-Li[1] WANG Wei[1]

Abstract The complexity of production scheduling problem in cold rolling line is analyzed, which is formulated as two parts, namely, the coil-merging optimization and the rolling batch planning. The optimization of steel coil merging is constructed as a multiple container packing problem (MCPP) that is computed by a new proposed algorithm, discrete differential evolution (DDE), in this paper. A specific double traveling salesman problem (DTSP) is modeled for the rolling batch planning, and a hybrid heuristic method on the basis of evolutionary mechanism and local search is presented to solve this model. The experimental results with real production data from Shanghai Baosteel Co. Ltd. show that the production scheduling method suggested in this paper is effective.

Key words Cold rolling production scheduling, multiple container packing problem (MCPP), differential evolution, traveling salesman problem, heuristic algorithm

Cold rolling is an important production procedure after continuous casting and hot rolling in steel industry. Recently, with the manufacturers in steel plant giving more attention to manufacturing executive system (MES) and because of the increase in diversity of cold rolling products, the production capability of tandem cold rolling mill makes a close influence on downstream logistic balance and economic benefits of whole plant. Therefore, production scheduling in tandem cold rolling mill has become a significant problem that connects the production management with the manufacturing process in steel industry.

Currently, research on cold rolling production in the published work mainly focuses on optimizing parameters of tandem rolling mill such as roller torque, tension, and rolling power, which are necessary to improve the production capability and reduce the setup costs[1−2]. However, few papers concentrated on the scheduling problems from the point of rolling production planning. [3] established a software framework on a cold rolling manufacturing line and applied a task distribution method to plan the online operations. The objective of that paper was to build an entire logistic model and distribution rules of rolling plant but without providing details of the cold rolling process. [4] constructed a just-in-time based model on cold rolling plant, and adopted a dynamic online algorithm associated with expert system to solve it. In that paper, the scheduling element was the production order, which did not aim at the steel coils material. [5] suggested a production distribution method to tandem cold rolling line for satisfying the customer′s requirement and the inventory limitation. However, that model only presented an appropriate productivity distribution based on machine capacity rather than the designed operational sequence.

In this paper, we formulate the optimized scheduling problem of tandem cold rolling mill in Shanghai Baosteel as a two-stage process, which consists of the steel coil merging and the cold rolling planning. First, for the purpose of optimizing the coil-merging process that is practically performed in the later period of cold rolling procedure, a multiple container packing problem (MCPP) model is summed up. A new rapid algorithm for combinational optimization called discrete differential evolution (DDE) is proposed to solve this MCPP under consideration of guaranteeing the solution quality. Second, a double traveling salesman problem (DTSP) to model cold rolling batch planning process is presented. Because of the width constraints in a batch plan, the specific DTSP is divided into two solving sections for reducing the complexity of this model. To the separated planning model, a heuristics based on evolutionary mechanism and local search with tabu list is investigated to obtain the optimized satisfactory solution in practice.

The rest of this paper is organized as follows. The production process in the rolling plant is described and the two-stage scheduling problems are modeled in Section 1. A new proposed DDE algorithm for merging optimization and the approach to the scheduling models in Section 1 are presented in Section 2. Section 3 gives computational experiments with production data from Baosteel Co. Ltd. Finally, Section 4 presents some conclusions.

1 Production scheduling model in cold rolling mill

1.1 Problem description

The cold rolling plant has a very complicated production flow, which includes picking line, cold rolling mill, annealing line, etc. Fig. 1 shows a flow instance of a cold rolling plant in Shanghai Baosteel Co. Ltd. and its corresponding products.

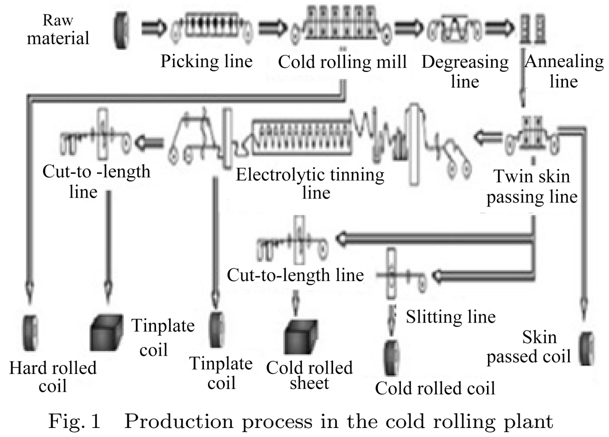

Fig. 1 Production process in the cold rolling plant

In the whole manufacturing line, most processes produce the outputs for not only the following units, but also sales market as ultimate products. Being a crucial upstream equipment, the cold rolling mill has the largest production capacity in the plant, over 900 thousand tons per year.

Received January 15, 2007; in revised form November 12, 2007
Supported by National Natural Science Foundation of China (60474058, 60604026), National High Technology Research and Development Program of China (863 Program) (2007AA04Z156)
1. Research Center of Information and Control, Dalian University of Technology, Dalian 116024, P. R. China
DOI: 10.3724/SP.J.1004.2008.00565

Therefore, its performance makes an intense influence on the other manufacturing lines of the plant. Besides, a more important reason is that the subsequent annealing line is the bottleneck of the entire production line. Because the number of coils charged in each annealing furnace is invariable, generally not more than 5, for improving the efficiency of this bottleneck line, we could merely increase the weight of steel coils charged in the annealing furnace. From the manufacturing technique aspect, the cold rolling is the unique unit that can adjust the coil weight in front of the annealing line. Thus, the production planning and scheduling of tandem cold rolling mill plays a significant role in the whole production line. The cold rolling process can be described as follows. First, a batch of steel coils is unwrapped by an open-coil machine, and then the coils are continuously welded and rolled. Second, at the end of the rolling line, the rolled strips are cut and re-coiled to generate new coils. In this phase, the previous coils may be merged to the new coils with appropriate weight for the annealing line. Thus, the scheduling in tandem cold rolling consists of two key operations: to schedule the rolling sequence of a batch of steel coils in the first stage, and to optimize the steel coil merging for a better coil weight in the second stage. The objective of sequence planning is to smoothen the rolling jumps of width, entrance gauge, and exit gauge for reducing the abrasion on rollers under many rolling constraints; whereas, the objective of coil-merging optimization is to maximize the re-coiled coil weight for the subsequent line. At present, the planning and scheduling approach through specialized workers′ experience is widely used in cold rolling plants, which makes the production capability of tandem rolling mill rather low.

In this paper, we study a two-stage method to deal with this production scheduling problem of cold rolling line. In order to simplify the coupled complex problem, it is conversely considered that the coil merging is first handled in this paper although such process is actually performed at the end of rolling in manufacturing. Second, on the basis of these virtual merged coils, the rolling sequence scheduling is arranged.

1.2 Coil-merging optimization

The coil-merging operation is actually completed at the end of cold rolling line in manufacturing process, so that the coils that have identical width and exit gauge in the batch for maximizing the single coil weight are merged. Depending on technical conditions, there are two coil-merging modes in the rolling plant of Shanghai Baosteel, which are forming one-coil (FOC) mode and forming two-coil (FTC) mode, respectively. FOC mode is that the steel coils satisfying the technical demands are to be merged into one coil whose weight is limited in the given restriction, whereas FTC mode is that the available coils are to be merged into two equal coils and the weight of each coil meets the given restriction. In this paper, we summarize this problem as an MCPP with unknown container number, which belongs to the family of NP-hard problems and is very unlikely to devise the polynomial algorithms to exactly solve it[6]. We regard the set of available coils as the items that need to be packed into containers; and the weight limitation of a single coil denotes the container capacity. According to above-mentioned production conditions, there are two kinds of containers in this MCPP model. Assuming that the set of available coils satisfy the merging constraints, the merging model is defined as follows.

$$\min\{K_1\sum_{j=1}^{C_1}(S_{\max}-w_j^{non})+K_2\sum_{k=1}^{C_2}(S_{\max}-w_k^{FOC})+ K_3\sum_{l=1}^{C_3}(2S_{\max}-w_l^{FTC})\} \tag{1}$$

s. t.

$$\sum_{i=1}^{n}x_{ij}w_i\le S_{\max},\qquad j=1,2,\cdots,C_1 \tag{2}$$

$$\sum_{i=1}^{n}y_{ik}w_i\le S_{\max},\qquad k=1,2,\cdots,C_2 \tag{3}$$

$$\sum_{i=1}^{n}z_{il}w_i\le 2S_{\max},\qquad l=1,2,\cdots,C_3 \tag{4}$$

$$\sum_{j=1}^{C_1}x_{ij}+\sum_{k=1}^{C_2}y_{ik}+\sum_{l=1}^{C_3}z_{il}=1,\qquad i=1,2,\cdots,n \tag{5}$$

where

$$x_{ij}=\begin{cases}1 & \text{if the } i\text{-th coil is the non-merged coil}\\ 0 & \text{otherwise}\end{cases}$$

$$y_{ik}=\begin{cases}1 & \text{if the } i\text{-th coil is packed into } k\text{-th FOC packet}\\ 0 & \text{otherwise}\end{cases}$$

$$z_{il}=\begin{cases}1 & \text{if the } i\text{-th coil is packet into } l\text{-th FTC packet}\\ 0 & \text{otherwise}\end{cases}$$

and $S_{\max}$ is the maximal allowable coil weight corresponding to the rolling width; w_i is the weight of the i-th coil before merging; C_1 is the amount of nonmerged coils; C_2 is the merged amount of FOC mode packets; C_3 is the merged amount of FTC mode packets; w_j^{non} represents the weight of the j-th nonmerging coil; w_k^{FOC} represents the weight of the k-th FOC mode packet; w_l^{FTC} represents the weight of the l-th FTC mode packet; n is the amount of the batch of available steel coils; K_1, K_2, and K_3 denote the penalty coefficients of the nonmerging mode and the two merging modes, respectively.

(1) is the objective function that is to minimize the total difference between the packet weight and the maximal allowable weight, i.e., to maximize the number of the merged coils. (2) is the requirement of coil weight before merging, which addresses the input restriction of this model. (3) and (4) are to meet the constraints of the FOC mode and the FTC mode, respectively. (5) guarantees the disposal uniqueness of the batch of steel coils.

1.3 Cold rolling batch planning

After coil-merging operation, the virtual merged coils will be arranged for the optimized rolling sequence based on technical constraints in order to reduce the rollers abrasion and the setup cost. In rolling process, the roller-changing operation is a time-consuming work, and it can augment the setup costs additionally. It is permitted to change the rollers once in a rolling batch plan because a width jump from narrow to wide is necessary; otherwise, the rolling marks will be transferred on the following steel coils. Fig. 2 shows the rolling width profile of a batch plan in production practice, in which the width must be transited from wide to narrow in a roller-changing period cycle. In manufacturing, smoothening the width jump and the gauges jumps of adjacent rolled coils is very important to improve rolling quality because the larger the jumps are, the more serious rollers abrasion generates. And that might debase the surface grade of the following steel coils.

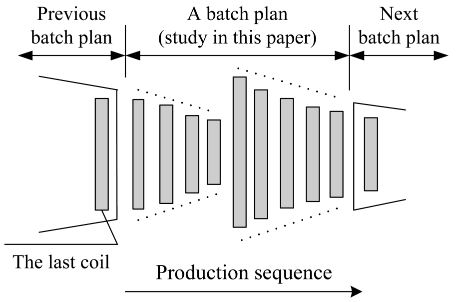

Fig. 2 The rolling width profile in a rolling batch plan

In this paper, a model of double traveling salesman problem (DTSP) without return is proposed to demonstrate this sequence optimization considering that there are two width-decreasing parts in a batch plan (see Fig. 2). For the global optimization, the batch planning needs not only to optimize the inner rolling sequence within a batch, but also to take into account the connection with the previous batch plan for flexible transition. In this paper, the last coil of the previous batch is regarded as the starting city of the first salesman, and the widest coil in current batch is regarded as the starting city of the second salesman. The jump penalty of the two adjacent coils on width and gauges is regarded as the distance of two cities. The objective of this model is to minimize the total distance traveled by the two salesmen under the constraints. The model is formulated as follows.

$$\min \left\{ \sum_{i,j} p_{ij} \cdot (u_{ij} + v_{ij}) \right\} \tag{6}$$

s. t.

$$\sum_{j=1}^{n} (u_{ij} + v_{ij}) = 1, \qquad i = 0, 1, 2, \cdots, n \text{ and } i \neq j \tag{7}$$

where

$$u_{ij} = \begin{cases} 1, & \text{if coil } j \text{ is continuously rolled} \\ & \text{after coil } i, \text{ and } i, j \text{ are in the} \\ & \text{first width-decreasing part} \\ 0, & \text{otherwise} \end{cases}$$

$$v_{ij} = \begin{cases} 1, & \text{if coil } j \text{ is continuously rolled} \\ & \text{after coil } i, \text{ and } i, j \text{ are in the} \\ & \text{second width-decreasing part} \\ 0, & \text{otherwise} \end{cases}$$

and $i = 0$ denotes the last steel coil in the previous batch plan; n is the number of virtual merged steel coils; $p_{ij} = width_{ij} + egauge_{ij} + ogauge_{ij}$, $width_{ij}$, $egauge_{ij}$ and $ogauge_{ij}$ represent the penalty of width, gauge in entrance, and gauge in exit, respectively.

(6) is the objective function of the planning model that minimizes the jumps penalty of continuous rolling steel coils. (7) denotes that each planned coils can be dealt with only once.

2 Solution to the scheduling models

Because the input steel coils used to plan rolling sequence are the virtual coil-merged new coils, the two-stage scheduling of cold rolling line need to be solved in order. For enhancing the practicability of the presented scheduling system, we add the human-machine coordination into the software system. The entire scheduling flow is firstly illustrated in Fig. 3.

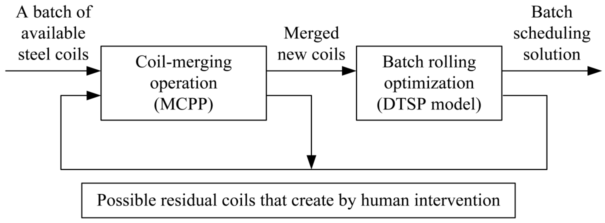

Fig. 3 The operation flow of batch scheduling in cold rolling line

For solving the engineering practice problem, it is crucial to sustain high running efficiency and simple implementation except for the system reliability. However, due to the NP-hard characteristics of this scheduling model, it is difficult to get a satisfactory solution in acceptable computation period using general mathematic programming even intelligent optimization[10]. In contrast, the amount of batch available coils to be scheduled is relative large, might be more than 100 sometimes. Therefore, to find a high-speed computational algorithm is necessary for the effective solution. [7− 8] demonstrated that the differential evolution (DE) had a good computational quality, especially on speed; and, the comparison with other intelligent search method (simulated annealing, stochastic search, and so on) proved its advantages. Although DE is to aim at the continuous problem, we are enlightened by its search idea and propose a new high-speed colony based algorithm for the discrete combinatorial optimization.

2.1 Discrete differential evolution algorithm

The DE calculating continuous optimization generates new individuals with mutation and recombination, and selects the colony with tournament selection. The detail is formulated below.

It is assumed that $P = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N\}$, $\boldsymbol{X}_i \in \mathbf{R}^n$ is a colony, where n is the dimension amount in solution space and N is the scale of this colony. The mutation and recombination of DE is based on (8) and (9).

$$\boldsymbol{u}_i = \boldsymbol{X}_{r1} + F \cdot (\boldsymbol{X}_{r2} - \boldsymbol{X}_{r3}) \tag{8}$$

$$\hat{X}{}_i^j = \begin{cases} X_i^j, & \text{if rand}(j) > CR \text{ and } j \neq I_i \\ u_i^j, & \text{otherwise} \end{cases} \tag{9}$$

The variables $\boldsymbol{X}_{r1}$, $\boldsymbol{X}_{r2}$, and $\boldsymbol{X}_{r3}$ are three random picked individuals from current generation except for the individual $\boldsymbol{X}_i$. The parameter F is the differential amplification ratio. (8) is to create a new coordinative vector, which produces a new solution $\overset{\wedge}{\boldsymbol{X}}_i$ corresponding to $\boldsymbol{X}_i$ (see

(9)). X_i^j is the value of jth dimension of $\boldsymbol{X}_i$; rand(j) is the random number related to the jth dimension, whose range is $[0, 1]$; CR is the recombining coefficient that affects the convergence speed; I_i is a random of selected integer among $1, 2, \cdots, n$. To decide whether the new vector will become a member of next generation, the trial vector $\hat{\boldsymbol{X}}_i$ is compared tothe target vector $\boldsymbol{X}_i$ with a greedy criterion.

Enlightened by the conception of DE algorithm, we propose the discrete differential evolution (DDE) in order to solve the combinatorial optimization, the MCPP of steel coil-merging problem. Considering a class of discrete problems, a feasible solution can be described as an n-bit integer string, which represents a location of the search space. For example, an integer string $(3, 1, 5, 4, 2, 6)$ represents a location in a 6-dimensional space. Since DE uses the principle of (8) and (9) to build the new trail vector, we need to redefine new concepts that include the difference between two vectors, amplifying vector, and recombining two vectors in DDE.

1) The difference between two n-bit vectors.

To the continuous problem, the difference between two vectors describes the relative transition between two positions in solution space. Depending on such thought, we redefine the difference through a set of ordinal position sequence exchanges $\boldsymbol{X}_1 - \boldsymbol{X}_2$, with which vector $\boldsymbol{X}_1$ can be constructed from $\boldsymbol{X}_2$. The element in the set is the position sequence of dimension. Here is an example below, which denotes that the vector $\boldsymbol{X}_2$ can be transferred to the vector $\boldsymbol{X}_1$ with orderly exchanges of the integers between 1st dimension and 6th dimension, 2nd dimension and 3rd dimension, 3rd dimension and 6th dimension, and 4th dimension and 5th dimension.

$\boldsymbol{X}_1$: (4, 5, 3, 1, 6, 2)

$\boldsymbol{X}_2$: (3, 2, 5, 6, 1, 4)

$\boldsymbol{X}_1 - \boldsymbol{X}_2$: $\{(1, 6), (2, 3), (3, 6), (4, 5)\}$

2) The amplifying operation.

On the basis of (8), the difference between two vectors should be multiplied by a differential amplification ratio. We redefine the amplification as (10) considering the discrete characteristics of this algorithm, which means that the amplification is to get a subset of the discrete difference of two vectors.

$$F \cdot (\boldsymbol{X}_{r2} - \boldsymbol{X}_{r3}) = \bigcup_{k=1}^{\mathrm{int}[F \times s]} (p_k, q_k) \tag{10}$$

The variable s is the amount of elements in the set of $\boldsymbol{X}_1 - \boldsymbol{X}_2$, and the operator int$[a]$ is to get the maximal integer not larger than a. Thus, (10) denotes that the amplification takes the preceding int$[F \times s]$ elements of the set to displace the location of $\boldsymbol{X}_{r1}$.

3) The recombination of the two vectors.

If (9) is directly used to build the new individual in DDE, then a situation that two identical integers exist in an n-bit string may appear, which violates the encoding rule. To avoid the violation and to realize the principle of partial information preservation simultaneously, a crossover on bit positions of a string is applied to the trail vector and the original vector for establishing a new individual. The crossover selects a group of positions in the two strings, and then exchanges the other genes except the selected ones. An example is as follows.

Crossover on the basis of bit position:

The selected bit position: 2 5 7

Parent 1: $3\ \boxed{4}\ 1\ 7\ \boxed{6}\ 2\ \boxed{5}\ 9\ 8$

Parent 2: $5\ \boxed{8}\ 9\ 1\ \boxed{3}\ 6\ \boxed{2}\ 7\ 4$

Child 1: $8\ \boxed{4}\ 9\ 1\ \boxed{6}\ 3\ \boxed{5}\ 2\ 7$

Child 2: $4\ \boxed{8}\ 1\ 7\ \boxed{3}\ 6\ \boxed{2}\ 5\ 9$

$$A = \mathrm{int}[n(1 - CR)] \tag{11}$$

In this operation, (11) is proposed to calculate the amount of the selected bit positions on the basis of concept of parameter CR. In addition, unlike DE, we suggest that the presented DDE should produce the new individuals by tournament selection within three vectors, the two crossover child-vectors, and the original vector, by which the search scope of DDE can be improved.

2.2 Solution to the coil-merging optimization

This paper applies the DDE algorithm that is proposed on the basis of engineering practice to the MCPP model. The conversion of individual code to solution space is to orderly fill the FOC and FTC containers with the sequential coils determined by an integer string. In this process, the next container will be loaded until the previous one has been packed to the maximal capacity, and the objective function can be calculated by (1). Because the parameter F affects the mutation degree, an adaptive value of this parameter is formulated as (12) for prevention from premature convergence, where I_m is the total number of iterations; I is the current iteration number.

$$F = F_{\min} + \frac{F_{\max} - F_{\min}}{I_m} \times I \tag{12}$$

The solving process is described as follows.

Step 1. Assign the number of the available batch coils that meet the merging technical requirements. Initialize the DDE algorithm parameters.

Step 2. Randomly generate N n-bit strings as the initial colony. Convert each individual of the current colony into the corresponding solution space using the mentioned container filling method. Calculate the objective function value of each solution and get the best-so-far solution and its function value.

Step 3. Update the parameter F by (12). Use the DDE to produce a new generation of the colony.

Step 4. Searching each new individual of current generation for best-so-far solution. Check whether the maximal iteration is reached. If so, output the best-so-far solution and the relevant optimized coil-merging solution. Otherwise, repeat Step 3.

2.3 Solution to the cold rolling batch planning

The rolling batch planning is to give an optimized rolling sequence of the merged steel coils. In this paper, a hybrid heuristic method to this planning model is proposed, in which we firstly arrange the coils whose width are larger than that of the last coil of the previous batch as the rolling section 1 because of the width profile in a batch plan and the connection with the previous batch (see Fig. 4). In this section, these steel coils will be rolled right after changing rollers. We summarize the planning of this section as a symmetrical TSP that is solved by an evolutionary local search after the initialization of conversely sorting the coils by width. The detail of the evolutionary local search can be seen in [9]. The objective function is calculated on the basis of (6).

In the second step, the rest of these coils are planned as another rolling section of total batch, in which we consider the connection with not only the previous batch, but also the rolling section 1. The width profile is illustrated in Fig. 5. This model is a typical DTSP without priority, which emphasizes on reasonable grouping and arrangement of these coils. In this paper, a speedy strategy based on evolutionary computation and local search with tabu list is presented to obtain the optimized result. The solving steps are as follows.

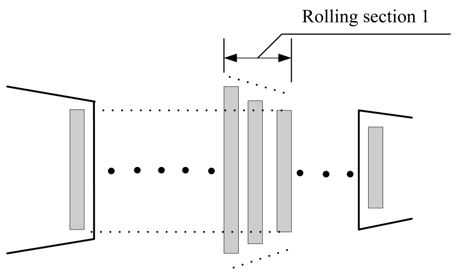

Fig. 4 Rolling section 1 in a rolling batch planning

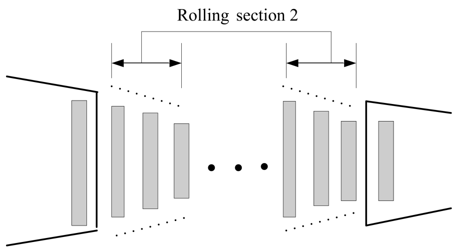

Fig. 5 Rolling section 2 in a rolling batch planning

Step 1. Number the rest of the r coils and generate the p r-bit binary strings, where "0" represents that the ith coil is distributed into the former rolling section processed before changing rollers; "1" represents that the ith coil is distributed into the latter rolling section. Initialize the parameters of algorithm including the length of tabu list, the maximal iteration number, the local search iteration number, etc.

Step 2. With the distributed result in Step 1, there will be two independent TSPs. A rapid algorithm on the basis of local search and tabu search is followed. First, conversely sort the coils in each parts of rolling section 2 respectively by width. Second, randomly select 3 coils in each part and permute them in the selected positions again, by which the local search set is generated. Third, search the local search set for a better feasible solution and put the selected coils into the tabu list.

Step 3. Repeat the local search for q times without violating the length of tabu list. Then obtain and keep in memory the optimized solution of current grouping. The objective function is the sum of the two TSPs fitness values.

Step 4. Clean up tabu list. Evolve the p binary strings with tournament selection and natural crossover for next generation. Then distribute the new individuals and repeat Steps 2 and 3 for a better grouping and its related rolling sequence.

Step 5. Check whether the maximal evolutionary iteration number is reached. If so, output the current best-so-far distribution and the corresponding rolling sequence. Otherwise, repeat Step 4.

Until now, the two rolling sections have been optimized, respectively. We incorporate them to form the ultimate optimized rolling batch plan.

3 Computational experiments

On the basis of the proposed method, a large number of computational experiments with real production data in the cold rolling plant of Shanghai Baosteel Co. Ltd. were implemented. The scheduling results including merging and batch planning were given a remarkable evaluation by the specialized scheduling workers of this plant and are superior to the results from manual scheduling approach that widely used before. The constraint relationship between width and maximal coil weight in practice is given in Table 1.

3.1 Coil-merging experiments

For indicating the computational efficiency and reliability of the new proposed DDE algorithm, we present a comparison between man-made coil-merging method (MM), Partheno-genetic algorithm (PGA), tabu search (TS) and DDE. The running parameters in TS, and PGA have been optimized to appropriate values. The reason why we used PGA instead of GA is that it is inconvenient for GA to run the crossover operation when encoding individuals with ordinal strings. Choosing TS is because it is a relatively speedy algorithm in intelligent optimization. In this paper, we show 5 batches of coils data, in which the amount of different coils in each batch are involved. The computational experiments were run on an IBM Server with 3.0 GHz, 4 G RAM. The comparison results are shown in Table 2. In

Table 1 The constraint between the rolling width and the maximal single weight

Rolling width (mm)	Maximal coil weight (t)	Rolling width (mm)	Maximal coil weight (t)
Less than 730	12.24	856 ~ 870	14.36
731 ~ 740	12.41	871 ~ 890	14.79
741 ~ 770	12.58	891 ~ 900	15.13
771 ~ 800	13.09	901 ~ 920	15.30
801 ~ 820	13.60	921 ~ 950	15.64
821 ~ 840	13.94	More than 950	16.00
841 ~ 855	14.28		

Table 2 The comparison between 4 kinds of solving methods

Batch	Original	Merged coil number				Real iteration number				Computation time			
No.	coil number	MM	PGA	TS	DDE	MM	PGA	TS	DDE	MM (about)	PGA (s)	TS (s)	DDE (s)
1	68	45	42	42	42	–	100	100	100	30 min	39.8	29.6	31.1
2	75	52	46	46	46	–	150	150	100	30 min	56.7	41.0	34.3
3	83	57	51	52	51	–	200	200	100	40 min	87.6	69.4	39.2
4	102	76	68	70	68	–	300	300	200	1 h	111.5	98.7	70.6
5	105	79	70	71	70	–	300	300	200	1 h	113.1	99.4	71.3

Table 3 The result of batch coil-merging optimization with manual method (52 merged coils)

No.	Coil No.	Width (mm)	Gauge in entrance (mm)	Gauge in exit (mm)	Merged weight (t)	Allowable maximum (t)	d_i (Maximum - merged)
1	C60055900	732	2.0	0.23	10.71	12.41	1.70
2	C56535200	732	2.0	0.23	10.56	12.41	1.85
3	C60055300	732	2.0	0.23	9.31	12.41	3.10
4	C56460610	732	1.8	0.23	7.70	12.41	4.71
5	C56460600	732	1.8	0.23	7.10	12.41	5.31
6	C56463400	780	2.0	0.25	12.29	13.09	0.80
7	C56475100	780	2.3	0.28	12.08	13.09	1.01
8	C56697710	830	2.0	0.25	12.58	13.94	1.36
9	C56459700	830	2.0	0.25	12.41	13.94	1.53
10	C56700900	840	1.8	0.20	13.50	13.94	0.44
11	C56697700	840	2.0	0.25	12.51	13.94	1.43
12	C56458200	840	2.0	0.25	11.26	13.94	2.68
13	C56458100	840	2.0	0.25	11.06	13.94	2.88
14	C56565400	840	2.0	0.25	9.96	13.94	3.98
15	C56565500	840	2.0	0.25	7.54	13.94	6.40
16	C56658800	847	2.0	0.22	12.68	14.28	1.60
17	C56698200	880	2.0	0.28	12.21	14.79	2.58
18	C56697600	880	2.0	0.28	11.59	14.79	3.20
19	C56698300	900	1.8	0.20	14.75	15.13	0.38
20	C56465920	920	2.0	0.23	13.16	15.30	2.14
21	C56466000	920	2.0	0.23	11.58	15.30	3.72
22	C56466100	920	2.0	0.23	10.19	15.30	5.11
23	C56463120	920	2.0	0.23	9.80	15.30	5.50
24	C56463130	920	2.0	0.23	7.58	15.30	7.72
25	C56465320	925	2.0	0.28	14.62	15.64	1.02
26	C56717900	925	2.0	0.28	14.54	15.64	1.10
27	C56656700	950	2.5	0.35	13.36	15.64	2.28
28	C56657700	950	2.5	0.35	12.15	15.64	3.49
29	C55649630	950	2.5	0.35	12.07	15.64	3.57
30	C56462000	950	2.5	0.35	8.25	15.64	7.39
31	C56462110	950	2.0	0.35	6.79	15.64	8.85
32	C60053900	1 000	2.3	0.33	14.60	16.00	1.40
33	C56702100	1 000	2.3	0.33	13.36	16.00	2.64
34	C56734800	1 000	2.3	0.33	13.79	16.00	2.21
35	C56734900	1 000	2.3	0.33	13.58	16.00	2.42
36	C56735600	1 002	3.0	0.57	15.41	16.00	0.59
37	C56459910	1 002	3.0	0.57	14.99	16.00	1.01
38	C56338100	1 020	2.5	0.35	14.21	16.00	1.79
39	C56328500	1 020	2.5	0.35	14.05	16.00	1.95
40	C56317700	1 020	2.5	0.35	12.82	16.00	3.18
41	C56319200	1 020	2.5	0.35	12.64	16.00	3.36
42	C56337300	1 020	2.5	0.35	10.16	16.00	5.84
43	C56315410	1 020	2.5	0.35	9.35	16.00	6.65
44	C56312560	1 020	2.5	0.35	8.40	16.00	7.60
45	C56338500	1 020	2.75	0.45	15.31	16.00	0.69
46	C56337140	1 020	2.75	0.45	15.19	16.00	0.81
47	C60014500	1 020	3.0	0.70	13.12	16.00	2.88
48	C55621780	1 020	3.0	0.70	12.01	16.00	3.99
49	C55659000	1 020	3.0	0.70	11.53	16.00	4.47
50	C55659120	1 020	3.0	0.70	9.35	16.00	6.65
51	C60054890	1 020	3.0	0.70	13.35	16.00	2.65
52	C56698900	1 020	3.0	0.50	14.79	16.00	1.21

The average difference between the maximal allowable weight and the merged weight: $\overline{d_i} = \frac{1}{52}\sum_{i=1}^{52} d_i = 3.13$ t.

Table 4 The result of batch coil-merging optimization with DDE (46 merged coils)

No.	Coil No.	Width (mm)	Gauge in entrance (mm)	Gauge in exit (mm)	Merged weight (t)	Allowable maximum (t)	d_i (Maximum-merged)
1	C60055900	732	2.0	0.23	12.22	12.41	0.19
2	C56535200	732	2.0	0.23	12.17	12.41	0.24
3	C60055300	732	2.0	0.23	11.91	12.41	0.50
4	C56460600	732	1.8	0.23	9.08	12.41	3.33
5	C56463400	780	2.0	0.23	12.29	13.09	0.80
6	C56475100	780	2.3	0.28	12.08	13.09	1.01
7	C56697710	830	2.0	0.25	13.41	13.94	0.53
8	C56459700	830	2.0	0.25	11.58	13.94	2.36
9	C56700900	840	1.8	0.20	13.50	13.94	0.44
10	C56697700	840	2.0	0.25	13.50	13.94	0.44
11	C56458200	840	2.0	0.25	13.11	13.94	0.83
12	C56458100	840	2.0	0.25	12.96	13.94	0.98
13	C56565400	840	2.0	0.25	10.76	13.94	3.18
14	C56658800	847	2.0	0.22	12.68	14.28	1.60
15	C56698200	880	2.0	0.28	14.41	14.79	0.38
16	C56697600	880	2.0	0.28	9.39	14.79	5.40
17	C56698300	900	1.8	0.20	14.75	15.13	0.38
18	C56465920	920	2.0	0.23	15.02	15.30	0.28
19	C56466000	920	2.0	0.23	13.78	15.30	1.52
20	C56466100	920	2.0	0.23	13.69	15.30	1.61
21	C56463120	920	2.0	0.23	9.82	15.30	5.48
22	C56465320	925	2.0	0.28	15.13	15.64	0.51
23	C56717900	925	2.0	0.28	14.04	15.64	1.60
24	C56656700	950	2.5	0.35	15.40	15.64	0.24
25	C56657700	950	2.5	0.35	15.29	15.64	0.35
26	C55649630	950	2.5	0.35	14.11	15.64	1.53
27	C56462000	950	2.5	0.35	7.82	15.64	7.82
28	C60053900	1 000	2.3	0.33	15.61	16.00	0.39
29	C56702100	1 000	2.3	0.33	15.35	16.00	0.65
30	C56734800	1 000	2.3	0.33	14.69	16.00	1.31
31	C56734900	1 000	2.3	0.33	9.68	16.00	6.32
32	C56735600	1 002	3.0	0.57	15.68	16.00	0.32
33	C56459910	1 002	3.0	0.57	14.72	16.00	1.28
34	C56338100	1 020	2.5	0.35	15.71	16.00	0.29
35	C56328500	1 020	2.5	0.35	15.45	16.00	0.55
36	C56317700	1 020	2.5	0.35	14.92	16.00	1.08
37	C56337300	1 020	2.5	0.35	14.24	16.00	1.76
38	C56315410	1 020	2.5	0.35	12.16	16.00	3.84
39	C56312560	1 020	2.5	0.35	9.15	16.00	6.85
40	C56338500	1 020	2.75	0.45	15.52	16.00	0.48
41	C56337140	1 020	2.75	0.45	14.98	16.00	1.02
42	C60014500	1 020	3.0	0.70	15.92	16.00	0.08
43	C55621780	1 020	3.0	0.70	15.46	16.00	0.54
44	C55659000	1 020	3.0	0.70	14.63	16.00	1.37
45	C60054890	1 020	3.0	0.70	13.35	16.00	2.65
46	C56698900	1 020	3.0	0.50	14.79	16.00	1.21

The average difference between the maximal allowable weight and the merged weight: $\overline{d_i} = \frac{1}{46}\sum_{i=1}^{46} d_i = 1.64$ t.

order to obtain basically equal optimized effectiveness (merged coil number), these 4 kinds of methods perform numbers of different iterations. We can see that although TS has a higher speed than DDE when iterating the same number (see Batch 1), TS needs more iteration numbers when the coil number in the batch is large (see Batchs 2∼5), which means that the computational time of DDE is less than that of TS. The PGA is the most time-consuming method in these search algorithms, and all of them are speedier than manual method. From search effectiveness aspect, DDE has the same quality with PGA, and both of them are better than TS method (see Batchs 3∼5). All in all, DDE algorithm is the relatively best search method in coil-merging optimization.

Additionally, for showing the coil-merging effects of the DDE and the manual method, we randomly selected Batch 2 in Table 2 as a comparison example, in which 75 original coils are involved. Because of the priority of the two merging modes and the average coil weight in FTC mode, the penalty coefficients of (1) were specified as $K_1 = 0.5$, $K_2 = 0.3$, and $K_3 = 0.2$. The parameters of DDE are as follows: The colony scale is 100, $CR = 0.2$, $F_{\max} = 1$, and $F_{\min} = 0.4$. Table 3 shows the result of manual method, which yields 52 merged coils and Table 4 shows the result of the DDE, which yields 46 merged coils. In each of the two tables, the last column gives the difference between the maximal allowable weight and the new merged weight (d_i). On comparing the results between Tables 3 and 4, obviously, we can conclude that the amount of new coils by DDE is smaller than that by manual method, which means the weight of new coil by DDE is larger. In addition, we can also calculate that the average difference between maximal allowable value and new merged weight by manual method is 3.13 tons, whereas the average difference by the DDE is 1.64 tons. It is indicated that for the same batch, the average merged weight by DDE is obviously closer to the allowable weight limitation than that by manual method. Therefore, the DDE optimization effect is much better than that of the manual method.

3.2 Rolling planning experiments

Following the merging optimization, the merged coils are arranged for batch planning. Table 5 gives the empirical settings of jumps penalty coefficients about width, gauge in entrance, and gauge in exit. The number of binary strings used for distribution is 50; the length of tabu list is 5.

For maintaining the continuity of whole scheduling process, we still take the Batch 2 as the rolling planning example. The result of merging optimization in Table 4 is as the input of batch rolling planning. The size of last coil in previous batch is that the width is 930 mm; the gauges in entrance and in exit are 2.0 mm and 0.33 mm, respectively. Fig. 6 shows the width-changing diagram of Batch 2 with manual scheduling and the proposed method. With the proposed method, the first rolling part has 8 merged coils whose widths are decreased from 920 mm to 780 mm. After the rollers are changed, the second part starts from the coil whose rolling sequence number is 9, and the width is decreased from 1 020 mm to 732 mm once again. Similarly, there are also two width-decreasing parts formed by manual method in Fig. 6, which meet the rolling technical requirement. However, for the jumps of gauges (entrance gauge and exit gauge) aspect, the proposed method performs better than manual method because it is obvious in Figs. 7 and 8 that the gauge changes with the proposed method are smoother than those with the man-made scheduling (see Figs. 7 and 8). It is indicated that the proposed models and algorithms have simultaneously considered the jumps of width, entrance gauge, and exit gauge under technical constraints to reduce the rollers abrasion. Therefore, the proposed method in this paper is feasible and effective.

In this paper, the merging optimization and the batch planning can separately complete the corresponding scheduling assignment in practical manufacturing process (illustrated in Fig. 3). In system application, the appropriate human-machine coordination can increase the intensity of human decision and improve the flexibility of this scheduling software system.

Table 5 Penalty structure of the width and gauge jumps

Width jumps (mm)	Penalty	Entrance gauge jumps (mm)	Penalty	Exit gauge jumps (mm)	Penalty
0 ∼ 5	1	0 ∼ 0.10	5	0 ∼ 0.02	5
6 ∼ 10	5	0.11 ∼ 0.20	10	0.03 ∼ 0.05	10
11 ∼ 30	10	0.21 ∼ 0.30	20	0.06 ∼ 0.10	30
31 ∼ 50	30	0.31 ∼ 0.40	30	0.11 ∼ 0.15	50
50 ∼ 100	50	0.41 ∼ 0.50	50	0.16 ∼ 0.20	100
100 ∼ 200	100	more than 0.50	100	more than 0.20	200
more than 200	200				

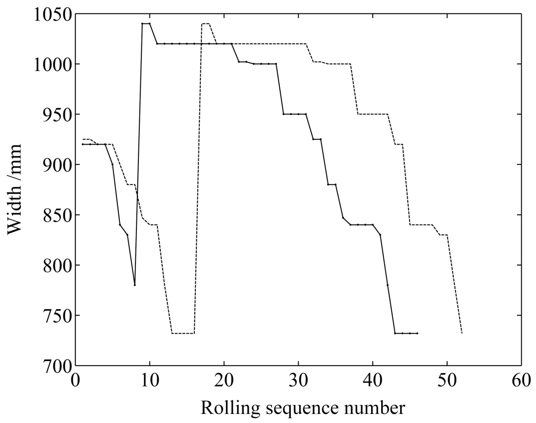

Fig. 6 The width changing in Batch 2

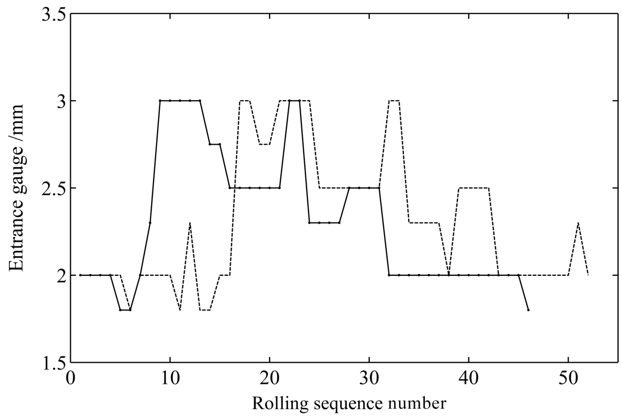

Fig. 7 The entrance gauge changing in Batch 2

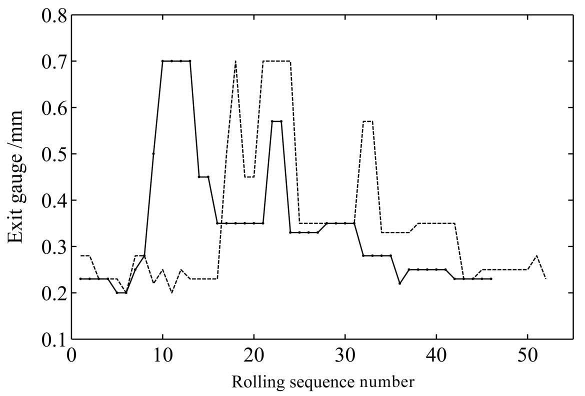

Fig. 8 The exit gauge changing in Batch 2

4 Conclusion

Because the requirements of increasing the efficiency and decreasing the setup cost are ceaselessly heightened in steel manufacturing industry, the key technologies in MES are regarded as one of the most important aspect to enhance the economic benefits of enterprises. In this study, two correlative programming problems in tandem cold rolling mill are modeled. A new combinatorial algorithm is established to solve the coil-merging optimization model, and the heuristic method on the basis of intelligent computation is proposed to plan cold rolling sequence. A great number of computational experiments with real data provided by Shanghai Baosteel have shown that the proposed scheduling approach are feasible and can be applied to the practical production.

References

1 Wang D D, Tieu A K, de Boer F G, Ma B, Yuen W Y D. Toward a heuristic optimum design of rolling schedules for tandem cold rolling mills. *Engineering Applications of Artificial Intelligence*, 2000, **13**(4): 397−406

2 Dixit U S, Dixit P M. Application of fuzzy set theory in the scheduling of a tandem cold-rolling mill. *Journal of Manufacturing Science and Engineering*, 2000, **122**(3): 494−500

3 Peng Wei, Shi Hai-Bo. Software framework and tasks distribution for unit-operation planning. *Computer Integrated Manufacturing Systems*, 2002, **8**(3): 213−218 (in Chinese)

4 Li Yao-Hua, Ning Shu-Shi, Wang Wei, Zhang Da-Bo. Production planning model and algorithm of rolling plant based JIT. *Control Engineering of China*, 2004, **11**(4): 37−40 (in Chinese)

5 Wang Wen-Peng, Yang Zai-Bu, Li Tie-Ke. Batch planning and scheduling method for cold rolling production line. *Metallurgical Industry Automation*, 2006, **30**(5): 11−15 (in Chinese)

6 Raidl G R. The multiple container packing problem: a genetic algorithm approach with weighted codings. *ACM SIGAPP Applied Computing Review*, 1999, **7**(2): 22−31

7 Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, **11**(4): 341−359

8 Storn R. Differential evolution design of an IIR-Filter with requirements for magnitude and group delay. In: Proceedings of International Conference on Evolutionary Computation. Nagoya, Japan: IEEE, 1996. 268−273

9 Freisleben B, Merz P. A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In: Proceedings of International Conference on Evolutionary Computation. Nagoya, Japan: IEEE, 1996. 616−621

10 Gen M, Cheng R W. *Genetic Algorithm and Engineering Design*. New York: Wiley-Interscience. 1997

ZHAO Jun Received his bachelor degree in industrial automation from Dalian Jiaotong University in 2003. Now he is a Ph. D. candidate in control theory and control engineering in Research Center of Information and Control at Dalian University of Technology. His research interest covers industrial production scheduling, computer integrated manufacturing, and intelligent optimization and application. Corresponding author of this paper.
E-mail: jun.z@163.com

LIU Quan-Li Received his bachelor and master degrees in control theory and engineering from Northeastern University, in 1998 and 2000 and received his Ph. D. degree from Department of Automation, Dalian University of Technology in 2004, respectively. Now he is an associate professor at Dalian University of Technology. His research interest covers production planning and scheduling, industrial process control, and discrete event simulation. E-mail: liuql@dlut.edu.cn

WANG Wei Received his bachelor, master, and Ph. D. degrees in industrial automation from Northeastern University, in 1982, 1986, and 1988, respectively. He is currently professor at Research Center of Information and Control, Dalian University of Technology. Previously he was a post-doctor at Division of Engineering Cybernetics, Norwegian Science and Technology University during 1990∼1992, professor and vice director of the National Engineering Research Center of Metallurgical Automation during 1995∼1999, and a research fellow at Department of Engineering Science, University of Oxford during 1998∼1999. His research interest covers adaptive control, computer integrated manufacturing, and computer control of industrial process.
E-mail: wangwei@dlut.edu.cn