Scheduling Hot Slabs in Parallel Hot Rolling Production Lines by Column Generation

Yang YANG1,2)* and Bo ZHANG1,3)

1) The Institute of Industrial Engineering and Logistics Optimization, Northeastern University, Shenyang, 110819 P. R. China.
2) State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819 P. R. China.
3) Liaoning Key Laboratory of Manufacturing System and Logistics, Northeastern University, Shenyang, 110819 P. R. China.

(Received on May 19, 2014; accepted on July 28, 2014)

This paper is to determine the slab schedule on parallel hot rolling lines with consideration of chain precedence constraints, which restricts the slabs belonging to a chain to be successively processed without interruption in one production line. Moreover, energy loss increases as its waiting time before being processed increases, since the slabs dynamically arrive with high temperature. The objective of the problem is to minimize the total energy loss costs. Two kinds of energy loss cost that are linearly and nonlinearly dependent of its waiting time are concerned. The strongly NP-hard feature of the problem stimulates us to develop branch-and-pricing algorithm to solve it. Firstly, the model is formulated as a set partitioning model. For the strongly NP-hard sub-problem, dynamic programming based on state-space relaxation method is derived to tackle it. A branch-and-bound procedure is applied for the case that the solution obtained at the root node is not integral. The whole algorithm is implemented by C language on a Pentium IV 3.0 GHz PC. Results on 25 different problem scenarios demonstrate that the proposed algorithm can solve the instances up to 120 slabs optimally within a reasonable computation time. The gaps between optimal solution and lower bound are no more than 0.0395% for the linear case and 0.0315% for the nonlinear case, respectively.

KEY WORDS: hot rolling; scheduling; chain precedence constraint; column generation; state-space relaxation.

1. Introduction

The motivation for this research stems from hot slab scheduling in parallel production lines of the practical Iron-and-Steel Complex. Hot rolling line (HRL) is an operation in which hot slabs are rolled down into hot strip (or plate) steel. As shown in Fig. 1, hot slabs in HRL are supplied by continuous casting (CC) whose task is casting liquid iron into hot slabs one by one. Two middle procedures, heat preservation pit and reheating furnace, are set between CC and HR in order to keep slabs’ temperature fit for hot rolling. The first one is to slow the cooling of hot slabs, the second one is to reheat the slabs so as to make their temperature always satisfy hot rolling temperature restriction. Longer waiting time means more energy needed to reheat the slabs to the ideal rolling temperature. Thus energy loss function is supposed as monotone increasing function dependent of the waiting time.

Nowadays, energy crisis has aroused the government and the enterprises’ concern in China. China is the second largest energy producer & consumer in the world, which occupies 22% of the total energy consumption of the whole world in 2002. Iron and Steel industry consumes above 10% of the national total energy consumption, which locates at a priority area in energy conservation. The national output of steel has occupied the first position in the world for 9 years since 1996, but energy consumption of Iron-and-Steel industry is 15%–20% higher than the developed country. Then it is necessary to reduce the energy consumption in order to improve the enterprise’s competitive power.

To reduce the energy loss in hot rolling line, a reasonable schedule of slabs with chain constraints should be deter-

* Corresponding author: E-mail: yangyang@ise.neu.edu.cn
DOI: http://dx.doi.org/10.2355/isijinternational.54.2837

Fig. 1. The flow between continuous casting and hot rolling in the Steel-making Complex.
The slabs have been formed to several chains in which the order of the slabs has been determined a priori. Then this hot slab scheduling problem can be denoted as \( P_{\text{chains}, r, \text{hot}}(f_{\text{slab}}(\text{wait}_{jk})) \), where \( \text{wait}_{jk} \) is the waiting time of slab \((j, k)\) before parallel hot rolling lines, ‘hot’ represents that the slabs arrive with high temperature and \( f_{\text{slab}}(\text{wait}_{jk}) \) is the energy loss cost that is dependent on the waiting time of slab \((j, k)\).

The rest of the paper is organized as follows. The review of the related works would be given in section 2. In section 3, we describe and model the hot slabs scheduling problem in parallel hot rolling production lines. The solution procedure for this problem will be presented in sections 4. Section 5 will report the experiment results. Brief conclusions will be given in section 6.

2. Review

Several optimization results about scheduling in iron and steel production have been achieved. Tang et al.\(^1\) focused on a charge batching problem in practice. Tang et al.\(^2\) classified the steel-making process problem as a hybrid flowshop problem with complicated technological constraints, and solved the model by Lagrangian relaxation. Tang et al.\(^3\) gave four different production processes from continuous casting line to hot rolling line. Generally, most of the slabs still should be reheated before being sent to hot rolling lines, nowadays. Lopez et al.\(^4\) focused on the production scheduling problem in the hot strip mill, and solved the selection and sequencing problem simultaneously by tabu search. Tang et al.\(^5\) set up a multiple traveling salesman problem model for hot rolling scheduling without considering time, and got a near-optimal solution by modified genetic algorithm. But they did not consider parallel production lines and chain constraints which differ from our work. Moreover, an exact algorithm is developed in our paper to obtain the optimal solution of the problem.

Previously, parallel machine scheduling problem involved various objective functions and constraints, including minimizing total weighted number of tardy jobs,\(^6\) minimizing the total weighted completion times,\(^6,7\) maximizing the weight of jobs which meet their respective deadlines,\(^8\) and minimizing total processing cost.\(^9\) However, the energy consumption is seldom considered, which implies our problem’s innovation.

Batch scheduling for parallel machine scheduling is active topic recently. However, chain constraints were seldom considered in parallel machine scheduling, which determine the order of the items in one chain and the makeup of one chain. Pinedo\(^10\) indicates that \( P_{2|\text{chains}}(\Sigma C_k) \) is strongly NP-hard, which is a special case of \( P_{\text{chains}, r, \text{hot}}(\Sigma f_{\text{slab}}(\text{wait}_{jk})) \). Therefore, \( P_{\text{chains}, r, \text{hot}}(\Sigma f_{\text{slab}}(\text{wait}_{jk})) \) is also strongly NP-hard, which considers the more complex objective function \( \Sigma f_{\text{slab}}(\text{wait}_{jk}) \) and the ready time.

To solve parallel machine scheduling problems, many algorithms have been tried, \textit{i.e.} dynamic programming,\(^11,12\) heuristic,\(^13\) column generation,\(^6,7,14\) state space relaxation.\(^15,16\) However, all the developed algorithms cannot be directly used to solve our problem, since the characteristics of the concerned problem in this paper differs from the previous works at the following several aspects. First of all, the candidate items in our problem are hot slabs with up to about 1200°C, which would incur more costs if they wait in the yard for longer time. Second, the slabs have been formed to several chains in which the slabs have appropriate features so that they can be processed successively without interruption. Third, energy loss cost is set as the objective function of scheduling. At last, parallel hot rolling lines are considered in this paper, and the previous work of scheduling in Iron & Steel Complex usually considers only one production line. To solve our problem optimally, branch and pricing algorithm based on column generation technology is developed in this paper.

3. Problem Description and Modeling

This problem sources from the hot rolling lines. To keep the rolling temperature no lower than a given lower limit, an energy loss cost is incurred by heating the slab. Slabs with similar features are scheduled priori as a chain in which the slabs’ order has been determined. This problem is described as a parallel machine scheduling problem with the objective of minimizing the total energy loss cost while satisfying the chain constraints. Each hot rolling line can roll at most one slab at one time and one slab can also be processed on at most one hot rolling line. It assumed that the time horizon is long enough to complete all the slabs.

\textit{Express of energy loss function}

The slab arrives at parallel hot rolling lines with high temperature. The large gap between the room temperature and the slab’s temperature leads to a fast increasing of the slab’s energy loss. The gap decreases as time passed, then the increasing of slab’s energy loss become slower. Based on the analysis, two kinds of energy loss functions are con-

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig2.png}
\caption{The sequence of total primary energy consumption (2002).}
\end{figure}
cerned: linear and nonlinear energy loss function, respectively. For the linear case, the energy loss of the \(k\)-th slab in chain \(j\) - \((j, k)\) is simplified as a linear function of its waiting time with assumed slope \(\beta_j > 0\). With respect to nonlinear case, the energy loss of slab \((j, k)\) is described as \(f_{jk}(\text{wait}_{jk}) = \alpha_j \times \text{wait}_{jk}^2 + \beta_j \times \text{wait}_{jk} + \gamma_j\), where \(\alpha_j < 0\), \(\beta_j > 0\), \(\gamma_j = 0\). These two supposed curves can reflect the trend of energy loss’s change in a certain extent.

**Notation**

- \(M\) the set of all hot rolling lines, \(M = \{1, 2, \ldots, m\}\), where \(m\) is the total number of hot rolling lines.
- \(\Omega\) the index set of all chains, \(\Omega = \{1, 2, \ldots, \kappa, \ldots, \text{nc}\}\) and the \(k\)-th chain is \(ch_k\), chain \(ch_k\) contains \(n_k\) slabs whose order has been determined a priori by chain constraint. The \(j\)-th slab in chain \(ch_k\) is slab \((j, k)\). And \(\text{nc} (= |\Omega|)\) is the total number of chains.
- \(\Omega_\nu\) the index set of all concerned chains in schedule \(s\).
- \(N\) the set of all slabs, there are totally \(n\) slabs in this set, then \(\sum_{i=1}^{n} n_i = n\).
- \(N_i\) the set of all concerned slabs in schedule \(s\).
- \(B_{jk}\) the set of all concerned slabs in schedule \(s\).
- \(p_{jk}\) the processing time of slab \((j, k)\).
- \(r_{jk}\) the ready time of slab \((j, k)\).
- \(\Gamma_j\) the set of all feasible acyclic schedules.
- \(\Gamma_u\) the set of the schedules in which slabs can be processed more than once.

**Decision variables**

\[
x_{ik} = \begin{cases} 
1, & \text{if the } i \text{-th chain is processed immediately before the } k \text{-th chain.} \\
0, & \text{else.}
\end{cases}
\]

\[
b_{jk} \quad \text{the starting time of slab } (j, k).
\]

\[
\text{wait}_{jk} \quad \text{the waiting time of slab } (j, k).
\]

\[
bc_k \quad \text{the start time of the first slab in } ch_k.
\]

Then the hot slab scheduling problem can be formulated as the following master problem and pricing problem. The problem can be solved through the natural interpretation of the master problem and pricing problem.

**Master problem:**

Define \(a_{ik}\) is the number of times that chain \(ch_k\) is processed in schedule \(s\) \((s \in \Gamma_f \cup \Gamma_u)\), \(f_s\) is the total energy loss cost of the slabs in the schedule \(s\). Define \(y_s\) equals 1 if schedule \(s\) \((s \in \Gamma_f \cup \Gamma_u)\) is used and 0 otherwise. The master problem (MP) is formulated as below:

\[
\begin{align*}
\text{min} & \quad \sum_{s \in \Gamma_f \cup \Gamma_u} f_s y_s & \quad \text{(1)} \\
\text{subject to} & \quad \sum_{s \in \Gamma_f \cap \Gamma_u} a_{ik} y_s = 1, \forall k \in \Omega_k & \quad \text{(2)} \\
& \quad \sum_{s \in \Gamma_f \cap \Gamma_u} y_s \leq m & \quad \text{(3)} \\
& \quad y_s \in [0, 1], \forall s \in \Gamma_f \cup \Gamma_u & \quad \text{(4)}
\end{align*}
\]

The master problem is completely equivalent to the original concerned problem if all the schedules are involved. Actually, not all the columns are concerned in the solution procedure, so the models in the solution procedure are the restricted version of master problem (RMP). The objective function (1) is minimizing the total energy loss costs of the slabs. Constraints (2) ensure that each slab is processed exactly once. Constraint (3) represents there are at most \(m\) available hot rolling production lines. The RMP is relaxed by relaxing the integer variables \(y_s\) to continuous variables in \([0, 1]\). The dual variables used in the pricing problem can be obtained by solving the relaxed version of RMP.

**Pricing problem:**

The pricing problem has the task of finding the better column with negative reduced cost. When no columns with negative reduced costs can be found, the optimal solution of the original problem has been obtained. Let \(\pi_k\) \((k \in \Omega)\) be the optimal dual solution corresponding to \(ch_k\) in constraints (2) and \(\bar{c}\) be the optimal dual solution corresponding to constraint (3). Then the pricing problem can be formulated as follow:

\[
\begin{align*}
\min & \quad f_s - \sum_{k \in \Omega} a_{ik} \pi_k - \alpha & \quad \text{(5)} \\
\text{subject to} & \quad x_{ik} = \sum_{l \in \Omega_k \cup \Omega_{s+1}} x_{l}, \forall k \in \Omega_k & \quad \text{(6)} \\
& \quad \sum_{(a, k, \lambda) \in B_{jk}} p_{a \lambda} + \sum_{(a, k, \lambda) \in B_{jk}} \left( b_{c \lambda} + \sum_{(j, k) \in \Gamma_k} p_{jk} \right) x_{a}, \forall (j, k) \in N_j & \quad \text{(7)} \\
& \quad \sum_{(j, k) \in \Gamma_k} p_{jk} + \sum_{(j, k) \in \Gamma_k} \left( b_{c \lambda} + \sum_{(j, k) \in \Gamma_k} p_{jk} \right) x_{a}, \forall k \in \Omega_k & \quad \text{(8)} \\
& \quad \text{wait}_{jk} = b_{jk} - r_{jk}, \forall (j, k) \in N_j & \quad \text{(9)} \\
& \quad b_{jk} \geq \text{wait}_{jk}, \forall (j, k) \in N_j & \quad \text{(10)} \\
& \quad f_s = \sum_{k \in \Omega} \left( a_{ik} \times \sum_{(j, k) \in \Gamma_k} \text{wait}_{jk} \right), \forall s \in \Gamma_f \cup \Gamma_u & \quad \text{(11)} \\
& \quad x_{ik} \in [0, 1], \forall i, k \in \Omega_k & \quad \text{(12)}
\end{align*}
\]
4. Solution Methodology

As it is mentioned before, $P_{|\text{chains}, r_{jk}, \text{hot}|} \Sigma f_{jk}(\text{wait}_{jk})$ is strongly NP-hard even for the linear case. To obtain the optimal solution of the problem, column generation based algorithm is adopted, which has got better results in solving parallel machine scheduling problem (Chen and Powell[10] and Vandenbergkler et al.[12]). In column generation procedure, the master problem and the pricing problem are solved iteratively. The linear relaxation of RMP is solved to provide dual variables to pricing problem, and the pricing problem is solved to find out the schedule which can further reduce the objective of the RMP. While the pricing problem cannot find the optimal solution of the scheduling problem that the slabs in the chains are scheduled last in the current sequence and finished at time $t$. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is computed by the following recursion (13).

$$F(S, t) = \min_{S, t} \left\{ \sum_{\text{chains}} P_{jk} + \sum_{\text{chains}} f_{jk} \left( t - \sum_{(a, k) \in S, \delta_k} p_{ak} - r_{jk} \right)^+ - \pi_j \right\},$$

where $\pi_j$ is determined by $\max_{\text{chains}} \left\{ \sum_{(a, k) \in S, \delta_k} P_{ak} \right\}$.

The number of the states for the above recursion is $O(T^2)$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is represented by the parameter $k$ of the $k$-th chain which is scheduled last in the current sequence and finished at time $t$. In the relaxed state space, there are $c n T$ states concerned, therefore, the solution of the pricing problem is accelerated. At the expense, some non-feasible paths of the pricing problem, in which one slab can be processed more than once can also be generated by solving the pricing problem. Thus the lower bound of the pricing problem is found instead of the optimal solution.

To tighten the lower bound, 2-cycles elimination is proposed, which can avoid one chain is processed twice consecutively. Let $F(k, t)$ be the minimum cost of the schedule that the $k$-th chain is scheduled last in the current sequence and finished at time $t$. As the objective function value is $O(T^{2n})$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is computed by the following recursion (13).

$$F(k, t) = \min_{k \in \Omega, t} \left\{ \sum_{\text{chains}} P_{jk} + \sum_{\text{chains}} f_{jk} \left( t - \sum_{(a, k) \in S, \delta_k} p_{ak} - r_{jk} \right)^+ - \pi_j \right\},$$

where $\pi_j$ is determined by $\max_{\text{chains}} \left\{ \sum_{(a, k) \in S, \delta_k} P_{ak} \right\}$.

The number of the states for the above recursion is $O(T^2)$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is represented by the parameter $k$ of the $k$-th chain which is scheduled last in the current sequence and finished at time $t$. In the relaxed state space, there are $c n T$ states concerned, therefore, the solution of the pricing problem is accelerated. At the expense, some non-feasible paths of the pricing problem, in which one slab can be processed more than once can also be generated by solving the pricing problem. Thus the lower bound of the pricing problem is found instead of the optimal solution.

To tighten the lower bound, 2-cycles elimination is proposed, which can avoid one chain is processed twice consecutively. Let $F(k, t)$ be the minimum cost of the schedule that the $k$-th chain is scheduled last in the current sequence and finished at time $t$. As the objective function value is $O(T^{2n})$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is computed by the following recursion (13).

$$F(k, t) = \min_{k \in \Omega, t} \left\{ \sum_{\text{chains}} P_{jk} + \sum_{\text{chains}} f_{jk} \left( t - \sum_{(a, k) \in S, \delta_k} p_{ak} - r_{jk} \right)^+ - \pi_j \right\},$$

where $\pi_j$ is determined by $\max_{\text{chains}} \left\{ \sum_{(a, k) \in S, \delta_k} P_{ak} \right\}$.

The number of the states for the above recursion is $O(T^2)$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is represented by the parameter $k$ of the $k$-th chain which is scheduled last in the current sequence and finished at time $t$. In the relaxed state space, there are $c n T$ states concerned, therefore, the solution of the pricing problem is accelerated. At the expense, some non-feasible paths of the pricing problem, in which one slab can be processed more than once can also be generated by solving the pricing problem. Thus the lower bound of the pricing problem is found instead of the optimal solution.

To tighten the lower bound, 2-cycles elimination is proposed, which can avoid one chain is processed twice consecutively. Let $F(k, t)$ be the minimum cost of the schedule that the $k$-th chain is scheduled last in the current sequence and finished at time $t$. As the objective function value is $O(T^{2n})$ where $T$ is the length of planning period. Thus the computation time of the recursion is exponential. To reduce the computation time, the state $(S, t)$ is computed by the following recursion (13).

$$F(k, t) = \min_{k \in \Omega, t} \left\{ \sum_{\text{chains}} P_{jk} + \sum_{\text{chains}} f_{jk} \left( t - \sum_{(a, k) \in S, \delta_k} p_{ak} - r_{jk} \right)^+ - \pi_j \right\},$$

where $\pi_j$ is determined by $\max_{\text{chains}} \left\{ \sum_{(a, k) \in S, \delta_k} P_{ak} \right\}$.
\[ P^{\text{max}} = \max_{g \in \Omega \setminus \{ k \}} \left\{ R_g \right\} - \rho \frac{i \cdot P_{\text{max}}}{m}, \]

\[ v = \left\lceil \frac{\max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\}}{P^{\text{max}}} \right\rceil. \]

To make the computation of \( D_k \) clearly, the definition of \( \delta_i \) is given as the number of chains whose earliest start time \( R_g \in [0, iP_{\text{max}}) \), for \( g \in \Omega \setminus \{ k \} \). Especially, \( \delta_0 \) involves the chains arriving at zero. Expression (17) is explained as follow with the assumption that the cost corresponding to chain \( c_k \) becomes larger as it completes later.

In the worst case, all the chains of slabs are processed before the \( k \)-th chain on one hot rolling line (as Fig. 4(a)), which means

\[ C_k \leq \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} + \sum_{i \in \Omega \setminus \{ k \}} P_i. \] .......................... (18)

In addition, an extreme case is considered in which the \( k \)-th chain is scheduled last in the optimal schedule. If the \( k \)-th chain is removed from the optimal schedule, a new schedule is obtained. Let the first completed production line in the new schedule is released at time point \( C_{\text{min}} \) (as Fig. 4(b)). Then it is obviously

\[ C_{\text{min}} \leq \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} + \left\lceil \frac{\sum_{i \in \Omega \setminus \{ k \}} P_i}{m} \right\rceil. \] .......................... (19)

in which all the slabs are processed after \( \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \).

Furthermore, there must be some slabs finished before \( \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \) in the optimal schedule, otherwise a better schedule can be obtained by arranging \( \sum_{i = 1}^{v} m \rho_{i} \) slabs belonging to set \( \left\{ R \in \Omega \setminus \{ k \} \mid R_{i} + P_{i} \leq \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \right\} \) to be finished before time point \( \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \) (as Fig. 4(c)).

The inequality can be tightened as

\[ C_{\text{min}} \leq \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} + \left\lceil \frac{\sum_{i \in \Omega \setminus \{ k \}} P_i}{m} \right\rceil - \sum_{i = 1}^{v} \rho_i P_{\text{min}}. \] .... (20)

in which \( \sum_{i = 1}^{v} \rho_i P_{\text{min}} \) is subtracted from the right item of (19) because at least \( \sum_{i = 1}^{v} m \rho_{i} \) slabs can be finished before \( \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \) instead of after it and these slabs’ processing time is no less than \( P_{\text{min}} \). Then the releasing of each machine can be preacted at least \( \sum_{i = 1}^{v} m \rho_{i} \times P_{\text{min}} \). Then the completion time of chain \( k \) should satisfy the following inequality,

\[ C_k \leq \max \left\{ R_{s_i}, \max_{i \in \Omega \setminus \{ k \}} \left\{ R_i \right\} \right\} + \left\lceil \frac{\sum_{i \in \Omega \setminus \{ k \}} P_i}{m} \right\rceil - \sum_{i = 1}^{v} \rho_i P_{\text{min}} \] + \left\lceil \frac{\sum_{i = 1}^{v} \rho_i P_{\text{min}}}{m} \right\rceil. \] .......................... (21)

Otherwise, a better result can be obtained by arranging the slabs in chain \( k \) processed at the last position of the first completed production line corresponding to \( C_{\text{min}} \).

4.2. Branching Strategy

If the optimal solution of the linear relaxation problem at root node does not satisfy the integer condition, branching is needed to find the optimal solution of the original problem. Here adopt the following branching strategy based on the most fractional relationship between two chains. For a node in the branching tree that has been explored, if the solution of linear relaxation problem is fractional, an appro-
prietate fractional relationship between two chains $i$ and $k$ is then selected to branch on next. Two child nodes are created. For the left child node, it restricts that chain $i$ cannot be processed immediately before chain $k$. For the right child node, it requires that chain $i$ must be processed immediately before chain $k$.

With respect to the case of linear energy loss function, a lemma is proposed to reduce the branching feasibilities. Because the energy loss function of slab $(j, k)$ is a linear function, which can be expressed as $f_{\text{disk}}(\text{wait}_{jk})=b_j^\prime \times \text{wait}_{jk} + \gamma_j^\prime$, so it has additivity property.

**Lemma 1:** In the optimal solution, chain $i$ can not be scheduled immediately before chain $k$, if $\sum_{(i,j) \in k} f_{\text{disk}}(P_i) - \sum_{(j,k) \in i} f_{\text{disk}}(P_j) < 0$.

Proof: The lemma is proven by contradiction. Suppose that chains $i$ is processed immediately before chain $k$ in the optimal solution. The objective function value can be written as

$$O = \sum_{g \in \Omega} \sum_{(j, g) \in c_h} f_{\text{disk}}(\text{wait}_{ji}) + \sum_{(j, k) \in i} f_{\text{disk}}(\text{wait}_{jk}).$$

The objective function value of the case that exchange chains $i$ and $k$ is written as

$$O' = \sum_{g \in \Omega} \sum_{(j, g) \in c_h} f_{\text{disk}}(\text{wait}_{ji}) + \sum_{(j, k) \in i} f_{\text{disk}}(\text{wait}_{jk}),$$

where $\text{wait}_{ji} = \text{wait}_{g}$ for $(j, g) \in c_h$; $g \in \Omega \setminus \{i, k\}$; $\text{wait}_{jk} - \text{wait}_{k} = \sum_{(a, i) \in k} p_{ai}$ for $(j, i) \in c_h$; $\text{wait}_{jk} = \sum_{(i, k) \in i} p_{ai}$ for $(j, k) \in c_h$.

Then $O' - O = \sum_{(j, i) \in k} f_{\text{disk}}(\sum_{(a, i) \in k} p_{ai}) - \sum_{(j, k) \in i} f_{\text{disk}}(\sum_{(i, k) \in i} p_{ai}) \geq 0$. It leads a contradiction, then the proof is completed.

For the problem whose energy loss function is linear, some feasible branching satisfying the condition of lemma 1 can be excluded because the optimal solution cannot exist in the child nodes generated after this kind branching.

### 4.3. Heuristic to Get The Initial Solution

Some initial columns must be generated to start the column generation procedure. The heuristic in Park and Kim\(^{14}\) is used to obtain the initial solution of parallel machine scheduling.

### 5. Computation Experiments

The above two algorithm are implemented by C language and tested on a Pentium IV with 3.0 GHz PC. The computation experience is reported to illustrate the performance of the the column generation algorithms. Linear programming problem inside the column generation algorithms are solved by IBM OSL, a commercial LP solver.

### 5.1. Experiment Design

The size parameters are given as follow.

- Number of slabs: $n \in \{60, 80, 100, 120\}$.
- Number of production lines: $m \in \{3, 5, 6, 8\}$.
- Number of chains: $cn \in \{20, 30, 40, 50, 60\}$.
- Processing time ($p_{jk}$) is an integer number uniformly drawn from $[1, 10]$. A schedule is constructed by selecting slabs $(j, k)$ at random and then scheduling them at the earliest possible time $t_i$. Ready times is set as $r_{jk} = t_i - U[1, 30]$, where random numbers are integer-valued. Adjustments are made to avoid negative ready times for the first slabs by increasing all schedule times $t_i$.

Parameters to determine energy loss function.

- linear case ($\beta_{jk}$). It is an integer number uniformly drawn from $[1, 1000]$.
- nonlinear case ($\alpha_{jk}, \beta_{jk}, \gamma_{jk}$). Firstly, $\alpha_{jk}$ is an integer number uniformly drawn from $[-10, -1]$. $\beta_{jk}$ is set as $-2\alpha_{jk}(D_{l}-r_{jk})$ in order to keep that the later the slab is processed, the larger the energy loss is. The determination of parameter $\beta_{jk}$ is based on the following analysis. The temperature of the slab $(j, k)$ cannot be less than room temperature based on heat exchange theory. Meanwhile the minimum reachable temperature of the slab $(j, k)$ at the parabola’s extreme point is assumed to correspond to the difference between $D_l$ and the ready time $r_{jk}$. $\gamma_{jk}$ must be zero because there is no energy loss when slab $(j, k)$ arrives just now.

### 5.2. Computational Results

The following two tables report the computational results for the problems with linear energy loss function and the problem with nonlinear energy loss function, respectively. The problem is determined by the parameters in the first three columns $n, m, cn$ where $n$ is the number of slabs, $m$ is the number of parallel hot rolling lines and $cn$ is the number of chains. The fourth column represents the average integrality gap, the integrality gap is equal to the ratio of the difference between the optimal solution and the lower bound to the lower bound, which is used to evaluate the performance of the lower bound. The fifth column represents the average computation time for solving these problems. The sixth column gives the number of the instances that is solved at root node. The last column represents the average number of the nodes in the solution procedure.

From Tables 1 and 2, the hot slab scheduling problems for both of the cases that the energy loss function is linear and nonlinear can be solved with up to 120 slabs and 60 chains in a reasonable time.

When the number of the chains is smaller, or the number of production lines is larger, the computation time is shorter. The computational time is insensitive with the number of the slabs. This probably can be explained by that the problem pays more attention to the scheduling of the chains, and the order of the slabs in one chain has been determined by the chain constraints, so the factors inflecting the algorithm complexity is in fact the number of the chains. This is to say that the algorithm is good at the slab scheduling problem with less varieties.

For the case whose energy loss function is linear, most instances can be solved at the root node from Table 1. The ratio of the instances that is solved at root node becomes less for the case whose energy loss function is nonlinear. The
small integrality gap in the experimental results illustrate that our proposed algorithm can achieve a tight lower bound, which can be obtained at the root node.

From Tables 1 and 2, the objective functions have week influences to the results of the algorithms except the average node number. This may be because a lemma is proposed in the linear case such that some nodes that cannot generate optimal solution are eliminated. For the other aspects, the energy loss function does not relate to the algorithm framework no matter for the linear case or for the nonlinear case. The energy loss function is just used to evaluate the schedule.

6. Conclusion

This paper studies the problem that determines the schedule of the hot slabs in parallel hot rolling production lines in which chain constraints are concerned, the objective function is minimizing the total energy loss costs. In this paper, two cases of the objective functions are concerned including the linear and nonlinear energy loss function. The problem can be classified to $P|\text{chains}, r_{jk}, \text{hot}|\sum w_{jk}$ Since it is strongly $NP$-hard, column generation combined with branch and bound is used to obtain the exact solutions of this problem. The experimental results illustrate that the proposed algorithm can optimally solve the medium scale instances with both linear and nonlinear energy loss function.

Acknowledgement

This research is partly supported by National Natural Science Foundation (Grant No. 71402020, 61374207), National 863 High-Tech Research and Development Program of China (2013AA040704).

REFERENCES

1) L. X. Tang, G. S. Wang and Z. L. Chen: Oper. Res., 62 (2014), 772.
2) L. X. Tang, P. B. Luh, J. Y. Liu and L. Fang: Int. J. Prod. Res., 40 (2002), 55.
3) L. X. Tang, J. Y. Liu, A. Y. Rong and Z. H. Yang: Eur. J. Oper. Res., 133 (2001), 1.
4) L. Lopez, M. W. Carter and M. Gentreau: Eur. J. Oper. Res., 106 (1998), 317.
5) L. X. Tang, J. Y. Liu, A. Y. Rong and Z. H. Yang: Eur. J. Oper. Res., 124 (2000), 267.
6) Z. L. Chen and W. B. Powell: Networks, 11 (1999), 78.
7) J. M. Vandenakker, J. A. Hoogeveen and S. L. Vandevelde: Oper. Res., 47 (1999), 862.
8) A. Bar-Noy, S. Guha, J. Naor and B. Schieber: SIAM J. Comput., 31 (2001), 331.
9) V. Jain and I. E. Grossmann: INFORMS J. Comput., 13 (2001), 258.
10) M. Pinedo: Scheduling: Theory, Algorithms, and Systems, 2nd ed., Tsinghua University Press, Beijing, (2005), 35.
11) T. C. E. Cheng, Z. L. Chen, M. Y. Kovalyov and B. M. T. Lin: IIE Trans., 28 (1996), 953.
12) S. Webster and M. A. Akgöz: Comput. Oper. Res., 28 (2001), 127.
13) M. W. Park and Y. D. Kim: Comput. Ind. Eng., 35 (1997), 793.
14) Z. L. Chen and W. Powell: Naval Res. Logist., 50 (2003), 823.
15) E. Choi and D. Tcha: Comput. Oper. Res., 34 (2007), 2080.
16) N. Christofides, A. Mingozzi and P. Toth: Networks, 11 (1981), 145.
17) T. S. Abdul-Razaq and C. N. Potts: J. Oper. Res. Soc., 39 (1988), 141.
18) T. Ibaraki and Y. Nakamura: Eur. J. Oper. Res., 76 (1994), 72.