INT. J. PROD. RES., 2002, VOL. 40, NO. 1, 55–70

Steel-making process scheduling using Lagrangian relaxation

LIXIN TANG†*, PETER B. LUH‡, JIYIN LIU§ and LEI FANG‡

The steel-making process, including steel-making and continuous casting, is usually the bottleneck in iron and steel production. Effective scheduling of this process is thus critical to improve productivity of the entire production system. Unlike the production scheduling in the machinery industry, steel-making process scheduling is characterized by the following features: job grouping and precedence constraints, set-up and removal times on the machines, and high job waiting costs. These features add extra difficulties to the scheduling problem. The objective is to ensure continuity of the production process and just-in-time delivery of final products. In this paper, a novel integer programming formulation with a 'separable' structure is constructed considering all the above-mentioned features. A solution methodology is developed combining Lagrangian relaxation, dynamic programming and heuristics. After relaxing two sets of 'coupling constraints', the relaxed problem is decomposed into smaller subproblems, each involving one job only. These subproblems are solved efficiently by using dynamic programming at the low level while the Lagrangian multipliers are iteratively updated at the high level by using a subgradient method. At the termination of such iterations, a two-stage heuristic is then used to adjust subproblem solutions to obtain a feasible schedule. A numerical experiment demonstrates that the method generates high quality schedules in a timely fashion.

1. Introduction

The metal-forming industry is an important link in the manufacturing chain, supplying extrusions, tubes, plates and sheets to many major manufacturing enterprises, including the automobile, aircraft, housing and food services, and beverage industries (Balakrishnan and Brown 1996). Iron and steel production includes several process phases (iron-making, steel-making — continuous casting and steel rolling), and is very extensive in investment and energy consumption. It is also characterized by high-temperature high-weight material flow with complicated technological processes. To accommodate customer requirements for different types of finished products with fluctuating demands, different rolling mills in the steel-rolling phase are designed with sufficient production capacity. Since the steel-making process (SP) phase needs expensive and energy-extensive equipment and runs in a continuous mode, its capacity is usually below the total capacity of the rolling stage.

Revision received April 2001.
† Department of Systems Engineering, Northeastern University, Shenyang, PR China.
‡ Department of Electrical and Computer Engineering, University of Connecticut, CT, USA.
§ Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Hong Kong, PR China.
* To whom correspondence should be addressed. Current address: Department of Operational Research, Faculty of Mathematical Studies, University of Southampton, Southampton SO17 1BJ, UK. e-mail: lt@maths.soton.ac.uk

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540110073000

Effective scheduling of SP resources is therefore vital, especially in today's highly competitive global steel market.

The steel-making process consists of three stages: steel-making, refining and continuous casting. Each stage further includes parallel machines, as shown in figure 1. The following is a brief description of the production process.

In the steel-making stage, carbon, sulphur, silicon, and other impurity contents of molten iron are reduced to desirable levels by burning with oxygen in a converter (CF) or electric arc furnace. The output from this stage is molten steel with the main alloy elements. The basic unit of steel-making production is a *charge*, which is defined as a 'job' in SP scheduling. It refers to the concurrent smelting in the same converter. The set-up and maintenance times at the steel-making stage are comparatively short. The steel in one charge may be cast into different slabs that are used to produce finished steel products for different customer orders. Charges must be so designed that (1) the orders in the same charge have identical steel grade and gauge; (2) the slab widths for these orders should be within certain limits; (3) delivery dates of the orders are close to each other; and (4) total slab weight ranges from 95% to 100% of the furnace capacity.

The molten steel from the steel-making stage is poured into ladles that are transported by a crane to a refining furnace (RF) for refining. The operation at this stage further refines the chemicals and eliminates impurities in the molten steel or adds the required alloy ingredients. If no RF is available when a new charge arrives, the charge has to wait until one of the RFs becomes available. The waiting time of a charge causes the charge temperature to drop, and reheating is needed. Energy consumption thus grows as the waiting time increases. The duration of the operation at the refining stage is usually similar to that of the operation at the steel-making stage. After refining, molten steel is poured into a tundish for casting.

In the casting stage, molten steel flows down from a hole at the bottom of the tundish into the crystallizer, the input unit, of a continuous caster. The molten steel continuously solidifies into slabs at the bottom of the caster. A sequence of charges that are consecutively cast on the same intermediate ladle and on the same continuous caster using the same crystallizer is called a cast, which is defined as a 'job group' in SP scheduling. Charges in the same cast need to satisfy the following technological constraints. (1) Steel grades for adjacent charges have to be identical or similar. (2) The slab gauge of different charges has to be identical. (3) The charges in the same cast must be sequenced so that their slab widths are in descending order. (4) The differences in slab widths of the charges in the same cast must be within a certain limit and the width jump between adjacent charges cannot exceed a given maximum value. (5) The total number of charges in a cast must be between a given lower bound and an upper bound that is determined by the life span of the intermediate ladle. (6) Delivery dates of different charges in the same cast should be as close as possible.

Set-up time on the caster is not needed between adjacent charges in the same cast. However, a relatively long set-up time is required between two casts on the same caster for changing the crystallizer. A removal time is also needed to clean equipment and tools, after a cast is finished. Both set-up and removal times are considered separate from the duration of operation because set-up and removal processes engage the equipment only, not the jobs.

As mentioned earlier, *charges* are 'jobs' and *casts* are 'job groups' in SP scheduling, and are defined at the lot planing level. After lot planning, the orders in each charge and their sequence are fixed. The charges in each cast and their sequence are also fixed. SP scheduling is then to decide the schedule of these jobs on the machines.

Planning and scheduling problems in iron and steel production have not drawn as much attention of the production and operations management researchers as many other industries, such as metal cutting and electronics manufacturing. A review of research on the integrated steel production planning and scheduling can be found in Tang *et al.* (2001).

Because steel-making scheduling with practical constraints is extremely complex, most approaches to SP scheduling treat the problem at three levels (Numao and Morishita 1991): (1) sub-scheduling, which fulfils the scheduling of individual charge sets; (2) rough scheduling, which merges sub-schedules; and (3) optimal scheduling, which eliminates machine conflicts. In our previous work (Tang *et al.* 2000), the sub-scheduling and rough scheduling were realized through human-computer interaction, a nonlinear mathematical model was built for the machine conflict problem, and the model was then converted into a linear programming model for easy solution. Kalagnanam *et al.* (2000) addressed the problem of satisfying customer orders in an order book with surplus slab inventory before scheduling the production for the remaining orders. The problem was formulated as a bicriteria multiple knapsack problem with additional constraints and was then solved by using a network-based heuristic. Cowling and Rezig (2000) developed a mixed integer programming model and a heuristic for integrated production planning of a steel continuous caster and hot strip mills. Significant savings from the implementation were reported. Chang *et al.* (2000) studied the lot planning problem to group charges into casts for a continuous slab caster in an integrated steel mill. An integer-programming model and an efficient heuristic were developed employing the column generation approach combined with a simple round-off scheme.

As can been seen from figure 1, the structure of the production system for the SP is similar to a hybrid flow shop (HFS). For many years, scheduling in HFS did not receive much attention, compared with that for the general flow shop and job shop environments. Gupta (1988) demonstrated that the two-stage HFS scheduling problem with the objective of minimizing the makespan is NP-complete. Vignier *et al.*

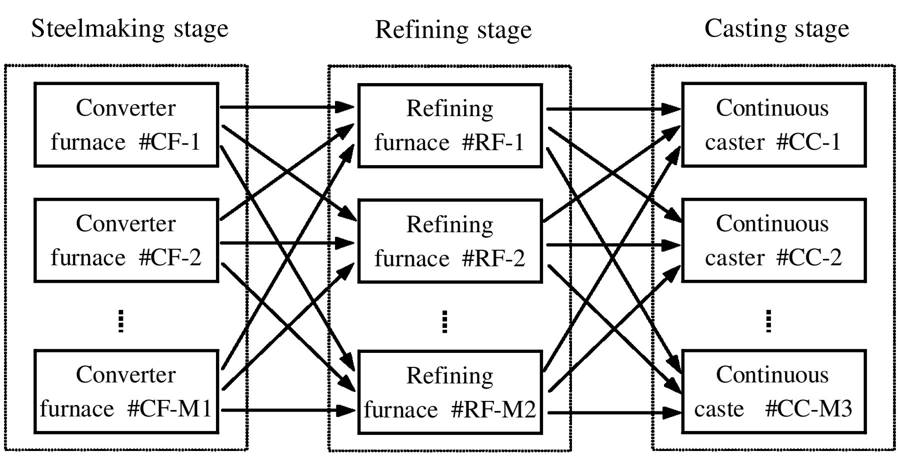

Figure 1. The steelmaking process.

(1999) gave a review on HFS scheduling problems. Most studies solve HFS scheduling problems by using the branch-and-bound method (e.g. Brah and Hunsucker 1991, Rajendran and Chaudhuri 1992, Portmann *et al*. 1998, Moursli and Pochet 2000) or heuristics (e.g. Rajendran and Chaudhuri 1992, Guinet and Solomon 1996, Gupta and Tunc 1998). Guinet and Solomon (1996) also gave a mixed integer programming formulation. However, these results cannot be used for SP scheduling for two main reasons: (1) SP has a number of additional practical constraints such as job groups and precedence that are not considered in HFS. (2) The scheduling criteria for SP are much more complex, involving waiting times and due dates, while HFS scheduling usually tries to minimize the makespan only.

In this paper, we develop an integrated model for the SP scheduling problem including all three SP stages, and present a Lagrangian relaxation method to solve it. The steel-making plant in Shanghai Baoshan Iron and Steel Complex (BaoSteel) is taken as the research background, and the goal is to generate daily schedules for SP and help BaoSteel planners to make better short-term (for a shift) scheduling decisions. The rest of the paper is organized as follows. Section 2 first summarizes the special technological features of the SP and outlines the requirements on the scheduling model. An integer programming formulation of the scheduling problem is then presented. Section 3 describes the solution methodology that combines Lagrangian relaxation and backward dynamic programming. Computational results on data abstracted from practical applications are reported in section 4.

2. Mathematical formulation of the problem

2.1. *The problem characteristics and modelling requirements*

After the charges and casts are defined by lot planning, the task of SP scheduling is to determine when and where (on which device) each charge should be processed at each production stage. The following general assumptions usually made in HFS scheduling are also viable for this problem.

(a) All charges follow the same process route: steel-making, refining, and then continuous casting. At each stage, a charge can be processed on any one of the machines at that stage, and the parallel machines at that stage are identical.
(b) A machine can process at most one job at a time.
(c) A job can be processed on at most one machine at any time.
(d) Job processing is non-pre-emptive.

However, it is clear from the introduction that the problem has some special features as compared with the scheduling in a general HFS. These features are summarized as follows.

(1) The number of charges to be scheduled within a shift is not large. However, each charge contains more than 100 tons of molten steel at high temperature. Transfer times between stages needs to be considered.
(2) Some specified charges (a cast) must be processed as a group on the same caster and there are precedence constraints among the charges within a group at the casting stage.
(3) Set-up and removal times are required, respectively, before and after a whole cast is processed on a caster. These times are separated from the processing times.

(4) Idle time on the caster between charges within a cast (cast break) is undesirable and involves cost.
(5) Waiting time of charges between the processing at different stages causes a temperature drop and results in cost for heating.
(6) Both earliness and tardiness on job completion lead to cost, e.g. for inventory or compensation to customers.

The above problem characteristics form the basis for developing a mathematical model. The modelling requirements are outlined below.

The objective is to ensure continuity of the production process and just-in-time delivery of final products through minimizing a cost function consisting of the following terms.

(1) Cast break loss penalties to discourage charges in the same cast being separated.
(2) Molten steel temperature drop cost due to job waiting between operations.
(3) Earliness/tardiness penalty used to ensure that slabs in each charge are delivered as punctually as possible.

The general assumptions and special characteristics presented above must be considered in the model to guarantee schedule feasibility.

(a) For the two consecutive operations for the same charge, only when the preceding operation has been finished can the immediate next one be started.
(b) For two consecutive charges processed on the same machine, only when the preceding charge has been finished can the immediate next one be started.

Set-up time is required from cast to cast on the same continuous caster. Set-up time must be considered before a new cast is to be processed.

Removal time is required for a caster after a cast has been finished on that caster.

2.2. *Notation*

To model the problem, the entire planning horizon (e.g. a shift) is divided into small time units, such that all the time parameters, for example processing, set-up, removal and transfer times, are of integer time units. The steel-making, refining and continuous casting stages are referred to as stages 1, 2 and 3, respectively. The model and solution methodology below can be easily modified for systems with more stages. The following symbols are used for defining the problem parameters and variables.

Parameters

- Ω set of all charges, $\Omega = \{1, 2, \ldots, N\}$, where N is the total number of production charges,
- Ω_g set of all charges in the gth cast, $g \in \{1, 2, \ldots, M\}$, where M is the total number of casts. $\Omega_h \cap \Omega_g = \emptyset$, for any h, $g \in \{1, \ldots, M\}$ and $h \neq g$. $\Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_M = \Omega$,
- s_{gp} pth charge in cast g, according to the charge sequence defined by lot planning, $p = 1, 2, \ldots, |\Omega_g|$. This notation simplifies the model presentation,
- d_i due date of charge i. It is a time point (the ending point of a time unit),
- $C1_g$ coefficient of cast break loss penalty for cast g,

$C2_{ij}$ coefficient of penalty cost for the waiting time of charge i after being finished at stage j,
$C3_i$ coefficient of penalty cost for the production of charge i being completed before its due date,
$C4_i$ coefficient of penalty cost for the production of charge i being late with respect to its due date,
T_{ij} processing time (duration) of charge i at stage j,
$t_{j,j+1}$ transportation time (duration) from stage j to stage $j+1$,
S_{ij} set-up time (duration) for charge i on a machine at stage j. When i is the first charge in a cast and $j = 3$, this is the required set-up time for the cast on a caster. For all other i, j, $S_{ij} = 0$,
R_{ij} removal time (duration) after processing charge i on a machine at stage j. When i is the last charge in a cast and $j = 3$, this is the required removal time of a caster after processing the cast. For all other i, j, $R_{ij} = 0$,
M_{jk} number of machines available at stage j in time unit k,
K total number of time units in the planning horizon.

Decision variables

$$\delta_{ijk} = \begin{cases} 1 & \text{if charge } i \text{ is processed at stage } j \text{ in time unit } k \\ 0 & \text{otherwise,} \end{cases}$$

$i \in \Omega; j = 1, 2, 3; k = 1, 2, \ldots, K,$

C_{ij} completion time of charge i on operation j, $i \in \Omega; j = 1, 2, 3$. It is a time point ('$C_{ij} = k$' means that the operation completes at the end of time unit k).

2.3. *The model*

Using the above symbols, the SP scheduling problem is formulated as follows. The objective function is to minimize the total cost due to cast breaking, job waiting and earliness/tardiness over the entire planning horizon, i.e.

$$\text{Minimize } Z, \text{ with } Z \equiv \sum_{g=1}^{M} \sum_{p=1}^{|\Omega_g|-1} C1_g (C_{s_{g,p+1},3} - T_{s_{g,p+1},3} - C_{s_{gp}3}) + \sum_{i=1}^{N} \sum_{j=1}^{2} C2_{ij} (C_{i,j+1} - T_{i,j+1} - t_{j,j+1}) + \sum_{i=1}^{N} C3_i \max(0, d_i - C_{i3}) + \sum_{i=1}^{N} C4_i \max(0, C_{i3} - d_i). \quad (1)$$

Subject to

$$C_{ij} + t_{j,j+1} \le C_{i,j+1} - T_{i,j+1}, \quad i \in \Omega; j = 1, 2. \quad (2)$$

$$C_{s_{gp}3} \le C_{s_{g,p+1},3} - T_{s_{g,p+1}3}, \; p = 1, 2, \ldots, |\Omega_g| - 1; \; g = 1, \ldots, M. \quad (3)$$

$$\sum_{k=1}^{K} \delta_{ijk} = T_{ij} + S_{ij} + R_{ij}, \quad i \in \Omega; \; j = 1, 2, 3. \quad (4)$$

$$k\delta_{ijk} \le C_{ij} + R_{ij}, \quad i \in \Omega;\ j = 1,2,3;\ k = 1,2,\ldots,K. \tag{5}$$

$$C_{ij} - T_{ij} - S_{ij} + 1 \le k + K(1 - \delta_{ijk}), \quad i \in \Omega;\ j = 1,2,3;\ k = 1,2,\ldots,K. \tag{6}$$

$$\sum_{i \in \Omega} \delta_{ijk} \le M_{jk},\ j = 1,2,3;\ k = 1,\ldots,K. \tag{7}$$

$$\delta_{ijk} \in \{0,1\}, \quad i \in \Omega;\ j = 1,2,3;\ k = 1,\ldots,K. \tag{8}$$

$$C_{ij} \in \{1,2,\ldots,K\}, \quad i \in \Omega;\ j = 1,2,3. \tag{9}$$

Each term in (1) represents one type of cost. Constraints (2) define operation precedence among the stages for a charge and ensure that a charge will not appear at more than one stage at the same time. Constraints (3) define the precedence in terms of (g,p) for charges within a cast to be processed on a caster. Constraints (4) impose the total time requirements, including set-up and removal if needed, for jobs on machines at each stage. Constraints (5) and (6) define the time interval for which a charge requires a machine at a stage. Together with constraint (4), they specify the requirement of a charge for a machine at each stage for the required continuous time period. Constraints (7) are machine capacity constraints, and (8) and (9) define the value range of the variables.

3. Solution methodology

Lagrangian relaxation (LR) is one of the efficient methods to solve large-scale integer programming problems (Fisher 1981). It introduces the 'coupling constraints' into the objective function by using a vector of Lagrangian multipliers to form a relaxed problem of the primal problem. For a given value of the vector of Lagrangian multipliers, the relaxed problem is usually much easier to solve than the original problem. The optimal values of the Lagrangian multipliers are searched through solving the Lagrangian dual problem by using a subgradient algorithm. The LR method has been successfully used in solving job shop scheduling problems in Hoitomt *et al.* (1993) and Luh *et al.* (1998). Based on the ideas of these two papers, the method presented in this section is utilized, as the problem involves a large number of integer variables and is difficult to solve directly.

3.1. *Lagrangian relaxation*

The problem formulation presented in section 2.3 shows that only constraints (3) and (7) couple different jobs. They are called the 'coupling constraints'. If they are relaxed, the problem can be decomposed to smaller subproblems, each involving only one job. Therefore, we form the following relaxed problem by introducing these two sets of constraints to the objective function through non-negative Lagrangian multipliers $\{u_i\}$ and $\{v_{jk}\}$, respectively:

(LR)

$$\text{Minimize } Z_{LR}, \text{ with } Z_{LR} \equiv \sum_{g=1}^{M} \sum_{p=1}^{|\Omega_g|-1} C1_g (C_{s_{g,p+1},3} - T_{s_{g,p+1},3} - C_{s_{gp}3})$$
$$+ \sum_{i=1}^{N} \sum_{j=1}^{2} C2_{ij} (C_{i,j+1} - T_{i,j+1} - C_{ij} - t_{j,j+1})$$
$$+ \sum_{i=1}^{N} C3_i \ \max(0, d_i - C_{i3}) + \sum_{i=1}^{N} C4_i \ \max(0, C_{i3} - d_i)$$
$$+ \sum_{g=1}^{M} \sum_{p=1}^{|\Omega_g|-1} u_{s_{gp}} (C_{s_{gp}3} - C_{s_{g,p+1},3} + T_{s_{g,p+1},3})$$
$$+ \sum_{k=1}^{K} \sum_{j=1}^{3} v_{jk} \left(\sum_{i \in \Omega} \delta_{ijk} - M_{jk} \right) \tag{10}$$

subject to constraints (2), (4), (5), (6), (8), (9), and

$$u_{s_{gp}} \geq 0, \ p = 1, 2, \ldots, |\Omega_g| - 1; g = 1, \ldots, M. \tag{11}$$

$$v_{jk} \geq 0, \ j = 1, 2, 3; \ k = 1, \ldots, K. \tag{12}$$

For given values of $\{u_i\}$ and $\{v_{jk}\}$, the relaxed problem can be decomposed into subproblems, each for one job. The subproblem for job i, $i \in \Omega$, is given as follows:

($\mathrm{LR_i}$)

$$\text{Minimize } Z_{LR}(i), \text{ with } Z_{LR}(i) \equiv \sum_{j=1}^{2} C2_{ij} (C_{i,j+1} - T_{i,j+1} - C_{ij} - t_{j,j+1})$$
$$+ C3_i \ \max(0, d_i - C_{i3}) + C4_i \ \max(0, C_{i3} - d_i)$$
$$+ \sum_{k=1}^{K} \sum_{j=1}^{3} v_{jk} \delta_{ijk} + \varphi(i) \tag{13}$$

subject to constrains (2), (4), (5), (6), (8), (9), where i is fixed for the ith subproblem and $\varphi(i)$ is defined, depending on the value of i, as shown below:

$$\varphi(i) \equiv (u_{s_{gp}} - C1_g) C_{s_{gp}3}, \text{ for } s_{gp} = i \text{ and } p = 1. \tag{14}$$

$$\varphi(i) \equiv (u_{s_{gp}} - C1_g) C_{s_{gp}3} + (C1_g - u_{s_{gp-1}})(C_{s_{gp}3} - T_{s_{gp}3}),$$
$$\text{for } s_{gp} = i \text{ and } p = 2, \ldots, |\Omega_g| - 1. \tag{15}$$

$$\varphi(i) \equiv (C1_g - u_{s_{g,p-1}})(C_{s_{gp}3} - T_{s_{gp}3}), \text{ for } s_{gp} = i \text{ and } p = |\Omega_g|. \tag{16}$$

In formulas (14) to (16), g and p correspond to a particular charge, i.e. charge i. Cast g is the cast that charge i belongs to, and p is the position of i in the cast, i.e. $s_{gp} = i$. Recall that i is the charge index in the charge set Ω, cast Ω_g is a grouping of charges with precedence constraints on them, and s_{gp} is the pth charge in Ω_g. Since $\Omega_j \cap \Omega_g = \varnothing$ for h, $g \in \{1, \ldots, M\}$ and $h \neq g$, and $\Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_M = \Omega$, there is a one-to-one correspondence between i and s_{gp}. For example, if there are ten charges in the problem, $\Omega = \{1, 2, \ldots, 10\}$, and the charges belong to three casts, $\Omega_1 = \{2, 3, 7\}$, $\Omega_2 = \{1, 4, 5, 10\}$, and $\Omega_3 = \{6, 8, 9\}$, then the unique relationship between i and s_{gp} can be represented in the form below:

$i = s_{gp}$	1	2	3	4	5	6	7	8	9	10
g	2	1	1	2	2	3	1	3	3	2
p	1	1	2	2	3	1	3	2	3	4

From such a form, the two notation systems can be easily converted to each other in the algorithm implementation. Variables using them as subscripts can be determined accordingly, e.g. for $g = 2$ and $p = 4$, it is clear that $i = s_{gp} = 10$ and $C_{s_{gp}3} = C_{i3} = C_{10,3}$.

3.2. *Dynamic programming for subproblems (LRi)*

The backward dynamic programming (DP) is used to solve the subproblems. The DP stages correspond to SP production stages, and the states at each stage correspond to possible job (charge) start times at that stage. Figure 2 shows a schematic of the backward DP. A line connecting two consecutive stages indicates a backward sweep route. For every node at Stage 1 or 2, there exist at most K lines connecting to K nodes at the immediate subsequent stage, and the connection is subject to operation precedence constraints (2). A solid line indicates the partial shortest path from a current node to the immediate subsequent stage.

The DP procedure starts from the last stage and proceeds to the first stage, moving in the direction opposite to the process flow direction. The cost for node i at the last (third) stage is given by:

$$V_{i3}(C_{i3}, \delta_{i3k}) = C2_{i2}(C_{i3} - T_{i3}) + C3_i \ \max(0, d_i - C_{i3}) + C4_i \ \max(0, C_{i3} - d_i) + \sum_{k=1}^{K} v_{3k}\delta_{i3k} + \varphi(i), \tag{17}$$

where $\varphi(i)$ is defined in (14) to (16).

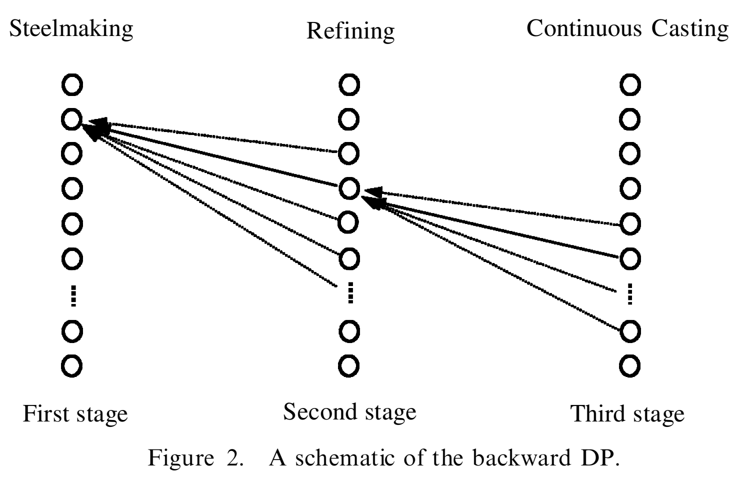

Figure 2. A schematic of the backward DP.

The cumulative cost for the second stage is given by

$$V_{i2}(C_{i2}, \delta_{i2k}) = C2_{i1}(C_{i2} - T_{i2}) - C2_{i2}(C_{i2} + t_{2,3}) + \sum_{k=1}^{K} v_{2k}\delta_{i2k} + \min_{\{C_{i3}, \delta_{i3k}\}} \{V_{i3}(C_{i3}, \delta_{i3k})\}. \quad (18)$$

The cumulative cost (total cost) for a node at the first stage is then

$$V_{i1}(C_{i1}, \delta_{i1k}) = -C2_{i1}(C_{i1} + t_{1,2}) + \sum_{k=1}^{K} v_{1k}\delta_{i1k} + \min_{\{C_{i2}, \delta_{i2k}\}} \{V_{i2}(C_{i2}, \delta_{i2k})\}. \quad (19)$$

The optimal subproblem cost is obtained as the minimal cumulative cost at the first stage. Finally, an optimal solution to subproblem (LR_i) can be obtained by tracing forwards the stages. The computational complexity of this DP procedure is $O(K^2)$.

3.3. *Obtain a feasible solution to the original problem*

A solution to the relaxed problem is generally infeasible for the original problem because the precedence constraints (3) and capacity constraints (7) have been relaxed. A two-phase heuristic has been developed to construct a feasible solution based on a solution to the relaxed problem. In the first phase, the solution is adjusted to ensure that the precedence constraints between jobs (charges) at the last stage are satisfied without considering machine capacity constraints. In the second phase, machine conflicts are resolved to produce a feasible solution to the original problem.

Let the solution of the relaxed problem at the nth iteration be denoted as $S^n = \{C^n_{ij},\ i \in \Omega;\ j = 1, 2, 3\}$. The starting times of operations can be expressed as $B^n_{ij} \equiv C^n_{ij} - T_{ij},\ i \in \Omega;\ j = 1, 2, 3$. Note that '$B_{ij} = k$' means that time unit $(k+1)$ is the first time unit for operation (i, j). The first phase of the heuristic algorithm is as follows.

Step 1. Set $g = 1$.
Step 2. For cast g, check the feasibility of job precedence constraints for the casting operation (3). If the constraints are satisfied, go to step 5.
Step 3. Let $B^n_{s_{g1}3} = \min_{j \in \Omega_g}\{B^n_{i3}\};\ B^n_{s_{gp}3} = B^n_{s_{gp-1}3} + T_{s_{gp-1}3},\ p = 2, \ldots, |\Omega_g|$. This step first finds the earliest starting time of the charge in cast g in the relaxed solution, lets this time be the staring time of the first charge in the cast, and then determines the starting times of other charges in the cast according to their defined precedence. In this way the starting time of the cast for the casting operation is not changed but any violation of precedence constraint is removed.
Step 4. Based on the starting time for the last operation of a charge, $B^n_{i3},\ i \in \Omega_g$, obtained in step 3, determine in a backwards fashion the starting times of the second and then the first operations of the charge, B^n_{i2} and B^n_{i1}, based on the required processing times, without considering capacity constraints.
Step 5. $g = g + 1$; if $g \leq M$, go to step 2; otherwise, proceed to the second phase.

Let $[i]$ denote the ith position in the sequence, and π_j the set of unscheduled jobs at stage j. The second phase of the heuristic is as follows.

Step 1. Set $j = 1,\ M1_{kj} = M_{jk}$ for $k = 1, \ldots, K$.

Step 2. Index the charges according their starting times, and then schedule them on stage j sequentially: for i from 1 to N, do the following sequentially. Let $B^n_{[i]j} = \min_{i1 \in \pi_j}\{B^n_{i1,j}\}$; find $k = \min\{k' | M1_{k'j} \neq 0$ and $k' \geq B^n_{[i]j}\}$; schedule charge $[i]$ to begin processing at stage j immediately after time point k, i.e. $B^n_{[i]j} = k$, set $M1_{k1,j} = M1_{k1,j} - 1$ for $k1 = k+1, \ldots, k + T_{[i],j}$; $\pi_j = \pi_j - \{[i]\}$.

Step 3. Update the charge starting times for the intermediate stage: let $j = j+1$; if $j < 3$, recalculate $B^n_{ij} = \max\{B^n_{i,j-1} + T_{i,j-1} + t_{j-1,j}, B^n_{ij}\}$ for $i \in \Omega$, go to step 2.

Step 4. Update the starting times of the first charges in all casts for the last stage such that the idle time on the caster between charges within a cast is as small as possible: Let $B1_{ij} = \max\{B^n_{i,j-1} + T_{i,j-1} + t_{j-1,j}, B^n_{ij}\}$ for $i \in \Omega$, and $B_{s_{g1}j} = B_{s_{g1}j} + \sum_{i1 \in \Omega_g} \max\{B1_{i1,j} - B^n_{i1,j}, 0\}, g = 1, \ldots, M$

Step 5. Index the casts according their starting times, and then sequentially schedule the casts for the last stage: for g, from 1 to M, do the following sequentially. Let $B_{s_{[g]1},j} = \min_{s_{g'1} \in \pi_j}\{S_{s_{g'1},j} + B_{s_{g'1},j}\}$; find $k = \min\{k' | M1_{k'j} \neq 0$ and $k' \geq B_{s_{[g]1},j}\}$; schedule the first charge in cast $[g]$ to start processing at stage j (the current casting stage) immediately after time point k and schedule the starting of other charges in the same cast at this stage accordingly, i.e. $B_{s_{[g]1},j} = k$, $B_{s_{[g]p},j} = B_{s_{[g]p-1},j} + T_{s_{[g]p-1},j}$, $p = 2, \ldots, |\Omega_{[g]}|$; set $M1_{k1,j} = M1_{k1,j} - 1$ for $k1 = k+1, \ldots, k + S_{[g]1} + \sum_{i \in \Omega_{[g]}} T_{ij} + R_{[g]|\Omega_{[g]}|}$; $\pi_j = \pi_j - \Omega_{[g]}$.

Step 6. Stop. $\{B^n_{ij}\}$ obtained above defines a feasible schedule.

3.4. *Updating Lagrangian multipliers*

The optimal values of the Lagrangian multipliers $\{u_i, v_{jk}\}$ are searched through by solving the Lagrangian dual problem:

(LD)

$$\text{Maximise } Z_D(u_i, v_{jk}), \text{ with } Z_D(u_i, v_{jk}) \equiv \min\ Z_{LR}$$

Subject to

$$u_{s_{gp}} \geq 0,\ p = 1, 2, \ldots, |\Omega_g| - 1;\ g = 1, \ldots, M,$$

$$v_{jk} \geq 0,\ j = 1, 2, 3;\ k = 1, \ldots, K.$$

A subgradient algorithm is applied to solve this dual problem. Lagrangian multipliers are updated at each iteration according to the feasible solution obtained from modifying the relaxed problem solution of the previous iteration. For a minimization problem, the solution to the Lagrangian dual problem provides a lower bound to the optimal primal cost and the feasible solution provides an upper bound. The procedure for solving the relaxed problem is as follows. Here tn is the step size for updating the multipliers and t^n is a factor for adjusting the step size at the nth iteration.

Step 1. Initialization: $n = 0$; $\alpha_0 = 2$; $Z^U = +\infty$; $Z^L = -\infty$; $u^0_{s_{gp}} = 0$; $p = 1, \ldots, |\Omega_g| - 1$, $g = 1, \ldots, M$; $v^0_{ij} = 0$, $j = 1, 2, 3$, $k = 1, \ldots, K$.

Step 2. Solve the relaxed problem by decomposition and DP as described in section 3.2. If the optimal objective value $Z_{D(u^n, v^n)} > Z^L$, then $Z^L = Z_{D(u^n, v^n)}$.

Step 3. Based on the solution of step 2, construct a feasible solution to the original problem P by using the heuristics presented in section 3.3. If the objective value $Z^n < Z^U$, then $Z^U = Z^n$.

Step 4. If one of the following conditions is satisfied, then stop; otherwise go to step 5.
(1) $(Z^U - Z^L)/Z^U < \delta$, where $\delta > 0$ is a very small number;
(2) Step size parameter $\alpha_n \leq 0.0005$;
(3) Number of iterations $n >$ the maximum iteration number chosen by the user;
(4) $||\gamma^n|| < \epsilon$, where γ^n is the subgradient at iteration n and $\varepsilon > 0$ is a very small number.

Step 5. Lagrange multipliers are updated as follows:

$$\gamma^n(u^n_{s_{gp}}) = (C^n_{s_{gp}3} - C^n_{s_{gp+1}3} + T^n_{s_{gp+1}3}),\ p = 1, \ldots, |\Omega_g| - 1,\ g = 1, \ldots, M;$$

$$\gamma^n(v^n_{jk}) = \sum_{i \in \Omega} \delta_{ijk} - M_{jk},\ j = 1, 2, 3,\ k = 1, \ldots, K;$$

$$t^n = \alpha_n(Z^U - Z_D(u^n, v^n))/\|\gamma^n(u^n, v^n)\|^2;$$

$$u^{n+1}_{s_{gp}} = \text{Max}\ \{0, u^n_{s_{gp}} + t^n\gamma^n(u^n_{s_{gp}})\},\ p = 1, \ldots, |\Omega_g| - 1,\ g = 1, \ldots, M;$$

$$v^{n-1}_{jk} = \text{Max}\ \{0, v^n_{jk} + t^n\gamma^n(v^n_{jk})\},\ j = 1, 2, 3,\ k = 1, \ldots, K;$$

$$\rho_n = 0.15^*[n/20];\ \alpha_{n+1} = \alpha^*_n\ \exp(-0.5\rho^2_n);$$

$$n = n + 1;\ \text{go to step 2.}$$

4. Computational experience

To test the performance of the method and to study the characteristics of the solutions, a computational experiment has been carried out on randomly generated problem instances, which were designed to reflect practical situations in iron and steel industries.

4.1. *Generation of problem instances*

To generate representative problem instances, we examined the actual production data from Baosteel Complex. Baosteel Complex is the largest and most advanced iron and steel enterprise in China. Its annual production is over 18 million tons of steel, and its auto sheet accounts for more than 60% of the domestic market share. In the steel-making plant of Baosteel Complex, every workday is divided into three shifts. Planners need to carry out SP scheduling in every shift for the next shift. The SP schedule generally includes about 5–7 casts for each workday, and a cast consists of 3-5 charges subject to technological constraints. The maximum number of charges to be scheduled in each workday is about 35 (about 12 per shift). Based on the above, the number of charges to be scheduled is set to be 12 for each instance. Since the minimum scheduling time unit in the iron and steel plant is in an exact number of minutes, the minute is taken as the basic time unit. The planning horizon is set to be 480 minutes, as this study intends to solve the SP scheduling problem for an eight-hour shift. Two other parameters are chosen to represent the problem structure as described below:

(1) The number of casts is set to vary at three levels: 3, 4, and 6.
(2) The number of machines at each stage is set to vary at three levels: 3, 4, and 5.

In order to reduce experimental cases, it is assumed that every stage has the same number of machines. However, our method presented here can deal with practical problems including different numbers of machines for different stages.

The combination of parameter levels gives nine problem scenarios, and for each scenario, ten different problem instances were randomly generated. Thus, a total of 90 problem instances were used in the experiment. According to practical data of the steel-making plant of Baosteel Complex, the processing times were randomly generated from a uniform distribution [30, 50]. The penalty coefficients for the cast break were set to be 500 while the penalty coefficients for the waiting time were randomly generated from a uniform distribution [100, 110]. Penalty coefficients for the earliness and tardiness were randomly generated from uniform distributions [10, 15] and [100, 120], respectively.

4.2. *Computational results*

The method was implemented by using Visual C++, and the experiment was carried out on a Pentium-II 400 MHz PC. Because Lagrangian relaxation cannot guarantee optimal solutions, the relative dual gap $(Z^{UB} - Z^{LB})/Z^{UB}$ is used as the measure of solution optimality, where Z^{UB} is the upper bound to the original problem and Z^{LB} is the lower bound. The optimality performance and running times of our Lagrangian relaxation method against different problem structures and against different problem sizes are presented in table 1. Figure 3 shows the evolution of the relative duality gap in the solution process for problems with different casts (the number of machines at each stage is fixed to 4), and figure 4 shows the duality gap evolution for problems with different numbers of machines (the number of casts is fixed to 3).

From the results presented in table 1, figures 3 and 4, the following observations can be made about our SP scheduling method.

Problem no.	Problem structures Casts × Machines	Optimality performances (%) Worst	Best	Average	Running time (s)
1	3 × 3	4.68	0.42	2.73	148.60
2	4 × 3	4.82	1.74	3.02	188.40
3	6 × 3	18.41	15.07	17.10	221.20
4	3 × 4	1.81	0.68	1.19	118.20
5	4 × 4	5.45	0.50	2.30	134.80
6	6 × 4	11.39	9.61	10.20	183.20
7	3 × 5	0.50	0.46	0.49	23.60
8	4 × 5	1.83	0.60	0.87	105.70
9	6 × 5	12.20	7.14	9.89	145.40

Note: Initial step size = 0.5, maximum number of iteration = 500.

Table 1. Optimality performances and running times of method.

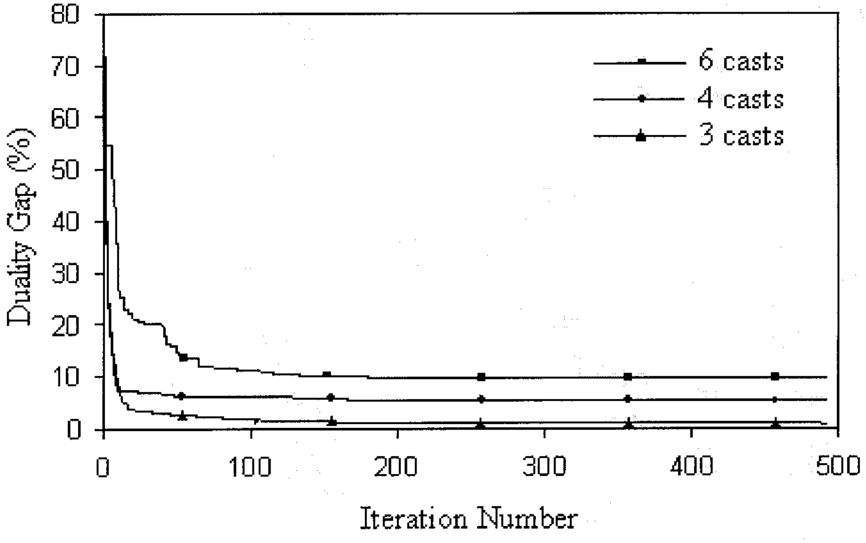

Figure 3. Duality gap for different number of casts.

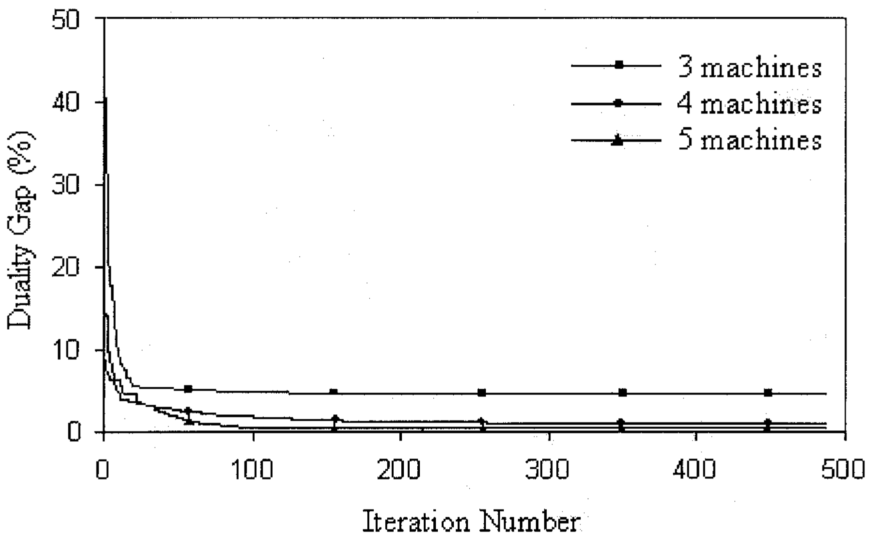

Figure 4. Duality gap for different number of machines.

(1) The overall average optimality performance is about 5.32% for 12 charges in a shift. Since this performance is measured using the relative duality gap, the actual relative distance to optimal solutions may be even smaller.
(2) As the number of machines increases, the optimality performance improves and the computation time decreases. This is consistent with the intuition that for a fixed number of charges, when the number of machines is larger, the resource is in less demand and the problem becomes easier to solve.
(3) The duality gap and the computational time increase as the number of casts increases. This is because, when the number of casts increases, there are

more precedence constraints on the charges. As a result, the problem size increases and the problem becomes more difficult to solve.

5. Conclusions

In this paper, the SP scheduling problem was viewed as a hybrid flowshop problem with complicated technological constraints, and was formulated as an integer-programming problem. The objective was to meet the requirements of just-in-time delivery and production operation continuity while considering all practical features, such as job set-up and removal times, job grouping and precedence constraints. A solution methodology that combined Lagrangian relaxation, dynamic programming, and heuristics was developed. A computational experiment on randomly generated realistic problems showed that the solution method is effective and efficient. The average relative duality gap is 5.32% and the average computational time is about 140 seconds on a Pentium-II 400-MHz PC. The method can be modified and extended to other hybrid flowshop scheduling problems where some technological constraints are removed or modified. It may also be extended to hot rolling scheduling problems with specific modifications on the implementation.

Acknowledgements

This research was supported in part by American National Science Foundation grants under Professor Peter Luh (Grant No. DMI-9813176). This research is also partly supported by National Natural Science Foundation of China (Grant No. 79700006), the National 863/CIMS Research Scheme of China (Grant No. 863-511-708-009), the Excellent Young Faculty Program and the Key Faculty Program of Ministry of Education, China. The authors would like to thank Professor Zihou Yang of Northeastern University, Dr Haoxun Chen, Mr Ming Fei and Professor L. S. Thakur of the University of Connecticut for providing insightful discussions and valuable help in algorithm development.

References

Balakrishnan, A. and Brown, S., 1996, Process planning for aluminium tubes: an engineering-operations perspective. *Operations Research*, **44**, 7–20.

Brah, S. A. and Hunsucker, J. L., 1991, Brach and bound algorithm for the flowshop with multiple processors. *European Journal of Operational Research*, **15**, 88–89.

Cowling, P. and Rezig, W., 2000, Integration of continuous caster and hot strip mill planning for steel production. *Journal of Scheduling*, **3**, 185–208.

Chang, S. Y., Chang, M. R. and Hong, Y. S., 2000, A lot grouping algorithm for a continuous slab caster in an integrated steel mill. *Production Planning & Control*, **11**, 363–368.

Fisher, M. L., 1981, Lagrangian relaxation method for solving integer programming. *Management Science*, **27**, 221–218.

Guinet, A. G. P. and Solomon, M. M., 1996, Scheduling hybrid flow shops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, **34**, 1643–1654.

Gupta, J. N. D., 1988, Two-stage hybrid flowshop scheduling problem. *Journal of Operational Research Society*, **39**, 359–364.

Gupta, J. N. D. and Tunc, E. A., 1998, Minimizing tardy jobs in a two-stage hybrid flowshop. *International Journal of Production Research*, **36**, 2397–2417

Hoitomt, D. J., Luh, P. B., Max, E. and Patipati, K., 1993, A practical approach to job-shop scheduling. *IEEE Transactions on Robotics and Automation*, **9**, 1–13.

Moursli, Y. and Pochet, Y., 2000, A branch-and-bound algorithm for the hybrid flowshop. *International Journal of Production Economics*, **64**, 113–125.

Kalagnanam, J. R., Dawande, M. W., Trumbo, M. and Lee, H. S., 2000, Surplus inventory matching problem in the process industry. *Operations Research*, **48**, 505–516.

Luh, P. B., Gou, L., Zhang, Y., Nagahora, T., Tsuji, M., Yoneda, K., Hasegawa, T., Kyoya, Y. and Kano, T., 1998, Job shop scheduling with group-dependent setups, finite buffers and long time horizon. *Annals of Operations Research*, **76**, 233–259.

Numao, M. and Morishita, S. I., 1991, Cooperative scheduling and its application to steel-making processes. *IEEE Transactions on Industrial Electronics*, **38**, 150–155.

Portmann, M. C., Vignier, A., Dardilhac, D. and Dezalay, D., 1998, Branch and bound crossed with GA to solve hybrid flow shops. *European Journal of Operational Research*, **107**, 389–400.

Rajendran, C. and Chaudhuri, D., 1992, A multi-stage parallel-processor flowshop problem with minimum flowtime. *European Journal of Operational Research*, **57**, 111–122.

Tang, L. X., Liu, J. Y., Rong, A. Y. and Yang, Z. H., 2000, A mathematical programming model for scheduling steelmaking-continuous casting production scheduling. *European Journal of Operational Research*, **120**, 423–435.

Tang, L. X., Liu, J. Y., Rong, A. Y. and Yang, Z. H., 2001, A review of integrated planning & scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, **133**, 1–20.

Vignier, A., Billaut, J. C. and Proust, C., 1999, Hybrid flowshop scheduling problems: State of the art. *Rairo-Recherche Operationnelle-Operations Research*, **33**, 117–183.