The Charge Batching Planning Problem in Steelmaking Process Using Lagrangian Relaxation Algorithm

Lixin Tang* and Shujun Jiang

Liaoning Key Laboratory of Manufacturing System and Logistics, The Logistics Institute, Northeastern University, Shenyang, China

In this paper, we investigate the charge batching planning problem (CBP) arising from practical steelmaking production. The CBP transforms the primary order requirements into various production batches (charges) subject to the steelmaking processing constraints and composite batch conditions according to the similarity in steelgrade, dimension, physical property, and due-date of orders. On the basis of a practical steelmaking process, a novel mixed-integer programming model for the CBP is presented by considering above constraints and features here, and two kinds of Lagrangian relaxation (LR) methods are proposed to solve the CBP by using different relaxation methods. In the first LR method, the relaxed problem is separated into subproblems by relaxing assignment constraints which are solved optimally by dynamic programming. In the second method, variable splitting is presented by introducing identical copies of some subsets of the original variables. To guarantee the equivalence to the primal problem, a number of equality coupling constraints are added into the model which are relaxed during the course of the second Lagangian relaxation. The multipliers in all above LR methods are then iteratively updated along subgradient directions. Computational experiments have been carried out and the results show that both LR methods can produce satisfactory average duality gaps and the second LR method is little better than the first method.

1. Introduction

The iron and steel industry has played an important role in the global market economy during the past decade. Along with rapid development, the iron and steel industry also faces fierce competition. To enhance their competitive power, many iron and steel corporations have changed production mode by transforming large lot production into small lot, with multiple varieties for satisfying their customer's diverse requirements. Since most of production equipment in the iron and steel industry is very large, it often operates in batch mode to save resources and energy consumption, but there is significant conflict between large batch mode and the customer's diverse requirements. Batching planning groups customer requirements into batches in order to resolve this contradiction and to improve the utilization of large production equipment.

This paper investigates a charge batching planning problem arising from practical steelmaking production operation management. A simplified steelmaking production process is illustrated in Figure 1. The steel production flow begins with iron making in the blast furnace. Iron ore is converted to pig iron which is transported by torpedo car to the steelmaking mill. Then pig iron is transformed to liquid steel in the converter. At last the melted steel is solidified into slab in continuous casters. In the steelmaking stage, a charge that is a basic production unit for steelmaking production refers to concurrent smelting in the same converter (or electric arc furnace). In the continuous casting stage, the charges are grouped into different casts. Each cast consists of several charges with similar steelgrades that are processed consecutively on the same continuous caster using the same crystallizer.

Reasonable design of charges and casts can improve productivity and cut down resource and energy consumption. The design of charges or casts is also regarded as charge batching planning or cast batching planning, respectively. The charge batching planning (CBP) is a key element of the production operation management in the process industry. It is converting the primary order requirements into various production batches (charges) subject to the steelmaking processing constraints. Figure 2 illustrates the process of making a charge batching plan. The rectangles in the left column in different colors denote steel orders with different steelgrade and specification. The orders with similar steelgrade and specification are denoted by close colors. As shown in Figure 2, orders in similar gray color are put into the first charge and orders in similar black color are put into the second charge.

The batching decision problem has recently received more attention from researchers. Trautmann and Schwindt[2] proposed a resource-constrained project scheduling method to deal with batch scheduling problems in process industries, where the products requirements are partitioned into batches. Neumann et al.[3] presented an advanced planning system for batch plant in process industries. The detailed production scheduling problem was decomposed into a batching problem and a batch scheduling problem, where the batching problem was about converting the primary requirements into individual batches, and the batch scheduling problem was about allocating the batches to resources. Prasad et al.[1] provided a MILP model for the

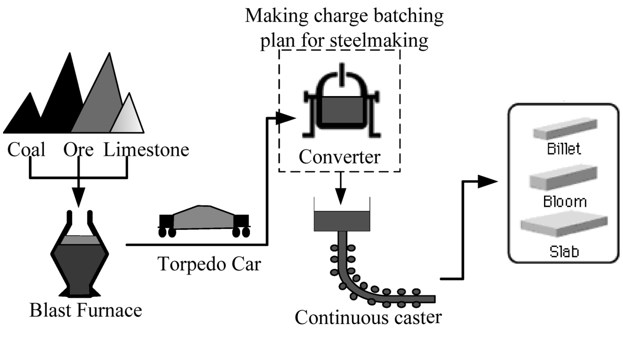

Figure 1. The steelmaking process.

* To whom correspondence should be addressed. Tel.: +86-24-83680169. Fax: +86-24-83680169. E-mail: qhjytlx@mail.neu.edu.cn.

10.1021/ie801454k CCC: $40.75
Published on Web 07/23/2009

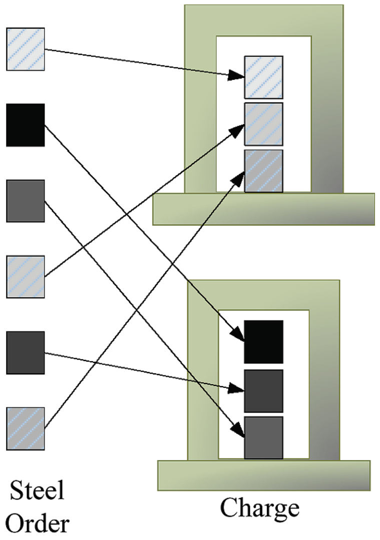

Figure 2. Sketch map of how to make a charge batching plan.

scheduling of the production of aluminum casts of different alloys with respect to the actual number of batches to be processed in the stages. As observed from the above, a variety of papers on the research of batching decisions discusses converting the primary requirements to batches. However our work presented in this paper discusses grouping orders into batches. The CBP is an operation management problem in the steelmaking stage. For operation management in this stage, most of the research efforts have concentrated on the scheduling problems. Numao and Morishita[14] proposed cooperative scheduling in which procedures, rules, and the users cooperate to make a feasible schedule for steelmaking−continuous casting (SCC). This approach was well applied to practical steelmaking industries in Japan. Tang et al.[5] formulated an integer programming model for the SCC scheduling problem which was solved by the algorithm that combines Lagrangian relaxation, dynamic programming, and heuristics. Pacciarelli and Pranzo[6] formulated steel ingots production in the SCC plant by means of the alternative graph, and it was solved by using a beam search procedure. Bellabdaoui and Teghem[4] presented a mixed-integer formulation for the SCC production scheduling with several technological constraints, solvable using standard software packages. The above literature takes the charge as their basic scheduling job while the composition of each charge is known in advance. The problem they considered is how to schedule the charges in the steelmaking process for optimizing the given objectives. However, our problem is to investigate the CBP in the steelmaking stage, which transforms the primary customer orders into various production batches (charges) subject to the steelmaking processing constraints.

The scope of this work is to develop a new mixed-integer programming optimization model for the CBP problem in the steelmaking process. We develop Lagrangian relaxation methods to solve the problem. The rest of the paper is organized as follows. Section 2 describes the problem and proposes a mixed-integer programming formulation for the CBP problem. Section 3 presents solution properties of the CBP model. Section 4 presents the framework of different Lagrangian relaxation methods. Details of the decomposed solution methodology, a heuristic method for obtaining feasible schedules, and the subgradient algorithm for solving the dual problem are given in this section. Computational results are presented in section 5 demonstrating the performance of the approach for the CBP of realistic sizes. Finally section 6 concludes the paper.

2. Problem Statement and Mathematical Formulation

2.1. Problem Description. The problem in this paper is establishing CBP in the steelmaking production. The production planning will be arranged according to the information about the customers' order. Each order has its own requirement on steelgrade, specification, due-date. And there is a difference of chemical and physical properties among orders. So it is necessary to combine different orders into various charges according to steelmaking requirements and order features. During the process of forming the CBP, the objectives are to maximize the finished product rate and to minimize the due-date difference, production cost, and amount of open orders under the smelting furnace capacity condition, where open orders are used to fill in the furnace but do not belong to any current orders.

2.2. Mathematical Formulation. The Model. Let us assume that a single order demand is less than furnace capacity and cannot be split. Parameters are listed as follows:

N = the number of orders to be grouped

P = the required number of charges to be formed

T = furnace capacity

P_j = additional cost coefficient of open order in the jth charge.

g_i = weight of the ith order.

h_i = penalty cost coefficient when ith order is not chosen.

C'_{ij} = steelgrade cost increase coefficient when order i is merged into order j.

Its value is given as follows (the steelgrade family is defined as the set of similar steelgrades):

$$C'_{ij} = \begin{cases} +\infty & \text{when order } i \text{ and order } j \text{ do not belong to the same steelgrade family} \\ +\infty & \text{when order } i \text{ and order } j \text{ belong to the same steelgrade family and steelgrade or order } i \text{ is higher than that of order } j \\ F_1(ST_i - ST_j) & \text{when order } i \text{ and order } j \text{ belong to the same steelgrade family and steelgrade of order } i \text{ is lower than that of order } j \\ 0 & \text{when order } i \text{ has the same steelgrade as order } j \end{cases}$$

where $i = 1, 2, ..., N$; $j = 1, 2, ..., N$, and F_1 = unit cost coefficient with difference of steelgrade. $C'_{ij} = F_1(ST_i - ST_j)$ indicates the cost of treating order i with steel grade value ST_i as order j with steelgrade value ST_j, where $ST_i > ST_j$. In practical production, the steelgrade value is generally taken in decreasing order of priorities of steelgrades, for example, if the steelgrade of order j is higher than order i, then $ST_i > ST_j$. When the steelgrade of order j is higher than that of order i, this coefficient represents the loss cost when the lower grade is treated as the higher one.

C''_{ij} = width penalty cost coefficient between order i and order j and its value is described as follows:

$$C''_{ij} = \begin{cases} 0 & \text{when order } i \text{ has the same width as order } j \\ +\infty & \text{if } [w_i, w_i + 100] \cap [w_j, w_j + 100] = 0, \\ F_2(b - a) & \text{if } [w_i, w_i + 100] \cap [w_j, w_j + 100] \neq 0 \end{cases}$$

where $i = 1,2, ..., N$, $j = 1,2, ..., N$, and F_2 is the unit width penalty cost coefficient, w_i is the rolling width of the ith order, a = the $\max\{w_i, w_j\}$, and $b = \min\{w_i + 100, w_j + 100\}$.

C'''_{ij} = due-date difference cost coefficient between order i and order j

$$C'''_{ij} = \begin{cases} F_3(d_i - d_j) & \text{if } d_i \geq d_j \\ F_4(d_j - d_i) & \text{if } d_i < d_j \end{cases} \quad i = 1, 2, ..., N;\ j = 1, 2, ..., N$$

where F_3 is the earliness penalty coefficient, F_4 is the tardiness penalty coefficient, and d_i is the due-date of order i.

Decision variables:

$$X_{ij} = \begin{cases} 1 & \text{order } i \text{ is merged into order } j \text{ when order } j \text{ is selected as a clustering center} \\ 0 & \text{otherwise} \end{cases}$$

The CBP model is given as follows:

(P):

$$\text{minimize} \sum_{i=1}^{N} \sum_{j=1}^{N} (C'_{ij} + C''_{ij} + C'''_{ij})X_{ij} + \sum_{j=1}^{N} p_j(TX_{jj} - \sum_{i=1}^{N} g_i X_{ij}) + \sum_{i=1}^{N} (1 - \sum_{j=1}^{N} X_{ij})h_i \quad (1)$$

subject to

$$\sum_{j=1}^{N} X_{ij} \leq 1, \quad i = 1, 2, ..., N \quad (2)$$

$$\sum_{j=1}^{N} X_{jj} = P \quad (3)$$

$$\sum_{i=1}^{N} g_i X_{ij} \leq TX_{jj}, \quad j = 1, 2, ..., N \quad (4)$$

$$X_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N \quad (5)$$

Objective function 1 minimizes the sum of loss cost caused by the difference of order steelgrade, specification, and due-date and the loss cost resulted from open order and unselected order; constraints 2 represent that each order can be assigned to only one clustering center at most; constraints 3 represent that the required number of charges to be formed is P; constraints 4 represent that total charge weight is less than furnace capacity when the jth is selected as a clustering center; constraints 5 express that variable X_{ij} has two possible values: 0 or 1.

3. Solution Properties of CBP Model

In this section, we present some properties by analyzing the CBP model. They are helpful for understanding solution structure and solving the CBP model.

Property 1: The CBP problem belongs to the NP-hard problem. The special case of the CBP is the capacitated p-median problem (CPMP). The CPMP consists in finding p facilities in the whole potential medians set to minimize the cost of serving all customers. Every customer must be assigned to some median. However, one major characteristic of the CBP is that each steel order in the order sets need not necessarily be assigned completely, but a penalty will be added into the objective function. We are in addition taking into account the open order quantity that will be penalized in the objective function of the CBP. Since our problem is a generalization of the CPMP and more complicated than the CPMP which is NP-hard,[8] the CBP problem is also NP-hard.

Table 1. Solution Matrix Structure

	$j = 1$	$j = 4$	$j = 2$	$j = 3$	$j = 5$
$i = 1$	1	0	0	0	0
$i = 4$	0	1	0	0	0
$i = 2$	1	0	0	0	0
$i = 3$	1	0	0	0	0
$i = 5$	0	1	0	0	0

Property 2: The open order quantity $(TX_{jj} - \sum_{i=1}^{N} g_i X_{ij})$ $(j = 1, 2, ..., N)$ can be solely determined once X_{ij} $(i = 1, 2, ..., N; j = 1, 2, ..., N)$ is determined.

Property 3: The possible element values in matrix are 0 or 1, and there is only one "1" or no "1" in each row.

Property 4: The sum of the primary diagonal element values is P.

Property 5: Solution matrix is asymmetrical.

Property 6: Nondiagonal element values depend on the diagonal element value; that is, when the diagonal element value is "0", the other element values situated in the same column are also "0".

Let us take five orders and two charges for example as shown in Table 1. So the possible value of i (or j) in the first line of Table 1 is from 1 to 5. The 0/1 value in Table 1 is the value of X_{ij}. For instance, if $i = 1$, $j = 1$ and the first number in the Table is 1, it means $X_{11} = 1$. Namely, order 1 is merged into order 1 and order 1 is selected as a clustering center. If $i = 2$, $j = 1$ and $X_{21} = 1$, order 2 is merged into order 1 and order 1 is selected as a clustering center. Property 3 shows that each order can be assigned to only one clustering center at most. Because the number of charges is P, given in advance, so property 4 holds. As property 6 shows, if some order is not selected as clustering center then other orders cannot be merged into it.

4. Lagrangian Relaxation Algorithm

Lagrangian relaxation (LR) is an efficient method to solve integer or mixed integer problems. By introducing Lagrangian multipliers, the coupling constraints are relaxed to the objective function to form a relaxed version of the primal problem, which can be decomposed into several independent subproblems easier to solve. Then the multipliers are updated along a subgradient direction. The objective value of the optimal solution of the relaxation problem is a lower bound to the primal problem. There are many researchers concentrated on LR algorithms and improved LR methods. Fisher[13] introduced traditional Lagrangian relaxation by dualizing the side constraints to produce a Lagrangian problem that is easy to solve. Suda et al.[9] presented six heuristics based on Lagrangian relaxation to solve the dual facility location problem. Except above ordinary Largangian relaxations, Jornsten et al.[15] put forward another Lagrangian relaxation approach based on variable splitting for mathematical programming problems. An application of this method to a generalized assignment problem (GAP) can be found in Kurt and Mikael.[10] Guignard and Kim[16] made full proofs and a more detailed treatment of this method. It has been proven that the bounds provided by variable splitting are at least as good as those provided by the ordinary Lagrangian relaxation. Paulo and Kurt[11] applied this method to obtain improved Lagrangian bounds for GAP successfully. Sridharan[12] provided a review of research on the capacitated plant location problem. The paper used many methods to solve the problem, such as Benders decomposition, variable splitting, dual ascent method, Lagrangian heuristic, and so on.

In this paper, two different Lagrangian relaxation frameworks are developed to solve the CBP. The upper bound is determined by a heuristic based on a solution to the relaxed problem. The multipliers are updated through a subgradient optimization method.

4.1. Lagrangian Relaxations. The detailed Lagrangian relaxation methods are given as follows. Lagrangian relaxation 1 is obtained by relaxing the assignment constraint. Thus, the relaxed problem is decomposed into knapsack subproblems. Lagrangian relaxation 2 is obtained by introducing a new variable and making use of the variable splitting technique. Let LR denote a Lagrangian relaxation problem.

We give the description of Lagrangian relaxation algorithm for solving the relaxed problem in pseudocode as follows. The structure of the algorithm is the same for both cases, the only difference is the solving of subproblems between LR1 and LR2. In the following description, **a** is for LR1, **b** is for LR2, and the remainder of pseudocode is the same for both. Here n is used to account for the number of iterations, Z^{U} is upper bound, Z^{L} is lower bound, and u is the multipliers. $Z_{\mathrm{D}}(u^n)$ is the optimal objective value of the relaxed problem at the nth iteration. Z^n is the objective value of the feasible solution through the constructed heuristic at the nth iteration.

```
Procedure for CBP
Initialize: $n = 0$; $Z^U = +\infty$; $Z^L = -\infty$; $u_i^0 = 0$, $i = 1, ..., N$.
While (stop criterion is not satisfied) do
    (a) Solve the relaxed problem by decomposition as described in section 4.1.1. If $Z_D(u^n) > Z^L$, then $Z^L = Z_D(u^n)$.
    (b) Solve the relaxed problem by decomposition as described in section 4.1.2. If $Z_D(u^n) > Z^L$, then $Z^L = Z_D(u^n)$.
    Construct a feasible solution to the original problem P by using the heuristics presented in section 4.2. If $Z^n < Z^U$, then $Z^U = Z^n$.
    Update Lagrangian multipliers $u$ by using the subgradient method presented in subsection 4.3.
    $n = n + 1$.
end_while;
write results;
end_procedure.
```

4.1.1. Lagrangian Relaxation 1. In problem P we observe that constraint 2 enforcing each order to be assigned to at most one clustering center combines all x variables. Consider a Lagrangian relaxation scheme that dualizes the contraint 2 to decompose the problem by order. By relaxing the constraint 2 to the objective function through nonnegative Lagrangian multipliers $\{u_i\}$, we obtain LR1.

LR1:

$$\text{minimize } L_1(u) \equiv \sum_{i=1}^{N}\sum_{j=1}^{N}(C'_{ij} + C''_{ij} + C'''_{ij})X_{ij} + \sum_{j=1}^{N} p_j(TX_{jj} - \sum_{i=1}^{N} g_i X_{ij}) + \sum_{i=1}^{N}(1 - \sum_{j=1}^{N} X_{ij})h_i + \sum_{i=1}^{N} u_i(\sum_{j=1}^{N} X_{ij} - 1) \quad (6)$$

subject to constraints 3−5.

For given values of $\{u_i\}$, the relaxed problem LR1 is transformed to

$$\text{minimize} \sum_{i=1}^{N}\sum_{j=1}^{N}(C'_{ij} + C''_{ij} + C'''_{ij} - p_j g_i - h_i + u_i)X_{ij} + \sum_{j=1}^{N} Tp_j X_{jj} + \sum_{i=1}^{N} h_i - \sum_{i=1}^{N} u_i \quad (7)$$

subject to

$$\sum_{j=1}^{N} X_{jj} = P \quad (8)$$

$$\sum_{i=1}^{N} g_i X_{ij} \le TX_{jj}, \quad j = 1, 2, ..., N \quad (9)$$

$$X_{ij} \in \{0, 1\} \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N \quad (10)$$

The Lagrangian dual to problem P is LD:

$$\max_{u \ge 0} L_1(u) \quad (11)$$

subject to constraints 3−5.

We note in addition that LR1 can be separated into N smaller subproblems. So we consider N separate problems LR1(j), one for each order. If we select order j to be a clustering center by setting X_{jj} equal to one, the subproblem LR1(j) will reduce to a generalization of the knapsack problem that we refer to as $\mathrm{KN_x}$. To solve problem LR1, we reformulate it as follows. Suppose that each order is thought as a clustering center. That is $X_{jj} = 1$ for j=1, ..., N. Then the contribution to LR1 involving this assumption would be $\mathrm{KN_x}$ which is given by the following:

$\mathrm{KN_x}$:

$$\mathrm{V1}_j = \text{minimize} \sum_{i=1}^{N}(C'_{ij} + C''_{ij} + C'''_{ij} - p_j g_i - h_i + u_i)X_{ij} + Tp_j \quad (12)$$

subject to

$$\sum_{i=1}^{N} g_i X_{ij} \le T, \quad j = 1, 2, ..., N \quad (13)$$

$$X_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N \quad (14)$$

Problem LR1 now can be reformulated as

$\mathrm{KN_y}$:

$$\text{minimize} \sum_{j=1}^{N} \mathrm{V1}_j X_{jj} \quad (15)$$

subject to

$$\sum_{j=1}^{N} X_{jj} = P \quad (16)$$

$$X_{jj} \in \{0, 1\}, \quad j = 1, 2, ..., N \quad (17)$$

Therefore, $L_1(u)$ can be written as

$$L_1(u) = \sum_{j=1}^{N} \mathrm{V1}_j X_{jj} + (\sum_{i=1}^{N} h_i - \sum_{i=1}^{N} u_i) \quad (18)$$

Tp_j is a constant given in advance. Model ($\mathrm{KN_x}$) is a knapsack problem. To solve $\mathrm{KN_x}$, we need to solve a 0−1 knapsack problem. Here we use dynamic programming algorithm to solve $\mathrm{KN_x}$.

4.1.1.1. Dynamic Programming for Subproblems ($\mathrm{KN_x}$). The 0−1 knapsack problem has n stages by using backward dynamic programming.[17] The strategy is to decide whether or not the kth goods is put into bag at stage k ($1 \le k \le n$). Let $f(i,j)$ be the optimal value of knapsack problem when knapsack capacity is j and the range of goods can be chosen from i to n. w_i denotes the weight of the goods and v_i denotes the value of the goods. The backward dynamic programming is used to solve the subproblems $\mathrm{KN_x}$. In the CBP problem, v_i denotes the penalty cost of ith order and w_i denotes the weight of ith order.

Recursion relation:

$$f(i,j) = \begin{cases} \max\{f(i+1,j), f(i+1,j-w_i)+v_i\} & j \ge w_i \\ f(i+1,j) & 0 \le j < w_i \end{cases} \tag{19}$$

Boundary conditions:

$$f(n,j) = \begin{cases} v_n & j \ge w_i \\ 0 & 0 \le j < w_i \end{cases} \tag{20}$$

$\mathrm{V1}_j$ is the jth optimal knapsack value. Then index the knapsack in the ascending order of $\mathrm{V1}_j$ value. The optimal value of model $\mathrm{KN_y}$ is the first P smallest numbers in the ordered sequence.

4.1.2. Lagrangian Relaxation 2. Another Lagrangian decomposition scheme based on variable splitting is applied for this problem. The method is based on a reformulation of the problem by introducing identical copies of some subset of the original variables. The new constraints are added to the original model to set original variables' copies equal to the original variables. The LR problem is formed by relaxing the new constraints to the objective function through the Lagrangian multiplier, yielding a number of smaller, decomposed subproblems which are easier to solve. The dual problem retains all constraints of the original problem using this method, and it would obtain the improved lower bound.

For example, when the original variable is X, the copy of X is S. $X = S$ becomes a new constraint which is introduced into the model. Then, by defining the copy and relaxing the coupling constraints $X = S$, it is possible to decompose the Lagrangian relaxation problem into one or more independent subproblems. In this way, the relaxed constraints are the newly introduced constraints, and the constraints in the relaxed problem are the same as (P). To make use of the variable splitting method in our model, a new variable S_{ij} (let $S_{ij} = X_{ij}$) is introduced. The variable X_{ij} is decomposed into $\alpha X_{ij} + \beta S_{ij}$ in the objective function. X_{ij} is replaced by S_{ij} in some of the constraints. Let $\alpha + \beta = 1$, then the original problem can be rewritten as follows.

$$\text{minimize} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha(C'_{ij} + C''_{ij} + C'''_{ij})X_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N}\beta(C'_{ij} + C''_{ij} + C'''_{ij})S_{ij} + \sum_{j=1}^{N} p_j \mathrm{T} X_{jj} - \sum_{i=1}^{N}\sum_{j=1}^{N} p_j g_i(\alpha X_{ij} + \beta S_{ij}) + \sum_{i=1}^{N} h_i - \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(\alpha X_{ij} + \beta S_{ij}) \tag{21}$$

subject to

$$\sum_{j=1}^{N} S_{ij} \le 1, \quad i = 1, 2, ..., N \tag{22}$$

$$\sum_{j=1}^{N} X_{jj} = P \tag{23}$$

$$\sum_{i=1}^{N} g_i X_{ij} \le \mathrm{T}X_{jj} \quad j = 1, 2, ..., N \tag{24}$$

$$X_{ij} = S_{ij}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N, \tag{25}$$

$$X_{ij}, S_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N, \tag{26}$$

By relaxing eq 25 to the objective function through Lagrangian multipliers $\{v_{ij}\}$ (unrestricated in sign), we obtain the relaxation. LR2:

$$\text{minimize } L_2(v) \equiv \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha(C'_{ij} + C''_{ij} + C'''_{ij})X_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N}\beta(C'_{ij} + C''_{ij} + C'''_{ij})S_{ij} + \sum_{j=1}^{N} p_j TX_{jj} - \sum_{i=1}^{N}\sum_{j=1}^{N} p_i g_i(\alpha X_{ij} + \beta S_{ij}) + \sum_{i=1}^{N} h_i - \sum_{i=1}^{N}\sum_{j=1}^{N} h_i(\alpha X_{ij} + \beta S_{ij}) + \sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij}(X_{ij} - S_{ij}) \tag{27}$$

subject to constraints 22−24 and 26.

For given values of $\{v_{ij}\}$, the relaxed problem LR2 separates into two subproblems, namely:

$\mathrm{LR2_s}$:

$$\text{minimize} \sum_{i=1}^{N}\sum_{j=1}^{N}(\beta C'_{ij} + \beta C''_{ij} + \beta C'''_{ij} - \beta p_j g_i - \beta h_i - v_{ij})S_{ij} \tag{28}$$

subject to

$$\sum_{j=1}^{N} S_{ij} \le 1, \quad i = 1, 2, ..., N \tag{29}$$

$$S_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N, \tag{30}$$

$\mathrm{LR2_x}$:

$$\text{minimize} \sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha C'_{ij} + \alpha C''_{ij} + \alpha C'''_{ij} - \alpha p_j g_i - \alpha h_i + v_{ij})X_{ij} + \sum_{j=1}^{N} Tp_j X_{jj} + \sum_{i=1}^{N} h_i \tag{31}$$

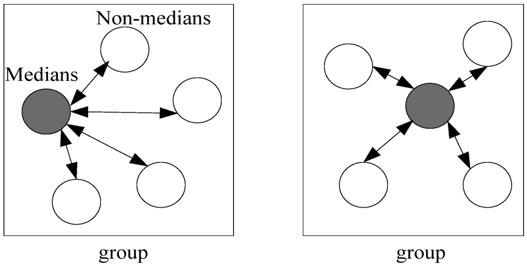

Figure 3. Illustation of swapping medians with nonmedians.

subject to

$$\sum_{j=1}^{N} X_{jj} = P \tag{32}$$

$$\sum_{i=1}^{N} g_i X_{ij} \le TX_{jj}, \quad j = 1, 2, ..., N \tag{33}$$

$$X_{ij} \in \{0, 1\}, \quad i = 1, 2, ..., N; \quad j = 1, 2, ..., N, \tag{34}$$

Then the dual problem is

LD:

$$\max_{v} L_2(v) \tag{35}$$

subject to constraints 22−24 and 26.

4.2. Construction of a Feasible Schedule. The solution X_{ij} from dual problems is always related to infeasible solutions because some of the relaxed constraints might be violated. But clustering centers obtained from dual problems can be used to produce feasible solutions. Therefore, a two-phases heuristic approach 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 solution satisfies the relaxed constraint. The method that translates the infeasible solutions to feasible solutions is similar in the above two different Lagrangian relaxation methods. So following we just take a heuristic of Lagrangian relaxation 1 for example. In the second phase, the solution is improved using local search heuristics. Let the solution of relaxed problem at the kth iteration be denoted as $\{X_{ij}^k, i = 1,..., N, j = 1,..., N\}$ and expressed by $\{X_{ij}, i = 1,..., N, j = 1,..., N\}$ in the heuristic algorithm. Let d_{ij} be the difference cost between order i and order j. $d_{ij} = C'_{ij} + C''_{ij} + C'''_{ij} - p_j g_i - h_i + u_i$, namely the coefficient of X_{ij} in the LR1. To describe easily, clustering sets are called "groups" for short, the clustering center is called "median" for short, and order is called "node" for short. The algorithmic steps of the heuristic are described as follows:

Phase I.

Step 1. Set $i = 1$.

Step 2. For each i, check whether $\sum_{j=1}^{N} X_{ij}$ is no greater than 1. If this is satisfied, go to step 4; else go to step 3.

Step 3. Calculate

$$d_{iJ} = \min_{j=1,...,N} \{d_{ij} | X_{jj} = 1, d_{ij} \le 0\},$$
$$\text{let} \quad X_{iJ} = 1,\ X_{ij} = 0, \quad \text{for} \quad j = 1, ..., N \quad \text{and} \quad j \ne J$$

Step 4. $i = i + 1$. If $i \le N$, go to step 2; otherwise, proceed to the second phase.

Phase II. Solution X_{ij} is further improved by a reallocation heuristic and an interchange heuristic. They are the local search methods to improve the feasible solutions found so far.

Table 3. Testing Results of Two Algorithms (Order Numbers = 30, 50, 70)

problem scenario	problem structure order no. × total charges	duality gap (%) LR1	duality gap (%) LR2	running time (s) LR1	running time (s) LR2
1	30 × 1	0.10	0.10	0.04	0.56
2	30 × 2	0.14	0.33	0.05	0.04
3	30 × 3	1.39	0.60	0.12	0.14
4	50 × 1	0.00	0.00	0.00	0.00
5	50 × 2	1.15	0.25	0.18	0.05
6	50 × 3	3.14	0.37	0.45	0.11
7	50 × 4	3.73	0.58	0.47	0.23
8	50 × 5	7.26	0.89	0.63	0.40
9	70 × 1	0.00	0.00	0.01	0.00
10	70 × 2	1.33	0.09	0.37	0.10
11	70 × 3	2.17	0.26	0.60	0.04
12	70 × 4	2.55	0.37	0.74	0.27
13	70 × 5	4.13	0.67	0.96	0.70
14	70 × 6	4.77	0.54	1.22	0.87
15	70 × 7	3.64	1.24	1.22	1.41
average		3.55	0.42	0.71	0.33

Table 4. Testing Results of Two Algorithms (Order Numbers = 100)

problem scenario	problem structure order no. × total charges	duality gap (%) LR1	duality gap (%) LR2	running time (s) LR1	running time (s) LR2
16	100 × 1	0.00	0.00	0.00	0.00
17	100 × 2	0.76	0.03	1.56	0.22
18	100 × 3	1.87	0.08	3.09	0.73
19	100 × 4	2.41	0.25	3.83	1.62
20	100 × 5	2.78	0.61	4.21	3.39
21	100 × 6	3.20	0.65	4.59	3.00
22	100 × 7	3.86	1.01	5.00	4.42
23	100 × 8	5.29	1.75	5.00	5.00
24	100 × 9	6.42	2.64	5.00	5.00
25	100 × 10	8.15	3.21	5.00	5.00
average		3.47	1.02	3.73	2.84

Reallocation Heuristic: Solution X_{jj} can be improved by searching for a new median inside each group. As Figure 3 shows, if the new solution by swapping the current median with a nonmedian and reallocating is better, we can repeat the reallocation process inside the new groups and all the process until no more improvements are reached.[7]

Interchange Heuristic: Repeat this step to find exchangeable nodes with reduced cost, either two nodes both in different groups or one node in one group the other is not selected to any group until no reduced cost can be found.

4.3. Updating Lagrangian Multipliers. The subgradient method is commonly used to solve Lagrangian dual problems that require optimally solving all subproblems at each iteration to obtain a subgradient direction. We also apply the subgradient method to two Lagrangian relaxation algorithms. Here take the first Lagrangian relaxation for example. In our algorithm of LR1, the multiplier is updated according to

$$u^{n+1} = u^n + s^n g^n \tag{36}$$

where n is the iteration index, $g^n = g(u^n)$ is the subgradient of $L(u)$ at u^n with elements equal to $(\sum_{j=1}^{N} X_{ij} - 1)$, and s^n is the

Table 2. Testing Results of LR2 on the Different Configurations of α, β

		duality gap (%)										
	α	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
problem structure	β	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
30 × 1		0.00	1.40	5.42	5.92	13.55	19.29	18.56	54.66	48.61	57.41	58.05
50 × 2		0.49	3.22	6.16	12.47	12.91	8.83	26.54	42.40	38.81	48.87	53.60
70 × 3		0.00	2.16	9.01	11.91	27.24	22.46	23.58	44.80	50.60	35.96	75.57
100 × 4		0.08	6.27	9.94	36.00	30.11	175.03	49.29	86.98	115.93	74.91	247.82
average		0.26	2.34	5.26	11.22	14.14	37.77	19.83	38.31	42.49	36.36	72.67

step size at the *n*th iteration given by

$$s^n = \lambda \frac{L^* - L^n}{||g^n||^2}, \quad 0 < \lambda < 2 \tag{37}$$

where L^* is the optimal value estimated by the best feasible solution obtained so far and L^n is the value of the dual function at the *n*th iteration. The algorithm terminates when the duality gap is less than a small number or a fixed iterative number is reached.

5. Computational Results

The method was implemented by using C++, and experiment was carried out on a PC with Pentium-IV (2.99 GHz) CPU. The algorithms are evaluated with two performance measures: average duality gap and computation time (seconds). The duality gap is computed by subtracting the value Z^L of the dual function (a lower bound of the original problem) from the objective value Z^U of a feasible schedule (an upper bound), dividing by the value Z^L of the dual function and converting to a percentage. The upper bound is the best objective value found so far through the constructed heuristic during the iteration process of LR algorithm.

According to practical data from Baosteel Complex, China, the due-dates of orders were randomly generated from a uniform distribution [1, 10], and weights of each order were randomly generated from a uniform distribution [10, 30]. Two other parameters were chosen to represent the problem structure: (1) The number of orders to vary at four levels: 30, 50, 70, 100. (2) The required number of charges is set to vary at different levels, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, according to the order numbers and the steelgrades of orders.

Our method can cope with practical problems with different number of orders and charges. Different configurations give 25 problem scenarios. For each configuration, 10 instances are randomly generated based on each set, therefore resulting in a total of 250 test problems used in this experiment. When the number of orders is set to be 30, 50, 70, the iteration number 200 is imposed as the stopping criterion. To compare the performance of the approaches for CBP of larger sizes, when the numbers of orders is extended to 100, a limited computational time, 5 s, is also considered as one of the stopping criteria in order to test two different methods on problems in the same time restriction.

In the LR2 algorithm, the value of α and β is selected from the pairs {(1, 0), (0.9, 0.1), (0.8, 0.2), (0.7, 0.3), (0.6, 0.4), (0.5, 0.5), (0.4, 0.6), (0.3, 0.7), (0.2, 0.8), (0.1, 0.9), (0, 1)}. Before the comparison of LR1 and LR2, the preliminary experiments are performed for setting the values of α, β in the LR2. Small scale data are tested using LR2 under the 11 pairs of α, β. Table 2 reports the experimental results of the data in detail, including the duality gap of the LR2 under the all configurations of α, β. From the preliminary experiments results in the Table 2, α = 1.0, β = 0.0 is a good configuration when using LR2. On the basis of this configuration, all data are tested in the experiment of LR2 by setting the configuration as α = 1.0, β = 0.0 to compare the effect of LR1 and LR2.

The optimality performance and running times of two LR methods against different problem structures and different problem sizes are presented in Table 3 and Table 4. We refer

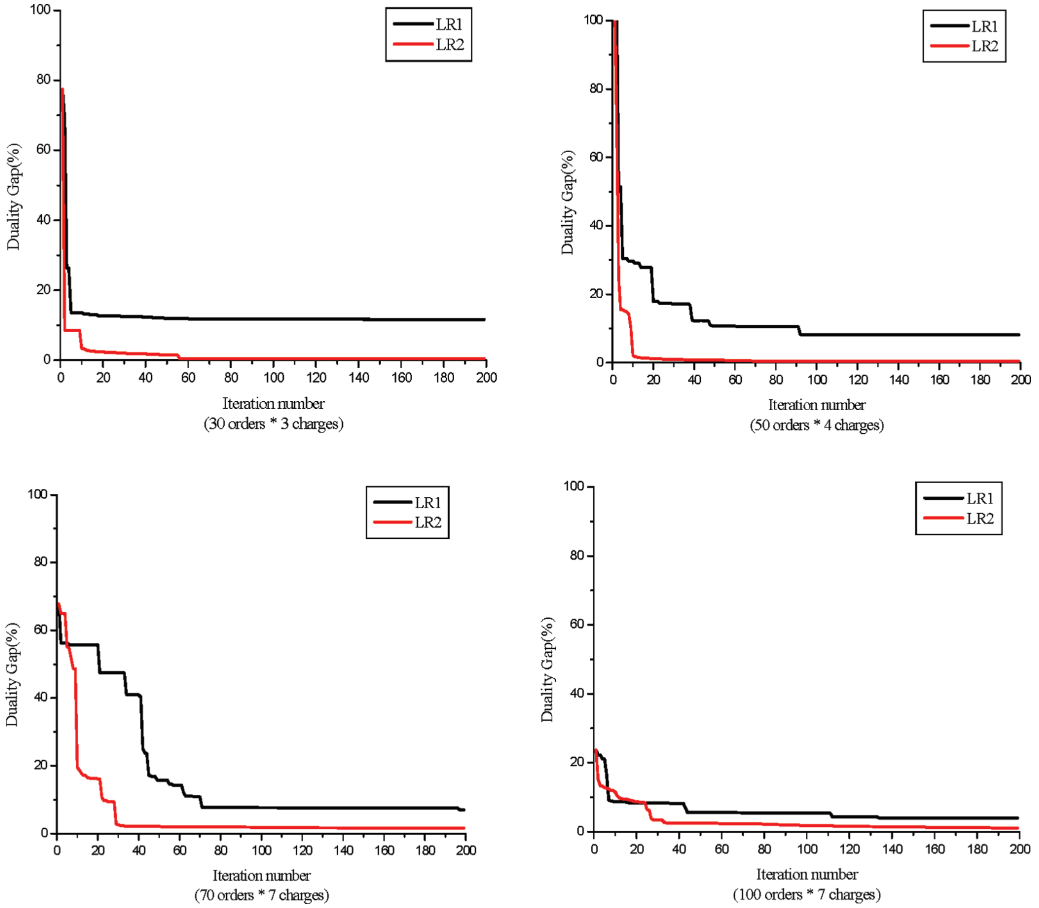

Figure 4. Duality gap for different number of orders and charges.

to ordinary LR algorithm as LR1 and another using variable splitting as LR2.

Figure 4 illustrates the evolution of the duality gap for different numbers of orders and charges. From the figures we can observe that as compared with LR1, LR2 converges much more quickly. It proves that LR2 can approach a much smaller duality gap in comparatively few iteration numbers.

From the results presented in Table 3 and Table 4, the following observations can be made about our CBP.

(1) All the values shown in the Table 3 and 4 above represent the average of performance measures for 10 instances of corresponding problem configuration. As shown in the Tables 3 and 4, the average duality gap of LR2 is smaller than LR1 in the most of instances. It shows that LR2 outperforms LR1. From the testing results shown in Table 4, we can see average duality gaps of two LRs for problems with 100 orders are 3.47 and 1.02, respectively. The average computation times are 3.73 and 2.84, respectively. From these results, we can draw the conclusion that both LR methods can produce satisfactory average duality gaps and LR2 generates better solutions in a much shorter computation time on average, as compared to LR1.

(2) When the number of orders is fixed, the duality gap augments and the computation time increases as the number of charge increases. This is consistent with the intuition that for a fixed number of orders, when the number of charge increase, the space of orders that can be chosen for each charge becomes smaller and the problem becomes harder to solve.

6. Conclusion

On the basis of the steelmaking process, a new CBP model is presented by considering the practical constraints and requirements on steelgrade, specification, and due-date of orders to be scheduled. Two different Lagrangian relaxation frameworks are developed to solve the CBP problem. Compared to the ordinary Lagrangian relaxation, the variable splitting method can obtain the tighter bound and smaller dual gap. Computational results show that the proposed method is very effective and can give good charge batching plans. Further research may be done on exploring feasible application of variable splitting in other aspects.

Acknowledgment

This research is partly supported by National Natural Science Foundation (Grant No. 70425003), National 863 High-Tech Research and Development Program of China through approved No. 2006AA04Z174, and National Natural Science Foundation of China (Grant No. 60674084).

Literature Cited

(1) Prasad, P.; Maravelias, C. T.; Kelly, J. Optimization of aluminum smelter casthouse operations. *Ind. Eng. Chem. Res.* **2006**, *45*, 7603–7617.

(2) Trautmann, N.; Schwindt, C. Batch scheduling in process industries: An application of resource-constrained project scheduling. *OR Spectrum* **2000**, *22*, 501–524.

(3) Neumann, K.; Schwindt, C.; Trautmann, N. Advanced production scheduling for batch plants in process industries. *OR Spectrum* **2002**, *24*, 251–279.

(4) Bellabdaoui, A.; Teghem, J. A mixed-integer linear programming model for the continuous casting planning. *Int. J. Prod. Econ.* **2006**, *104*, 260–270.

(5) Tang, L. X.; Luh, P. B.; Liu, J. Y.; Fang, L. Steel-making process scheduling using Lagrangian relaxation. *Int. J. Prod. Res.* **2002**, *40*, 50–70.

(6) Pacciarelli, D.; Pranzo, M. Production scheduling in a steelmaking-continuous casting plant. *Comput. Chem. Eng.* **2004**, *28*, 2823–2835.

(7) Lorena, L. A. N. Local search heuristics for capacitated p-median problems. *Networks Spatial Econ.* **2003**, *3*, 407–419.

(8) Luiz, A. N.; Edson, L. F. A column generation approach to capacitated p-median problems. *Comput. Operat. Res.* **2004**, *31*, 863–876.

(9) Suda, T.; John, H.; Mikael, R. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *Eur. J. Operat. Res.* **1997**, *102*, 611–625.

(10) Kurt, J.; Mikael, N. A new Lagrangian Relaxation approach to the generalized assignment problem. *Eur. J. Operat. Res.* **1986**, *27*, 313–323.

(11) Paulo, B.; Kurt, J. Improved Lagrangian decomposition: An application to the generalized assignment problem. *Eur. J. Operat. Res.* **1990**, *46*, 84–92.

(12) Sridharan, R. The capacitated plant location problem. *Eur. J. Operat. Res.* **1995**, *87*, 203–213.

(13) Fisher, M. L. The Lagrangian Relaxation method for solving integer programming problems. *Eur. J. Operat. Res.* **1987**, *28*, 3–21.

(14) Numao, M.; Morishita, S. Cooperative scheduling and its application to steelmaking process. *IEEE Trans. Ind. Electron.* **1991**, *38*, 150–155.

(15) Jornsten, K.; Nasberg, M.; Smeds, P. Variable Splitting: A New Lagrangean Relaxation Approach to Some Mathematical Programming Models. Working paper LITH-MAT-R-85-04, 1985, Linkoping University.

(16) Guignard, M.; Kim, S. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Math. Program.* **1987**, *39*, 215–228.

(17) Ibaraki, T. Enumerative approaches to combinatorial optimization, part II. *Ann. Operat. Res.* **1987**, *11*, 343−602.

Received for review September 27, 2008
Revised manuscript received June 3, 2009
Accepted July 6, 2009

IE801454K