Computers and Chemical Engineering 106 (2017) 133–146

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Review

An improved Lagrangian relaxation approach to scheduling steelmaking-continuous casting process

Haijuan Cui, Xiaochuan Luo*

State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110004, China

ARTICLE INFO

Article history:
Received 5 November 2016
Received in revised form 24 May 2017
Accepted 25 May 2017
Available online 6 June 2017

Keywords:
Scheduling problem
Steelmaking continuous casting
Surrogate subgradient method
Lagrangian relaxation
Heuristic method

ABSTRACT

In the steelmaking continuous-casting (SCC) process, scheduling problem is a key issue for the iron and steel production. To improve the productivity and reduce material consumption, optimal models and approaches are the most useful tools for production scheduling problems. In this paper, we firstly develop a mixed integer nonlinear mathematical model for the SCC scheduling problem. Due to its combinatorial nature and complex practical constraints, it is extremely difficult to cope with this problem. In order to obtain a near-optimal schedule in a reasonable computational time, Lagrangian relaxation approach is developed to solve this SCC scheduling problem by relaxing some complicated constraints. Owing to the existence of the nonseparability coming from the product of two binary variables, it is still hard to deal with this relaxed problem. By making use of their characteristics, the subproblems of the relaxed problem can be converted into different difference of convex (DC) programming problems, which can be solved effectively by using the concave–convex procedure. Under some reasonable assumptions, the convergence of the concave–convex procedure can be established. Furthermore, we introduce an improved conditional surrogate subgradient algorithm to solve the Lagrangian dual (LD) problem and analyze its convergence under some appropriate assumptions. In addition, we present a simple heuristic algorithm to construct a feasible schedule by adjusting the solutions of the relaxed problem. Lastly, some numerical results are reported to illustrate the efficiency and effectiveness of the proposed method.

Contents

* Corresponding author.
E-mail address: luoxch@mail.neu.edu.cn (X. Luo).

http://dx.doi.org/10.1016/j.compchemeng.2017.05.026
0098-1354/

1. Introduction

The iron and steel industry, one of the cornerstone industries, makes a material contribution to the world economy by providing raw materials for a number of other important industries, such as, machinery manufacturing, shipbuilding, petro-chemical and construction industry. Unlike other industries, the process of iron and steel production runs at high-temperature and high-weight material flow with complicated technological processes and extensive energy consumption (Li et al., 2012). In the iron and steel industry, SCC process plays a significant role, since it is one of the largest bottlenecks in the manufacturing process. Production scheduling in the iron and steel industry has been recognized as one of the most difficult industrial scheduling problems (Harjunkoski and Grossmann, 2001). In the SCC process, two main tasks of scheduling are to determine which orders are allocated to each machine, and assign the sequence of orders allocated at each stage (Li et al., 2011; Tang et al., 2000). Due to its combinatorial nature, strict requirements on material continuity and complex practical constraints, it is extremely challenging to solve the scheduling problems of the SCC process. Optimal scheduling of the SCC process can effectively improve machine productivity, reduce material and energy consumption and minimize production cost (Li et al., 2016b; Ye et al., 2014). Therefore, it is critical to develop an effective and efficient optimization model and approach to cope with the complicated scheduling problem of the SCC process.

In recent years, most published works on optimization models and approaches for scheduling problems of the SCC process can be roughly classified into three categories: mathematical programming methods, artificial intelligence methods and heuristic methods. Using mathematical programming methods, Bellabdaoui and Teghem (2006) presented a mixed integer mathematical model for the scheduling of steelmaking continuous casting production, which can be solved by using some standard software packages. Harjunkoski and Grossmann (2001) presented a decomposition algorithm to split the large scheduling problem of steel industry into smaller subproblems that can often be solved optimally by using mathematical programming methods. Mao et al. (2014) modeled the SCC scheduling problem as a mixed-integer linear programming problem and proposed a novel Lagrangian relaxation approach to solve this problem. Mao et al. (2015) presented a time-index formulation for the SCC scheduling problem and developed an effective subgradient method and dynamic programming approach to deal with this scheduling problem. Tang et al. (2002) formulated a novel integer programming formulation with a separable structure for SCC scheduling problem and developed an improved solution method by combining Lagrangian relaxation, dynamic programming and heuristics to solve this problem. Ye et al. (2014) introduced robust optimization and stochastic programming approaches for addressing a medium-term production scheduling of the large-scale steelmaking continuous casting process under demand uncertainty. With respect to artificial intelligence methods, Atighehchian et al. (2009) investigated a novel iterative algorithm by combining ant colony optimization and nonlinear optimization methods for scheduling of the SCC production. Jiang et al. (2015) investigated a mathematic programming model for the SCC scheduling problem with controllable processing times and proposed a meta-heuristic algorithm by comparing differential evolution algorithm with a variable neighborhood decomposition search to address this problem. Li et al. (2014) formulated a realistic hybrid flowshop scheduling problem model for steelmaking casting process and developed an effective fruit fly optimization algorithm to solve the steelmaking casting problem. Li et al. (2016) proposed a hybrid fruit fly optimization algorithm and successfully applied to solve the hybrid flowshop rescheduling problem with flexible processing time in steelmaking casting systems. Long et al. (2016) studied a dynamic scheduling model with NP-hard feature for the SCC scheduling problem under the continuous caster breakdown and developed a hybrid algorithm featuring a genetic algorithm combined with a general variable neighbourhood search to solve this model. Pan (2016) addressed a new SCC scheduling problem arising from iron and steel production process, modeled this problem as a combination of two coupled sub-problems and presented a novel cooperative co-evolutionary artificial bee colony algorithm with two sub-swarms to address the sub-problems of this scheduling problem, respectively. Tang and Wang (2010) designed an improved particle swarm optimization algorithm for the hybrid flowshop scheduling problem in the integrated production process of steelmaking continuous-casting. Tang et al. (2014) studied an improved differential evolution algorithm to solve a challenging problem of dynamic scheduling in the SCC production. Zhao et al. (2011) formulated a mathematical programming model for the SCC scheduling problem and proposed a tabu search algorithm to deal with the allocation and sequencing decisions. As for heuristic methods, Missbauer et al. (2009) proposed a mixed integer linear programming model for the SCC scheduling problem and presented a three-stage heuristic solution procedure to improve the schedule by means of a linear programming model. Pacciarelli and Pranzo (2004) modeled the SCC scheduling problem by means of the alternative graph and described a beam search procedure to tackle with this problem. Yu and Pan (2012) proposed a three-stage rescheduling method including the batches splitting, forward scheduling method and backward scheduling method for solving a novel multi-objective nonlinear programming model of the SCC production process. Yu et al. (2016) considered a job start-time delay issue for the SCC rescheduling problem and carried out an effective heuristic rescheduling algorithm for the SCC production system to quickly respond to any disruption with a proper rescheduling plan.

Inspired by the above existing literatures, the motivation and main contribution of this paper are in following directions. Firstly, the optimization models for the SCC scheduling problems are usually described by adopting a big-M strategy (Harjunkoski and Grossmann, 2001; Jiang et al., 2015; Li et al., 2016a; Long et al., 2016; Missbauer et al., 2009; Mao et al., 2014; Pan, 2016; Tang et al., 2002, 2014; Tang and Wang, 2008; Ye et al., 2014), which play a significant role in improving the productivity and reducing the cost of the entire production process. In the big-M strategy, the main drawbacks are that the computation time will increase owing to the existence of redundant constraints (Tang et al., 2013; Vallada and Ruiz, 2011) and the big-M formulation usually produces much looser lower bound (Mao et al., 2015). As a result, we address a new mixed integer nonlinear mathematical model for the SCC scheduling problem without using the big-M strategy to avoid above weaknesses. Secondly, in most cases, scheduling problems of the iron and steel industry are NP-hard, which implies that no algorithm can optimally solve these problems within a reasonable computation time (Chen and Luh, 2003). In 1988, Gupta (1988) has proved that the two-stage flowshop problem with identical multiple machines at each stage is NP-hard and two-stage flowshop problem is also NP-hard even if the number of machines at one of the two stage is one. Due to the complexity, the SCC scheduling problem addressed in this paper is much more complicated than the two stages flowshop scheduling problem (Gupta, 1988), which means that the SCC scheduling problem is also NP-hard. Therefore, the SCC scheduling problem cannot be solved optimally within the reasonable computation time. Thus, Lagrangian relaxation approach is introduced to deal with the SCC scheduling problem, because this approach can provide a lower bound to evaluate the optimality of solutions and yield a near-optimal schedule in a reasonable computational time (Nishi and Hiranaka, 2013). Up to now, published works on the Lagrangian relaxation approaches have mainly focused on relaxing the compli-

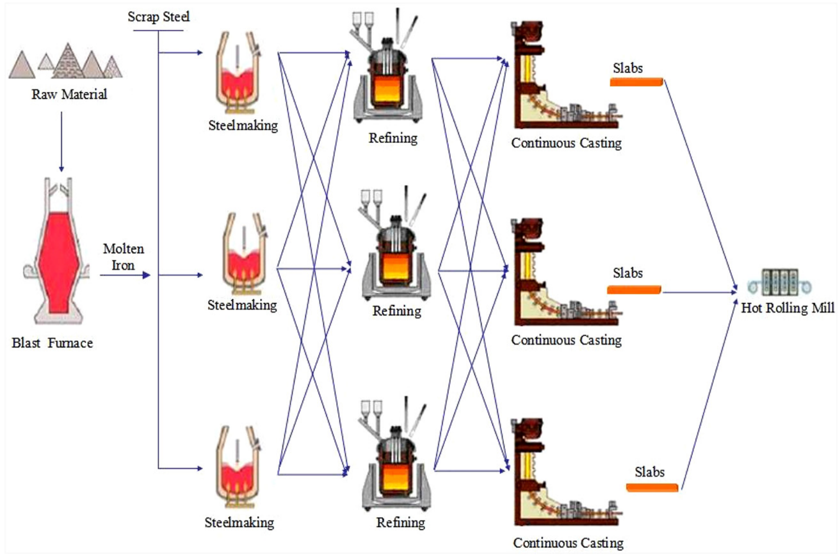

Fig. 1. The main process of steelmaking-continuous casting.

cated constraints to decompose relaxed problems into some simple subproblems, whose optimal solutions can be obtained easily (Buil et al., 2012; Fu and Diabat, 2015; Mao et al., 2014, 2015; Nishi et al., 2010; Nishi and Hiranaka, 2013; Sun and Yu, 2015; Tang et al., 2002; Tang and Liu, 2007). Nevertheless, for some complicated scheduling problems, it is extremely difficult to decompose the relaxed problem into some simple subproblems, due to the nonseparability coming from the product of two binary variables. To the best of our knowledge, few studies have been reported on how to cope with the nonseparable issue of the relaxed problem for the SCC scheduling problems. By relaxing the complex constraints, the relaxed problem can be decomposed into two separable subproblems. However, it is extremely hard to obtain the optimal solutions of these two subproblems, because the Lagrange function is nonseparable. By making use of the characteristics of the relaxed problem, the subproblems are converted into different DC programming problems that can be solved by using the concave–convex procedure. Thirdly, the multipliers are usually updated along the subgradient directions that can be obtained by fully optimizing the relaxed problem, which means that the convergence of the concave–convex procedure plays a crucial role in updating the Lagrangian multipliers. Under some reasonable assumptions, its convergence is analyzed to guarantee that the sequence generated by the concave–convex procedure can converge to a stationary point of the relaxed problem. Fourthly, another challenge task in the Lagrangian relaxation approach is to maximize the dual function effectively and efficiently (Bertsekas, 1999). Due to its low memory requirements, subgradient method is frequently used to optimize the dual function. However, zigzagging phenomena will occur for the standard subgradient method and the subgradient directions are obtained by fully optimizing the relaxed problem, which can influence the convergence speed. To improve the efficiency, we design an improved conditional surrogate subgradient algorithm to solve the LD problem and prove its convergence under some appropriate assumptions. Lastly, the solution obtained by solving the LD problem may not satisfy machine capacity constraint or operation precedence relationship constraint of the SCC scheduling problem, which means that it is not a feasible schedule. Hence, a heuristic algorithm is constructed to obtain a feasible schedule by adjusting appropriately the solutions of the relaxed problem.

The rest of the paper is organized as follows. In the next section, we give a brief description of the production process in steelmaking-continuous casting. In Section 3, we formulate a mixed integer nonlinear mathematical model for the SCC scheduling problem. In Section 4, an improved solution methodology is designed by combining Lagrangian relaxation, the concave–convex procedure, conditional surrogate subgradient algorithm and a simple heuristic algorithm to solve the SCC scheduling problem. Some numerical results are reported to show the efficiency and effectiveness of the conditional surrogate subgradient method in Section 5. In the last section, we make some conclusions about this study.

2. Process description of steelmaking-continuous casting

The SCC production is regarded as a bottleneck in the steel production because its production capacity is in general lower than that of hot rolling and cold rolling production (Tang and Wang, 2008). The main production process of SCC can be broadly classified into three stages: steelmaking, refining and continuous casting as illustrated in Fig. 1.

Steelmaking process is one of the most energy-intensive processes for producing molten steel from the main raw materials including iron ore and scrap on electric furnace or basic oxygen furnace. In steelmaking process, impurities such as nitrogen, silicon, phosphorus, sulfur and excess carbon are removed from the raw iron, and alloying elements such as manganese, nickel, chromium and vanadium are added to produce different grades of steel. The temperature of molten steel and the ingredients of alloy materials are adjusted to ensure the quality of the products cast from the liquid steel. On the same electric furnace or basic oxygen furnace, a group of molten iron is called as a charge, which is the basic unit of steelmaking process (Mao et al., 2014). In the refining process, the impurities are further eliminated and the required alloy ingredients are further added into molten steel to improve the quality of the molten steel. Continuous casting, is a main process of steelmaking-continuous casting whereby molten metal is solidified into qualified steel billet. At this stage, a sequence of charges must be continuously cast on the same continuous caster (Pan, 2016). Moreover, a sequence of charges processed on the same continuous caster is described as a cast that is a basic unit in the steelmaking-continuous casting production.

In the practical SCC process, the following general assumptions are usually viable for the scheduling problem.

(1) each stage usually consists of identical parallel machines;

(2) all the jobs must be processed in the same sequence without skipping;
(3) in the steelmaking and refining stages, the setup times are usually very short, which can be ignored (Pan et al., 2013). However, a relatively long sequence dependent setup time is considered between two casts to change equipment;
(4) a machine can at most process one job at a time;
(5) for the two consecutive operations of the same charge, only when the preceding operation has been finished, can the immediate next one be started;
(6) a job can at most be processed on one machine at any time;
(7) all the jobs must be processed continuously on the same batch, which should not be interrupted.

3. Mathematical formulation of the SCC scheduling problem

The following notations are adopted to establish a mathematical model for the SCC scheduling problem.

Sets and number of elements

j:	index of stage;
i, r:	charge index;
n:	cast index;
M_j:	number of the identical parallel machines in stage j;
Ω:	the set of all charges, $\lvert\Omega\rvert$ is the total number of charges;
Ω_n:	the set of all charges in the nth cast, $n = \{1, 2, \ldots, N\}$, where N is the total number of casts, $\Omega_{n_1} \bigcap \Omega_{n_2} = \emptyset$, $\Omega_{n_1} \cup \Omega_{n_2} \cup \ldots \cup \Omega_{n_n} = \Omega$, for all $n_1 \neq n_2 \in \{1, 2, \ldots, N\}$;
B_k:	the set of indices of all casts on the kth machine at the last stage;
$s(n)$:	index of the last job in the nth cast, $\Omega_n = \{s(n-1)+1, \ldots, s(n)\}$, $s(n) = s(n-1) + \lvert\Omega_n\rvert$, $s(0) = 0$, $s(N) = \lvert\Omega\rvert$;
$b(k)$:	index of the last cast on the machine k at the last stage, $b(k) = b(k-1) + \lvert B_k\rvert$, $b(0) = 0$, $b(M_S) = N$, $1 \le k \le M_S$, $B_k = \{b(k-1)+1, \ldots, b(k)\}$, where S is the total number of stages.

Parameters

P_{ij}:	processing time of the job i at the stage j;
$T_{j,j+1}$:	transportation time between the stage j and stage $j+1$;
d_n:	the predefined start time of a cast n in the stage of continuous casting;
Su_n:	the setup time between the adjacent casts Ω_n and Ω_{n+1} on the same machine in the last stage;
C_j:	penalty coefficient for the waiting time of any job from the stage j to stage $j+1$;
D_1:	penalty coefficient for the total of starting time before its predefined time in each cast;
D_2:	penalty coefficient for the total of starting time after its predefined time in each cast;

Decision variables

x_{ijk}:	0/1 variable, equal to 1 if and only if the job i is processed in the stage j on the kth machine;
y_{irk}:	0/1 variable, equal to 1 if and only if the job i precedes the job r processed on the same machine k;
t_{ij}:	starting time of the job i in the stage j.

In the SCC process, based on the customers' demand, the production planning is designed well in advance to determine the order of the tasks, the relation between charges and so on (Lin et al., 2016; Sun et al., 2015). Because of the complexity, the production planning and scheduling problem of the SCC process are usually treated sequentially. According to the results of the production planning system, scheduling problem of the SCC process is considered about how to determine which orders are allocated to each machine and assign the sequence of orders allocated at each stage. Thus, the sequences of charges in cast are known in advance for the SCC scheduling problem (Hao et al., 2015; Jiang et al., 2016; Mao et al., 2014; Pan, 2016). Consequently, combining the above notations with the assumptions described in Section 2, the mathematical model for the SCC scheduling problem can be formulated as follows:

Objective function:

$$\min G = G_1 + G_2 + G_3, \tag{1}$$

with

$$G_1 = \sum_{i=1}^{|\Omega|}\sum_{j=1}^{S-1} C_j(t_{i,j+1} - t_{i,j}), \tag{2}$$

$$G_2 = D_1 \sum_{n=1}^{N} \max\{0, d_n - t_{s(n-1)+1,S}\}, \tag{3}$$

$$G_3 = D_2 \sum_{n=1}^{N} \max\{0, t_{s(n-1)+1,S} - d_n\}. \tag{4}$$

Constraints:

$$t_{i,j+1} - t_{i,j} - P_{i,j} \ge T_{j,j+1}, \ \forall i \in \Omega, \quad \forall j \in \{1, 2, \ldots, S-1\}. \tag{5}$$

$$t_{i+1,S} = t_{i,S} + P_{i,S}, \ \forall i, i+1 \in \Omega_n, \quad \forall n \in \{1, 2, \ldots, N\}. \tag{6}$$

$$t_{i+1,S} - t_{i,S} - P_{i,S} \ge Su_n, \ i = s(n), \ \forall \Omega_n, \Omega_{n+1} \in B_k. \tag{7}$$

$$\sum_{k=1}^{M_j} x_{ijk} = 1, \ \forall i \in \Omega, \quad \forall j \in \{1, 2, \ldots, S-1\}. \tag{8}$$

$$y_{rik} + y_{irk} = x_{ijk}x_{rjk}, \ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{9}$$

$$y_{rik}(t_{i,j} - t_{r,j} - P_{r,j}) \ge 0, \ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{10}$$

$$t_{i,j} \ge 0, \forall i \in \Omega, \ \forall j \in \{1, 2, \ldots, S\}. \tag{11}$$

$$x_{ijk} \in \{0, 1\}, \ \forall i \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{12}$$

$$y_{irk} \in \{0, 1\}, \ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{13}$$

In the model formulated above, the objective functions G_1, G_2 and G_3 are to minimize the total waiting time of the jobs between the adjacent stages to reduce the temperature loss of molten steel (Jiang et al., 2015) and the earliness/tardiness of cast starting to achieve the goal of the just-in-time principle (Jiang et al., 2016; Tang et al., 2002), respectively. Constraint (5) ensures that there exists operation precedence constraint, that is, for the same job, the next operation can be started after finishing the preceding operation and transferring into the next operation. Constraint (6) enforces any two adjacent jobs must be processed continuously without any waiting time in the same batch. Constraint (7) implies there is a setup time between two adjacent batches on the same machine to change equipment in the last stage. Constraint (8) expresses that any job must be processed on one and only one machine at each stage. Constraint (9) guarantees there exists an operation precedence relationship for the jobs processed on the same machine. Constraint (10) assures there exists the machine capacity constraint, which means that a machine can process at most one job at a time. Constraints (11)–(13) make sure the range of the decision variable values.

In the above model, it is easy to see that the objective functions G_2 and G_3 are nonlinearity. In order to overcome such difficulties, let $t^l_{s(n-1)+1,S} = \max(0, d_n - t_{s(n-1)+1,S})$ and $t^u_{s(n-1)+1,S} = \max(0, t_{s(n-1)+1,S} - d_n)$, then the objective functions G_2 and G_3 can be reexpressed as follows

$$G_2 = D_1 \sum_{n=1}^{N} t^l_{s(n-1)+1,S}, \ G_3 = D_2 \sum_{n=1}^{N} t^u_{s(n-1)+1,S}. \tag{14}$$

Furthermore, the new variables $t^l_{s(n-1)+1,S}$ and $t^u_{s(n-1)+1,S}$ should satisfy the additional constraints (Tang et al., 2000), given as follows

$$t_{s(n-1)+1,S} = t^u_{s(n-1)+1,S} - t^l_{s(n-1)+1,S} + d_n,\ 1 \le n \le N. \tag{15}$$

$$t^u_{s(n-1)+1,S} \ge 0,\ t^l_{s(n-1)+1,S} \ge 0,\ 1 \le n \le N. \tag{16}$$

4. Solution methodology

4.1. Lagrangian relaxation

In the model formulated above, there exist different variables in the constraints (9) and (10), which make this model difficult to deal with. By relaxing the constraints (9) and (10), we can receive the relaxed problem, given as follows

$$L(\mu, \lambda) = \min(F_1 + F_2 + F_3 + F_4), \tag{17}$$

with

$$F_1 = \sum_{i=1}^{|\Omega|}\sum_{j=1}^{S-1} C_j(t_{i,j+1} - t_{i,j}), \tag{18}$$

$$F_2 = \sum_{i=1}^{N}(D_1 t^l_{s(n-1)+1,S} + D_2 t^u_{s(n-1)+1,S}), \tag{19}$$

$$F_3 = \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j}\mu_{irjk}(y_{rik} + y_{irk} - x_{ijk}x_{rjk}), \tag{20}$$

$$F_4 = -\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j}\lambda_{irjk}y_{rik}(t_{i,j} - t_{r,j} - P_{r,j}). \tag{21}$$

Subject to (5)–(8), (11)–(13), (15) and (16) and

$$\mu_{irjk} \in R,\ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{22}$$

$$\lambda_{irjk} \ge 0,\ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{23}$$

Here, μ and λ are the vectors of Lagrangian multipliers $\{\mu_{irjk}\}$ and $\{\lambda_{irjk}\}$ associated with the constraints (9) and (10), respectively. From the constraints (9) and (12)), it follows that

$$y_{irk} + y_{rik} \le 1,\ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{24}$$

Thus, we consider to add this new constraint (24) into the relaxed problem to improve the quality of the dual problem, which is defined by

$$\max_{\mu,\lambda} L(\mu, \lambda), \tag{25}$$

subject to (22) and (23).

Let $F(\mu, \lambda) = -L(\mu, \lambda)$, then the dual problem can be substituted by

$$\min_{\mu,\lambda} F(\mu, \lambda), \tag{26}$$

subject to (22) and (23).

4.2. Solution of the subproblems

For the given Lagrangian multipliers $\{\mu_{irjk}\}$ and $\{\lambda_{irjk}\}$, the relaxed problem can be decomposed into two subproblems, a mixed integer nonlinear programming problem and a 0–1 quadratic integer programming problem. Between them, the mixed integer nonlinear programming problem can be derived as follows

$$\min L_1 = F_1 + F_2 + F_4 + F_5, \tag{27}$$

where

$$F_5 = \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j}\mu_{irjk}(y_{rik} + y_{irk}), \tag{28}$$

subject to (5)–(7), (11), (13), (15), (16) and (24).

In the above subproblem, it is obvious to find that the Lagrange function F_4 is nonseparable, which makes this subproblem difficult to solve from a computational point of view. Recently, much effort has been done to deal with the nonseparability for augmented Lagrangian relaxation problems, such as, two-level optimization method (Li and Ierapetritou, 2010), auxiliary problem principle method (Beltran and Heredia, 2002), diagonal quadratic approximation method (Ding and Bie, 2017) and block coordinate descent method (Liu et al., 2010). In the current research, there is no convergence analysis for the two-level optimization method or auxiliary problem principle method. In the Lagrange relaxation approach, the convergence plays a crucial role in updating the Lagrangian multipliers effectively. As a result, these two methods are not available to solve this subproblem. Currently, many researchers have made a lot of effort on the convergence of the block coordinate descent method and diagonal quadratic approximation method. For the block coordinate descent method, its convergence can be established under the requirements that the minimizer is unique at each step or the objective function is quasi convex (Razaviyayn et al., 2013; Tseng, 2001). In contrast, the convergence of the diagonal quadratic approximation method can be guaranteed for convex programming problems (Ding and Bie, 2017). Unfortunately, for the above subproblem (27), it is difficult to satisfy the convergence requirements of the block coordinate descent method or diagonal quadratic approximation method, so these methods are still not suitable to solve this subproblem. Inspired by the above ideals, we design an approximate version of the Lagrange functions by taking full advantage of the characteristics of the relaxed problem. As a result, the mixed integer nonlinear programming problem can also be decomposed into two tractable subproblems by using the concave–convex procedure. Under some appropriate assumptions, we can provide a unified convergence analysis for the concave–convex procedure. Based on the formula (13), we have $y^2_{rik} = y_{rik}$, then it follows that

$$\begin{aligned} y_{rik}(t_{i,j} - t_{r,j} - P_{r,j}) \quad = \quad & \frac{1}{2}((t_{i,j} - t_{r,j} + y_{rik})^2 - y_{rik} \\ & -(t_{i,j} - t_{r,j})^2) - y_{rik}P_{r,j}. \end{aligned} \tag{29}$$

Thus, together with the above formula (29), the subproblem (27) can be transformed into the following problem

$$\min L_1 = F_1 + F_2 + F_5 + F_6 + F_7, \tag{30}$$

where

$$F_6 = \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j}\lambda_{irjk}\left(\frac{1}{2}y_{rik} + \frac{1}{2}(t_{i,j} - t_{r,j})^2 + y_{rik}P_{r,j}\right), \tag{31}$$

and

$$F_7 = -\frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j}\lambda_{irjk}(t_{i,j} - t_{r,j} + y_{rik})^2, \tag{32}$$

subject to (5)–(7), (11), (13), (15), (16) and (24).

Let $f_1 = F_1 + F_2 + F_5 + F_6$ and $f_2 = F_7$. If the functions f_1 and f_2 are continuous, then it is easy to prove that the functions f_1 and f_2 are convex and concave, respectively. Hence, we relax the restriction of the constraint (13) further by the following constraint

$$0 \le y_{irk} \le 1,\ \forall i \neq r \in \Omega, \quad 1 \le j < S, \quad 1 \le k \le M_j. \tag{33}$$

As a consequence, we obtain a relaxed problem of the subproblem L_1 that is derived as follows

$$\min L_1' = f_1 + f_2, \tag{34}$$

subject to (5)–(7), (11), (15), (16), (24) and (33).

Therefore, we know that the objective function L_1' is a DC function (Horst and Thoai, 1999). Besides, there is no doubt to see that all the constraints are linear equalities or inequalities, which implies that the problem (34) is a DC optimization problem (Horst et al., 1991). Due to its simple implementation and good convergence, the concave–convex procedure, an efficient iterative algorithm, has been extensively and successfully applied to solve DC optimization problems (Sriperumbudur and Lanckriet, 2012). In the concave–convex procedure, a sequence $\{t_{ij}^l, y_{irk}^l\}$ is generated by solving the following sequence of convex programs through linearizing the concave part (Lanckriet and Sriperumbudur, 2009; Razaviyayn et al., 2013), given as follows

$$(t_{ij}^{l+1}, y_{irk}^{l+1}) = arg\ \min(f_1 + f_3 + f_4 + R), \tag{35}$$

where

$$f_3 = \sum_{i=1}^{|\Omega|}\sum_{j=1}^{S-1} \nabla f_2(t_{ij}^l)(t_{ij} - t_{ij}^l), \tag{36}$$

$$f_4 = \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \nabla f_2(y_{irk}^l)(y_{irk} - y_{irk}^l), \tag{37}$$

and

$$R = f_2(t_{ij}^l, y_{irk}^l), \tag{38}$$

subject to (5)–(7), (11), (15), (16), (24) and (33).

In the formulas (36) and (37), the parameters $\nabla f_2(t_{ij}^l)$ and $\nabla f_2(y_{irk}^l)$ are calculated in the following ways

$$\nabla f_2(t_{ij}^l) = \sum_{r=1, r\neq i}^{|\Omega|}\sum_{k=1}^{M_j}((\lambda_{rijk} + \lambda_{irjk})(t_{rj}^l - t_{ij}^l) - \lambda_{irjk} y_{rik}^l + \lambda_{rijk} y_{irk}^l), \tag{39}$$

and

$$\nabla f_2(y_{irk}^l) = -\lambda_{rijk}(t_{rj}^l - t_{ij}^l + y_{irk}^l). \tag{40}$$

Thus, the minimization of the above problem (35) can be decomposed into two subproblems to update the sequences $\{t_{ij}^l\}$ and $\{y_{irk}^l\}$, respectively. The first subproblem can be determined by

$$t_{ij}^{l+1} = \underset{t_{ij}}{argmin}(F_1 + F_2 + f_3 + \frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \lambda_{irjk}(t_{ij} - t_{rj})^2), \tag{41}$$

subject to (5)–(7), (11), (15) and (16).

Obviously, it is easy to see that the subproblem (41) is a quadratic programming problem, which can be solved directly by using some standard software packages, such as, cplex.

The second subproblem can be used to generate the sequence $\{y_{irk}^l\}$, which is deemed by

$$y_{irk}^{l+1} = \underset{y_{irk}}{argmin}\left(F_5 + f_4 + \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \lambda_{irjk}\left(\frac{1}{2}y_{rik} + y_{rik}P_{r,j}\right)\right), \tag{42}$$

subject to (24) and (33).

Clearly, the above subproblem (42) is a simple linear programming problem. For the fixed multipliers μ_{irjk} and λ_{irjk}, it is easy to obtain its optimal solution, given as follows

$$y_{irk}^{l+1} = \begin{cases} 1, & \text{if } \bar{\lambda}_{irjk} \leq 0 \text{ and } \bar{\lambda}_{irjk} < \bar{\lambda}_{rijk}, \\ 0, & \text{otherwise,} \end{cases} \tag{43}$$

where

$$\bar{\lambda}_{irjk} = \lambda_{rijk}\left(\frac{1}{2} + P_{i,j}\right) + \mu_{irjk} + \mu_{rijk} + \nabla f_2(y_{irk}^l), \tag{44}$$

for all $i \neq r \in \Omega$, $1 \leq j < S$, $1 \leq k \leq M_j$.

According to the relaxed problem (17), the other subproblem, the 0–1 quadratic integer programming problem, can be written as follows

$$\min L_2 = -\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \mu_{irjk} x_{ijk} x_{rjk}, \tag{45}$$

subject to (8) and (12).

Based on the formula (12), we have $x_{rjk}^2 = x_{rjk}$ and $x_{ijk}^2 = x_{ijk}$. Therefore, we can obtain

$$\begin{aligned} -x_{ijk}x_{rjk} &= \frac{1}{2}(x_{ijk}^2 + x_{rjk}^2 - (x_{ijk} + x_{rjk})^2) \\ &= \frac{1}{2}(x_{ijk} + x_{rjk} - (x_{ijk} + x_{rjk})^2), \end{aligned} \tag{46}$$

and

$$\begin{aligned} -x_{ijk}x_{rjk} &= \frac{1}{2}((x_{ijk} - x_{rjk})^2 - x_{ijk}^2 - x_{rjk}^2) \\ &= \frac{1}{2}((x_{ijk} - x_{rjk})^2 - x_{ijk} - x_{rjk}). \end{aligned} \tag{47}$$

Let $\hat{\mu}_{irjk} = \min\{0, \mu_{irjk}\}$ and $\bar{\mu}_{irjk} = \max\{0, \mu_{irjk}\}$, then from (46) and (47), it follows that

$$\begin{aligned} &- \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \mu_{irjk} x_{ijk} x_{rjk} \\ &= -\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} (\hat{\mu}_{irjk} + \bar{\mu}_{irjk}) x_{ijk} x_{rjk} \\ &= \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \hat{\mu}_{irjk}\left(\frac{1}{2}((x_{ijk} - x_{rjk})^2 - x_{ijk} - x_{rjk})\right) \\ &+ \sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} \bar{\mu}_{irjk}\left(\frac{1}{2}(x_{ijk} + x_{rjk} - (x_{ijk} + x_{rjk})^2)\right) \\ &= \frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} (\bar{\mu}_{irjk} - \hat{\mu}_{irjk})(x_{ijk} + x_{rjk}) \\ &+ \frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} (\hat{\mu}_{irjk}(x_{ijk} - x_{rjk})^2 - \bar{\mu}_{irjk}(x_{ijk} + x_{rjk})^2). \end{aligned} \tag{48}$$

Based on the above formula (48), we have $L_2 = f_5 + f_6$, where the functions f_5 and f_6 are calculated by

$$f_5 = \frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} (\bar{\mu}_{irjk} - \hat{\mu}_{irjk})(x_{ijk} + x_{rjk}), \tag{49}$$

and

$$f_6 = \frac{1}{2}\sum_{i=1}^{|\Omega|}\sum_{r=1, r\neq i}^{|\Omega|}\sum_{j=1}^{S-1}\sum_{k=1}^{M_j} (\hat{\mu}_{irjk}(x_{ijk} - x_{rjk})^2 - \bar{\mu}_{irjk}(x_{ijk} + x_{rjk})^2). \tag{50}$$

To solve the subproblem (45), we consider to solve its relaxed problem, given as follows

$$\min L_2' = f_5 + f_6, \tag{51}$$

subject to (8) and

$$0 \leq x_{ijk} \leq 1, \ \forall i \in \Omega, \quad \forall j \in \{1, 2, \ldots, S-1\}. \tag{52}$$

Based on the definition of the problem L_2', it is not difficult to see that this problem is also a DC optimization problem. Thus, the problem L_2' can be equivalent to solving the following convex problem

$$x_{ijk}^{l+1} = \arg\min \left(f_5 + \sum_{i=1}^{|\Omega|} \sum_{j=1}^{S-1} \sum_{k=1}^{M_j} \nabla f_6(x_{ijk}^l)(x_{ijk} - x_{ijk}^l) + f_6(x_{ijk}^l) \right), \quad (53)$$

with

$$\nabla f_6(x_{ijk}^l) = \sum_{r=1, r\neq i}^{|\Omega|} ((\hat{\mu}_{irjk} + \hat{\mu}_{rijk})(x_{ijk}^l - x_{rjk}^l) - (\bar{\mu}_{irjk} + \bar{\mu}_{rijk})(x_{ijk}^l + x_{rjk}^l)), \quad (54)$$

subject to (8) and (52).

For the given multiplier μ_{irjk}, the optimal solution to the above problem (53) is listed as follows

$$x_{ijk}^{l+1} = \begin{cases} 1, & k = \arg\min_{1\le k\le M_j} \tilde{\mu}_{ijk}, \\ 0, & \text{otherwise}, \end{cases} \quad (55)$$

where

$$\tilde{\mu}_{ijk} = \frac{1}{2} \sum_{r=1, r\neq i}^{|\Omega|} (\bar{\mu}_{irjk} - \hat{\mu}_{irjk} + \bar{\mu}_{rijk} - \hat{\mu}_{rijk}) + \nabla f_6(x_{ijk}^l). \quad (56)$$

4.3. Convergence analysis of the concave–convex procedure

In the following, our aim is to prove that the sequence $\{t_{ij}^{l+1}, y_{irk}^{l+1}, x_{ijk}^{l+1}\}$ generated by the concave–convex procedure can converge to a stationary point of the relaxed problem under some reasonable assumptions (Razaviyayn et al., 2013), which are summarized as follows.

Assumption 4.1. Let the approximation function $f(.,.)$ of $g(.)$ satisfy the following conditions:

- $f(y, y) = g(y), \forall y \in X$,
- $f(x, y) \ge g(x), \forall x, y \in X$,
- $\nabla f(x, y;d)|_{x=y} = \nabla g(y;d), \forall d$ with $y + d \in X$,
- $f(x, y)$ is continuous in (x, y).

Proposition 4.1. *Assume that the variables $x = (t_{ij}, y_{irk})$ and $y = (t'_{ij}, y'_{irk})$ are any feasible solutions of the problem (34). If $g(y) = f_1(y) + f_2(y)$ and $f(x, y) = f_1(x) + \nabla f_2(t'_{ij})(t_{ij} - t'_{ij}) + \nabla f_2(y'_{irk})(y_{irk} - y'_{irk}) + f_2(y)$, then (A1), (A2), (A3) and (A4) in Assumption 4.1 hold.*

Proof. According to the definitions of the functions $g(y)$ and $f(x, y)$, it follows that

$$f(y, y) = g(y). \quad (57)$$

In the problem (34), the function f_2 is concave, then we have

$$f_2(x) \le f_2(y) + \nabla f_2(y)(x - y), \ \forall x, y \in X, \quad (58)$$

which means that

$$f(x, y) \ge g(x). \quad (59)$$

Obviously, the function $f(x, y)$ is continuously differentiable due to the constraint (33), which implies that the assumption A4 holds. Moreover, the directional derivative of f with respect to the variable t_{ij} and y_{irk} are determined by

$$\nabla f(t_{ij}) = \nabla f_1(t_{ij}) + \nabla f_2(t'_{ij}), \quad (60)$$

and

$$\nabla f(y_{irk}) = \nabla f_1(y_{irk}) + \nabla f_2(y'_{irk}). \quad (61)$$

Thus, it follows that

$$\nabla f(x, y; d)|_{x=y} = \nabla f_1(y; d) + \nabla f_2(y; d) = \nabla g(y; d). \quad (62)$$

Hence, the assumptions A1–A3 also hold due to the formulas (57), (58) and (62), respectively.

Theorem 4.1. *Suppose that Assumption 4.1 is satisfied, then every limit point of the sequence $\{t_{ij}^l, y_{irk}^l\}$ generated by (41) and (43) is a stationary point of the problem (34) for the fixed multipliers.*

Proof. The concrete proof procedures are similar to Theorem 1 (Razaviyayn et al., 2013).

Theorem 4.2. *For the given multipliers, let (t_{ij}^*, y_{irk}^*) be a limit point of the sequence $\{t_{ij}^l, y_{irk}^l\}$ generated by (41) and (43), then (t_{ij}^*, y_{irk}^*) is also a stationary point to the corresponding subproblem (27).*

Proof. Obviously, for the given multipliers, the sequence $\{y_{irk}^l\}$ generated by the formula (43) satisfies the constraint (13), which implies that the sequence $\{t_{ij}^l, y_{irk}^l\}$ generated by (41) and (43) is a feasible solution to the relaxed problem (34). Therefore, the limit point (t_{ij}^*, y_{irk}^*) of the sequence $\{t_{ij}^l, y_{irk}^l\}$ is also a feasible solution to the subproblem (27). Thus, we have

$$L_1'(\mu, \lambda) \ge L_1(\mu, \lambda). \quad (63)$$

Since the problem (34) is a relaxed problem to the problem (27), then we can obtain

$$L_1'(\mu, \lambda) \le L_1(\mu, \lambda). \quad (64)$$

Therefore, together with the formulas (63) and (64), we know that (t_{ij}^*, y_{irk}^*) is also a stationary point to the subproblem (27).

4.4. Conditional surrogate subgradient method

In the Lagrange relaxation framework, the major challenge is to optimize the dual function effectively and efficiently, which is nondifferentiable and concave (Bertsekas, 1999). Due to its low storage requirements and simple formulation, subgradient method has been successfully applied to solve large-scale nondifferentiable convex problem. For the traditional subgradient method, the subdifferential of F at $\eta = [\mu, \lambda]$ is defined as follows

$$\partial F(\eta) = \{g \in R^n | F(\eta_1) \ge F(\eta) + g^T(\eta_1 - \eta), \ \forall \eta_1 \in R^n\}. \quad (65)$$

In the traditional subgradient method, the subgradient directions $g(\mu)$ and $g(\lambda)$ are calculated by

$$g_{i,r,j,k}(\mu) = y_{rik}^* + y_{irk}^* - x_{ijk}^* x_{rjk}^*, \quad (66)$$

and

$$g_{i,r,j,k}(\lambda) = y_{rik}^*(t_{i,j}^* - t_{r,j}^* - P_{r,j}). \quad (67)$$

In the above formulas (66) and (67), the parameters $t_{i,j}^*$, y_{irk}^* and x_{ijk}^* stand for the optimal solution of the relaxed problem.

Based on Theorem 4.2, it is easy to obtain the optimal solution $(t_{i,j}^*, y_{irk}^*, x_{ijk}^*)$ of the relaxed problem by solving the subproblems (41), (43) and (55), respectively. However, it may increase computational cost to obtain an optimal solution of the relaxed problem, especially for the large-scale optimization problems. In order to reduce the calculation time, the surrogate subgradient method is introduced to solve the LD problem, which does not require the relaxed problem to be fully optimized (Bragin et al., 2015). For the given multipliers μ_m and λ_m, let $x^l = (t_{i,j}^l, y_{irk}^l, x_{ijk}^l)$ and

$x^{l+1} = (t_{i,j}^{l+1}, y_{irk}^{l+1}, x_{ijk}^{l+1})$ be any feasible solutions of the relaxed problem, a simple surrogate optimality condition is required to satisfy (Bragin et al., 2015), given as follows

$$L(\mu_m, \lambda_m, x^{l+1}) < L(\mu_m, \lambda_m, x^l), \quad (68)$$

then the direction $g_m = [g_m^{\mu}, g_m^{\lambda}]$ is a surrogate subgradient of $F(\mu, \lambda)$ at $\eta_m = [\mu_m, \lambda_m]$, which satisfies the definition of subgradients (65), where $g_m^{\mu} = [g_{1,2,1,1}^m(\mu), g_{1,3,1,1}^m(\mu), \ldots, g_{|\Omega|,|\Omega-1|,S-1,M_{S-1}}^m(\mu)]$ and $g_m^{\lambda} = [g_{1,2,1,1}^m(\lambda), g_{1,3,1,1}^m(\lambda), \ldots, g_{|\Omega|,|\Omega-1|,S-1,M_{S-1}}^m(\lambda)]$ are computed by

$$g_{i,r,j,k}^m(\mu) = y_{rik}^{l+1} + y_{irk}^{l+1} - x_{ijk}^{l+1} x_{rjk}^{l+1}, \quad (69)$$

and

$$g_{i,r,j,k}^m(\lambda) = y_{rik}^{l+1}(t_{i,j}^{l+1} - t_{r,j}^{l+1} - P_{r,j}). \quad (70)$$

In the above surrogate subgradient method, the zigzagging of multipliers may occur due to the fact that the subgradients are almost antiparallel to the normals of the feasible set (D'Antonio and Frangioni, 2009). To avoid the zig-zagging phenomenon, we design a conditional surrogate subgradient algorithm to solve the LD problem, where the conditional surrogate subgradient directions are calculated by

$$d_m^{\mu} = \begin{cases} g_m^{\mu}, & m = 0, \\ g_m^{\mu} + \beta_m^{\mu} d_{m-1}^{\mu}, & m \geq 1, \end{cases} \quad (71)$$

and

$$d_m^{\lambda} = \begin{cases} \hat{g}_m^{\lambda}, & m = 0, \\ \hat{g}_m^{\lambda} + \beta_m^{\lambda} d_{m-1}^{\lambda}, & m \geq 1, \end{cases} \quad (72)$$

with

$$\beta_m^{\mu} = \frac{2\|g_m^{\mu}\|^2}{\|d_{m-1}^{\mu}\|\|g_m^{\mu}\| + |g_m^{\mu T} d_{m-1}^{\mu}|}, \quad (73)$$

and

$$\beta_m^{\lambda} = \frac{2\|\hat{g}_m^{\lambda}\|^2}{\|d_{m-1}^{\lambda}\|\|\hat{g}_m^{\lambda}\| + |\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}|}. \quad (74)$$

The specific iterative steps of the proposed conditional surrogate subgradient algorithm are similar to those of the improved subgradient method (Mao et al., 2014), which can be described as follows:

Step 1: Initialization.

Select $\varepsilon_1 > 0, \varepsilon_2 > 0, M > 0, 0 < p < 1, \delta > 0, \delta_{l_1} > 0$ (δ_{l_1} denotes the reduction corresponding to the estimation level F_{lev}^m), $\sigma_{max} > 0$ (σ_{max} denotes the threshold for adjusting δ_{l_1}), $W > 0$ (W is an even number), $t \in (0, 1)$. Given an initial feasible solution $x^0 = (t_{i,j}^0, y_{irk}^0, x_{ijk}^0)$ of the relaxed problem. Set $F_{best}^0 = \infty$, $P_{best}^0 = \infty$, the initial accumulated path $\sigma_1 = 0$, the iteration number $m = 1$, the weak-oscillation number $r = 0$, the strong-oscillation number $s = 0$, $l_1 = 1$, $M[l_1] = 1$ ($M[l_1]$ will denote the iteration number when the l_1th update of F_{best}^m occurs). Denote $h(s) = 1/(s+1)$, which is used to adjust the threshold σ_{max}.

Step 2: Function evaluation.

- If $F(\mu_m, \lambda_m) < F_{rec}^{m-1}$, set $F_{rec}^m = F(\mu_m, \lambda_m)$ $\mu_{best} = \mu_m$ and $\lambda_{best} = \lambda_m$; otherwise, set $F_{rec}^m = F_{rec}^{m-1}$, where $F_{rec}^m = \min\{F(\mu_j, \lambda_j) | 0 \leq j \leq m\}$.
- If $P_{best} < P(\mu_m, \lambda_m)$, then set $P_{best} = P(\mu_m, \lambda_m)$.
- If $P_{best} + F(\mu_m, \lambda_m) < \delta_{l_1}$, then set $\delta_{l_1} = P_{best} + F(\mu_m, \lambda_m)$.

Step 3: Sufficient descent.

If $F(\mu_m, \lambda_m) \leq F_{rec}^m - 0.5\delta_{l_1}$, then set $M[l_1 + 1] = m, \sigma_m = 0, \delta_{l_1+1} = \min\{\beta_{l_1}\delta_{l_1}, \delta\}, \beta_{l_1} = 1 + 1/Ml_1^p, l_1 = l_1 + 1$.

Step 4: Weak oscillation detection.

Let $D_r = F(\mu_m, \lambda_m), r = r + 1$. If $r > W$ and $\Sigma_{n=1}^{W-2} |\frac{D[n+2]-D[n]}{D[n]}| \frac{2}{W} < \varepsilon_2$, set $m_1 = m + 1 + \lfloor (h(l_1)\sigma_{\max} - \sigma_m + 1)/\|d_m\| \rfloor, \sigma_{m_1} = h(l_1)\sigma_{\max} + 1, F_{rec}^{m_1} = F_{rec}^m, \mu_{m_1} = \mu_m, \lambda_{m_1} = \lambda_m, d_{m_1} = d_m$ and $m = m_{[1]}$. If $r > W$, set $r = 0$.

Step 5: Strong oscillation detection.

If $\sigma_m > h(s)\sigma_{max}$, then $M[l_1 + 1] = m, \sigma_m = 0, \delta_{l_1+1} = \beta_{l_1}\delta_{l_1}, \beta_{l_1} = 1 - 1/Ml_1^p, l_1 = l_1 + 1, s = s + 1$.

Step 6: Calculation of the subgradient and multipliers.

Set $F_{lev}^m = F_{rec}^{M[l_1]} - \delta_{l_1}$ and $d_m = [d_m^{\mu}, d_m^{\lambda}]$. The step size α_m is calculated in the following way

$$\alpha_m = t(F(\mu_m, \lambda_m) - F_{lev}^m)/\|d_m\|^2, \quad (75)$$

then the multipliers μ_{m+1} and λ_{m+1} are updated by

$$\mu_{m+1} = \mu_m - \alpha_m d_m^{\mu}, \quad (76)$$

and

$$\lambda_{m+1} = P_{\Phi}(\lambda_m - \alpha_m d_m^{\lambda}). \quad (77)$$

If the formula (68) holds or $\|x^{l+1} - x^l\| < \varepsilon_1$, then the subgradient directions d_m^{μ} and d_m^{λ} are computed by (71) and (72). Otherwise, set $x_{ijk}^l = x_{ijk}^{l+1}$, $y_{irk}^l = y_{irk}^{l+1}$ and $t_{ij}^l = t_{ij}^{l+1}$, resolve the subproblems (41), (43) and (55) to generate new feasible solution $x^{l+1} = (t_{i,j}^{l+1}, y_{irk}^{l+1}, x_{ijk}^{l+1})$ of the relaxed problem, until formula (68) holds or $\|x^{l+1} - x^l\| < \varepsilon_1$, where $P_{\Phi}(x) = \operatorname{argmin}_{y \in \Phi} \|y - x\|^2$.

Step 7: Path update.

Let $\sigma_{m+1} = \sigma_m + \|\alpha_m d_m\|$, $m = m + 1$.

Step 8: Termination check.

If $\alpha_m < \varepsilon_1$ or $|\delta_{l_1}/F_{best}^{M[l_1]}| < \varepsilon_1$, then stop; otherwise, go to the Step 2.

In Step 6, the vector $\hat{g}_m^{\lambda}$ is calculated as follows (Wang, 2003)

$$\hat{g}_m^{\lambda} = \begin{cases} 0, & g_{m,i}^{\lambda} \geq 0 \text{ and } \lambda_{m,i} = 0, \\ g_m^{\lambda}, & \text{otherwise}. \end{cases} \quad (78)$$

Note 1. Different from the iterative steps of the improved subgradient method (Mao et al., 2014), Steps 3 and 5 are also modified to adjust the estimation level F_{lev}^m effectively. In the Step 3, if $F(\mu_m, \lambda_m) \leq F_{rec}^m - 0.5\delta_{l_1}$, then from Steps 2 and 6, we know that the estimation level F_{lev}^m can be further improved by enlarging the reduction δ_{l_1}. By contract, the adaptive strategy of the parameter β_{l_1} is designed in Step 5 to prevent the reduction δ_{l_1} converging to zero prematurely.

Note 2. In Step 6, if the condition (68) holds, then the directions calculated by (69) and (70) are the surrogate subgradient directions obtained by approximately solving the relaxed problem. Otherwise, if the condition (68) do not hold, then the condition $\|x^{l+1} - x^l\| < \varepsilon_1$ is satisfied, which means that the solution x^l converge to a limit point of the relaxed problem based on Proposition 4.1, which means that the relaxed problem is solved optimally. At this time, the directions calculated by (69) and (70) become the traditional subgradient directions.

From (72) and (74), we have

$$\begin{aligned} d_m^{\lambda T} d_{m-1}^{\lambda} &= \hat{g}_m^{\lambda T} d_{m-1}^{\lambda} + \beta_m^{\lambda}\|d_{m-1}^{\lambda}\|^2 \\ &= \hat{g}_m^{\lambda T} d_{m-1}^{\lambda} + \frac{2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|^2}{\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\| + |\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}|} \\ &= \frac{\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}(\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\| + |\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}|) + 2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|^2}{\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\| + |\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}|} \\ &\geq \frac{-2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|^2 + 2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|^2}{\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\| + |\hat{g}_m^{\lambda T} d_{m-1}^{\lambda}|} \\ &\geq 0. \end{aligned} \quad (79)$$

Similarly, from (71) and (73), it follows that

$$d_m^{\mu T} d_{m-1}^{\mu} \ge 0, \quad (80)$$

which implies any two consecutive surrogate subgradient directions generated by the above algorithm form an acute angle. Therefore, this algorithm can avoid the zig-zagging phenomenon (D'Antonio and Frangioni, 2009; Larsson et al., 1996).

4.5. Convergence analysis of the conditional surrogate subgradient method

In this subsection, let $\eta=[\mu,\lambda]$, $\hat{\eta}=[\hat{\mu},\hat{\lambda}]$, $\eta_0=[\mu_0,\lambda_0]$, $\bar{\eta}_1=[\bar{\mu}_1,\bar{\lambda}_1]$, $\eta_1=[\mu_1,\lambda_1]$, $\beta_m=[\beta_m^{\mu},\beta_m^{\lambda}]$, $d_m=[d_m^{\mu},d_m^{\lambda}]$, $g_m=[g_m^{\mu},g_m^{\lambda}]$ and $\hat{g}_m=[g_m^{\mu},\hat{g}_m^{\lambda}]$, then the convergence of the conditional surrogate subgradient method can be established under the following assumption.

Assumption 4.2. There exists a positive scalar $C>0$, such that

$$\|g_{\mu},g_{\lambda}\| \le C, \quad (81)$$

where $[g_{\mu},\ g_{\lambda}]\in\partial F^{\Phi}(\eta)=\{g\in R^n|F(\eta_1)\ge F(\eta)+g^T(\eta_1-\eta),\ \forall\,\eta_1\in\Phi\}$, and $\partial F^{\Phi}(\eta)$ is the set of all conditional subgradients of F at η.

Lemma 4.1. *Suppose that the directions d_m^{μ} and d_m^{λ} are generated by* (71) and (72)*, then the Assumption* 4.2 *holds.*

Proof. From the formula (72), it follows that

$$\begin{aligned}\|d_m^{\lambda}\| &\le \|\hat{g}_m^{\lambda}\|+|\beta_m^{\lambda}|\|d_{m-1}^{\lambda}\|\\ &\le \|\hat{g}_m^{\lambda}\|+2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|/(\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\|+|\hat{g}_m^{\lambda T}d_{m-1}^{\lambda}|)\\ &\le \|\hat{g}_m^{\lambda}\|+2\|\hat{g}_m^{\lambda}\|^2\|d_{m-1}^{\lambda}\|/(\|\hat{g}_m^{\lambda}\|\|d_{m-1}^{\lambda}\|)\\ &= \|\hat{g}_m^{\lambda}\|+2\|\hat{g}_m^{\lambda}\|\\ &\le 3\|\hat{g}_m^{\lambda}\|.\end{aligned} \quad (82)$$

Let $H=\sum_{i\in\Omega}(\sum_{j=1}^{S}P_{ij}+\sum_{j=1}^{S-1}T_{j,j+1})$, which is the upper bound of the starting time of each job Mao et al. (2014). Based on the formula (70), we can obtain $\|g_m^{\lambda}\|\le 3H|\Omega||\Omega-1|(S-1)M_{S-1}$. Hence, from (78) and (82), we have

$$\|d_m^{\lambda}\|\le 3\|\hat{g}_m^{\lambda}\|\le 3\|g_m^{\lambda}\|\le 9H|\Omega||\Omega-1|(S-1)M_{S-1}. \quad (83)$$

In the similar way, it follows that

$$\|d_m^{\mu}\|\le 3\|g_m^{\mu}\|\le 9|\Omega||\Omega-1|(S-1)M_{S-1}. \quad (84)$$

Thus, set $C=(9+9H)|\Omega||\Omega-1|(S-1)M_{S-1}$, it follows that

$$\|d_m\|=\|d_m^{\mu},d_m^{\lambda}\|\le\|d_m^{\mu}\|+\|d_m^{\lambda}\|\le C. \quad (85)$$

This completes the proof.

Lemma 4.2. Projection theorem: *Given some $\eta\in R^n$, a vector $\bar{\eta}\in\Phi$ is equal to $P_{\Phi}(\eta)$ if and only if*

$$(\eta_1-\bar{\eta})^T(\eta-\bar{\eta})\le 0,\ \forall\ \ \eta_1\in\Phi. \quad (86)$$

Proof. Lemma 4.2 can be proved according to Propositions 5.1.2 and B.11 (Bertsekas, 1999).

Lemma 4.3. *Given some vector $\eta_0\in\Phi$, $d\in R^n$, α is positive scalar and $\bar{\eta}_1=\eta_0+\alpha d$. Let $\eta_1=P_{\Phi}(\bar{\eta}_1)$ and $\bar{g}=\eta_1-\bar{\eta}_1$, then we have*

$$\bar{g}^T d\ge 0. \quad (87)$$

Proof. Based on the norm definition, we have

$$\begin{aligned}\|\bar{\eta}_1-\eta_0\|^2 &= \|\bar{\eta}_1-\eta_1+\eta_1-\eta_0\|^2\\ &= \|\bar{\eta}_1-\eta_1\|^2+\|\eta_1-\eta_0\|^2+2(\bar{\eta}_1-\eta_1)(\eta_1-\eta_0),\end{aligned} \quad (88)$$

and

$$\begin{aligned}\|\eta_1-\eta_0\|^2 &= \|\eta_1-\bar{\eta}_1+\bar{\eta}_1-\eta_0\|^2\\ &= \|\eta_1-\bar{\eta}_1\|^2+\|\bar{\eta}_1-\eta_0\|^2+2(\eta_1-\bar{\eta}_1)(\bar{\eta}_1-\eta_0).\end{aligned} \quad (89)$$

Add these two formulas, we have

$$\|\bar{\eta}_1-\eta_1\|^2+(\bar{\eta}_1-\eta_1)(\eta_1-\eta_0)+(\eta_1-\bar{\eta}_1)(\bar{\eta}_1-\eta_0)=0. \quad (90)$$

Together with Lemma 4.2, we have

$$(\bar{\eta}_1-\eta_1)(\eta_1-\eta_0)\ge 0, \quad (91)$$

then from (90), it follows that

$$g^T d=-\frac{1}{\alpha}(\eta_1-\bar{\eta}_1)(\bar{\eta}_1-\eta_0)\ge 0. \quad (92)$$

Lemma 4.4. *Suppose that the directions d_m^{μ} and d_m^{λ} are generated by* (71) and (72)*. Let $\hat{\eta}$ and η_m be such that $F(\eta_m)>F(\hat{\eta})$. If the step size α_m is updated by* (75)*, then we have*

$$F(\hat{\eta})\ge F(\eta_m)+d_m^T(\hat{\eta}-\eta_m), \quad (93)$$

and

$$\|\eta_{m+1}-\hat{\eta}\|^2\le\|\eta_m-\hat{\eta}\|^2-2\alpha_m(F(\eta_m)-F(\hat{\eta}))+\alpha_m^2\|d_m\|^2. \quad (94)$$

Proof. From the definition of the direction $\hat{g}_m$, it is easy to obtain

$$g_m^T(\hat{\eta}-\eta_m)\ge\hat{g}_m^T(\hat{\eta}-\eta_m),\ \forall m\ge 0. \quad (95)$$

Assume that

$$\hat{g}_m^T(\hat{\eta}-\eta_m)\ge d_m^T(\hat{\eta}-\eta_m),\ \forall m\ge 0. \quad (96)$$

Clearly, from (71) and (72), the conclusion (96) holds for $m=0$. Thus, from (65) and (95), it follows that

$$F(\hat{\eta})-F(\eta_0)\ge g_0^T(\hat{\eta}-\eta_0)\ge d_0^T(\hat{\eta}-\eta_0). \quad (97)$$

Now, assume that the formula (96) holds for m, then together (65) and (95), we have

$$F(\hat{\eta})-F(\eta_m)\ge g_m^T(\hat{\eta}-\eta_m)\ge d_m^T(\hat{\eta}-\eta_m). \quad (98)$$

Due to the formula (75), we have

$$\alpha_m\|d_m\|^2\le F(\eta_m)-F(\hat{\eta}). \quad (99)$$

From (98) and (99), we have

$$d_m^T(\hat{\eta}-\eta_m)+\alpha_m\|d_m\|^2\le 0. \quad (100)$$

Take $\bar{g}_m=P_{\Phi}(\eta_m-\alpha_m d_m)-(\eta_m-\alpha_m d_m)$, from Lemma 4.3, we have

$$\begin{aligned}d_m^T(\hat{\eta}-\eta_{m+1}) &= d_m^T(\hat{\lambda}-P_{\Phi}(\lambda_m-\alpha_m d_m))\\ &= d_m^T(\hat{\eta}-\eta_m+\alpha_m d_m-\bar{g}_m)\\ &= d_m^T(\hat{\eta}-\eta_m)+\alpha_m\|d_m\|^2-\bar{g}_m^T d_m\\ &\le d_m^T(\hat{\eta}-\eta_m)+\alpha_m\|d_m\|^2.\end{aligned} \quad (101)$$

Therefore, from (100) and (101), we can obtain

$$d_m^T(\hat{\eta}-\eta_{m+1})\le 0. \quad (102)$$

Owing to the formulas (73) and (74), it follows that

$$\beta_{m+1}\ge 0. \quad (103)$$

Multiplying d_{m+1} by the vector $(\hat{\eta}-\eta_{m+1})$, we have

$$\begin{aligned}d_{m+1}^T(\hat{\eta}-\eta_{m+1}) &= \hat{g}_{m+1}^T(\hat{\eta}-\eta_{m+1})+\beta_{m+1}d_m^T(\hat{\eta}-\eta_{m+1})\\ &\le \hat{g}_{m+1}^T(\hat{\eta}-\eta_{m+1}),\end{aligned} \quad (104)$$

which means that the formula (96) holds for $m+1$, so the conclusion (98)) is also true for $m+1$. Hence, the direction d_m is a conditional subgradient of F at η_m. Then, it follows that

$$\begin{aligned}\|\hat{\eta}-\eta_{m+1}\|^2 &= \|\hat{\eta}-P_\Phi(\eta_m-\alpha_m d_m)\|^2\\ &\le \|\hat{\eta}-(\eta_m-\alpha_m d_m)\|^2\\ &= \|\hat{\eta}-\eta_m\|^2-2\alpha_m d_m^T(\eta_m-\hat{\eta})+\alpha_m^2\|d_m\|^2\\ &\le \|\hat{\eta}-\eta_m\|^2-2\alpha_m(F(\eta_m)-F(\hat{\eta}))+\alpha_m^2\|d_m\|^2.\end{aligned} \tag{105}$$

Lemma 4.5. *Assume that the step size α_m is computed by (75), such that*

$$\alpha_m>0,\ \sum_{m=0}^{\infty}\alpha_m=\infty,\ \sum_{m=0}^{\infty}\alpha_m^2<\infty. \tag{106}$$

Thus, the sequence $\{\mu_m, \lambda_m\}$ updated by the conditional surrogate subgradient algorithm can converge to a limit point of the Lagrangian dual problem.

Proof. This proof is similar to the proof of the related literatures (Bertsekas, 2011; Larsson et al., 1996).

Theorem 4.3. *For the conditional surrogate subgradient algorithm, we have*

$$\inf_{m\ge 0}F(\eta_m)=F^*. \tag{107}$$

Proof. Assume that

$$\lim_{l_1\to\infty}\delta_{l_1}>0, \tag{108}$$

from Step 3, it is not difficult to know that

$$\inf_{m\ge 0}F(\eta_m)=-\infty, \tag{109}$$

then the conclusion can be obtained.

$$\lim_{l_1\to\infty}\prod_{i=1}^{i=l}\beta_{l_1}=0. \tag{110}$$

Therefore, the cardinality of L is infinite.

$$\sigma_{m+1}=\sigma_m+\|\alpha_m d_m\|=\sum_{i=M[l]}^{m}\|\alpha_i d_i\|, \tag{111}$$

so that $M[l_1+1]=m+1$ and $l_1+1\in L$.
Thus, from Step 5 and (111), we have

$$\sum_{i=M[l_1]}^{m+1}\|\alpha_i d_i\|>h(s)\sigma_{\max}. \tag{112}$$

Hence, based on Lemma 4.1, we have

$$\sum_{i=M[l_1]}^{m+1}\alpha_i>h(s)\sigma_{\max}/C. \tag{113}$$

Owing to Step 5, it follows that

$$\sum_{l_1\in L}h(l_1)=\sum_{s=1}^{\infty}\frac{1}{s+1}, \tag{114}$$

then we have

$$\sum_{i=M[l_1]}^{\infty}\alpha_i\ge\sum_{l_1\ge\hat{l}_1,l_1\in L}\ \sum_{i=M[\hat{l}_1]}^{M[l_1+1]-1}\alpha_i>\sum_{s=\hat{s}}^{\infty}\frac{\sigma_{\max}}{C(s+1)}=\infty. \tag{115}$$

Now, to arrive at a contradiction, assume that

$$\inf_{m\ge 0}F(\eta_m)>F^*, \tag{116}$$

so that for some $\hat{\eta}\in\Phi$ and some $\varepsilon>0$, it follows that

$$\inf_{m\ge 0}F(\eta_m)-\varepsilon\ge F(\hat{\eta}). \tag{117}$$

Because $F(\eta)$ is continuous over Φ and $\delta_{l_1}\to 0$, there is a sufficiently large $\hat{l}_1$ such that $\delta_{\hat{l}_1}\le\varepsilon$, then we have

$$F_{lev}^m=F_{rec}^{M[l_1]}-\delta_{l_1}\ge\inf_{m\ge 0}F(\eta_m)-\varepsilon\ge F(\hat{\eta}),\ \forall m\ge M[\hat{l}_1]. \tag{118}$$

Using this relation and Lemma 4.4, for $\eta=\hat{\eta}$ and $\alpha_m=t(F(\eta_m)-F_{lev}^m)/\|d_m\|^2$, we obtain

$$\begin{aligned}\|\hat{\eta}-\eta_{m+1}\|^2 &\le \|\hat{\eta}-\eta_m\|^2-2\alpha_m(F(\eta_m)-F(\hat{\eta}))+\alpha_m^2\|d_m\|^2\\ &\le \|\hat{\eta}-\eta_m\|^2-2\alpha_m(F(\eta_m)-F_{lev}^m)+\alpha_m^2\|d_m\|^2\\ &\le \|\hat{\eta}-\eta_m\|^2-t(2-t)(F(\eta_m)-F_{lev}^m)^2/\|d_m\|^2\\ &\le \|\hat{\eta}-\eta_m\|^2-(2-t)\alpha_m^2\|d_m\|^2/t.\end{aligned} \tag{119}$$

By summing these inequalities over $m\ge M[\hat{l}_1]$, we have

$$(2-t)/t\sum_{m=M[\hat{l}_1]}^{\infty}\alpha_m^2\|d_m\|^2\le\|\eta_{M[\hat{l}_1]}-\eta\|^2. \tag{120}$$

If $\|d_m\|\to 0$, we know that $F(\eta_m)\to F^*$, because $F(\eta_m)$ is a convex function; otherwise, according to the formula (120), we have

$$\sum_{m=0}^{\infty}\alpha_m^2<\infty. \tag{121}$$

Because $\alpha_m>0$ and $\sum_{m=0}^{\infty}\alpha_m=\infty$, according to Lemma 4.5, the conclusion can be proved.

4.6. Constructing feasible solutions to the primal problem

In general, the solution of the relaxed problem is not a feasible solution to the primal problem, which means that machine capacity constraint or the operation precedence relationship constraint is violated. In order to obtain a better feasible solution of the primal problem, we design an effective heuristic method to construct a feasible scheduling, which are listed as follows:

- Set the solution $\{t_{i,j}\}$ of the relaxed problem as an initial list. Let $x_{ijk}=0(i\in\Omega, 1\le j<S, 1\le k\le M_j)$. $j=1$.
- According to the initial list $\{t_{i,j}\}$, a nondecreasing sequence T_j can be obtained. If the adjacent elements are equal in the sequence T_j, exchanging the positions of the two elements when the corresponding multipliers of the latter element is larger than the previous one. Let $n=1$, $T_{j,k}=0$, $J_{j,k}=\emptyset$, $|J_{j,k}|=0$, $1\le k\le M_j$.
- Take $T_j(n)$ as the nth element of the sequence T_j. Set $i=arg_{i\in\Omega}\{t_{i,j}=t_{T_j(n),j}\}$, the job i will process on the machine k^*, i.e, $x_{ijk^*}=1$, where k^* is calculated in the following way:

$$k^*=arg\min_{1\le k\le M_j}(T_{j,k}-t_{i,j}), \tag{122}$$

with

$$T_{j,k^*}=\max\{T_{j,k^*},t_{i,j}\}+P_{i,j}. \tag{123}$$

- Substitute the variable x_{ijk} obtained from Step 3 into the constraint (9), it follows that

Table 1
The numerical results of CSS and ISS about the average, duality gaps, running time and iteration (Svs.M = 3vs.4).

Bvs.J	Duality gaps (%)		Running time (s)		Iteration	
	CSS	ISS	CSS	ISS	CSS	ISS
4vs.11	2.384	2.384	10.33	28.2	58	194
4vs.12	2.834	2.834	13.08	37.44	61	215
4vs.13	2.918	2.918	15.73	45.44	62	214
4vs.14	**3.111**	3.112	22.11	59.89	69	249
4vs.15	**2.5**	2.517	22.57	61.97	66	221
4vs.16	**2.617**	2.62	28.56	104.39	70	308
4vs.17	**2.829**	2.831	27.4	98.01	62	266
4vs.18	**2.781**	2.818	31.05	85.86	63	212
4vs.19	2.243	2.243	32.73	92.74	60	208
4vs.20	3.133	3.133	36.81	106.14	60	211
5vs.11	**2.58**	2.614	18.1	73.84	64	319
5vs.12	2.706	2.706	21.39	70.9	65	261
5vs.13	**2.884**	2.929	25.06	69.19	64	216
5vs.14	**2.589**	2.592	29.87	81.14	64	213
5vs.15	**2.217**	2.222	35.32	93.37	66	210
5vs.16	**2.302**	2.303	44.55	121.97	65	214
5vs.17	2.197	2.197	44.79	120.94	64	208
5vs.18	**1.914**	1.915	45.46	124.77	61	203
5vs.19	2.186	2.186	48.6	135.52	60	203
5vs.20	**2.339**	2.34	54.39	153.04	59	202
Avg.	**2.5632**	2.5707	30.4	88.24	63	227

The bold values represent that the results of CSS method are better than those of ISS method in terms of the duality gaps.

$$y_{irk}+y_{rik}=\begin{cases}1, & x_{ijk}x_{rjk}=1,\\ 0, & \text{otherwise.}\end{cases} \tag{124}$$

Combining with the solution $\{t_{i,j}\}$ of the relaxed problem, the variable y_{irk} can be determined by

$$y_{irk}=\begin{cases}1, & \text{if } t_{ij}\le t_{rj} \text{ and } x_{ijk}x_{rjk}=1,\\ 0, & \text{otherwise.}\end{cases} \tag{125}$$

For all $i \neq r \in \Omega$.

- Let $J_{j,k^*}=J_{j,k^*}\bigcup\{i\}$, then $|J_{j,k^*}|=|J_{j,k^*}|+1$, $n=n+1$. If $n\le|\Omega|$, go to Step 3; otherwise, let $j=j+1$, go to the next Step.
- If $j<S$, then go to Step 2; otherwise, go to the next Step.
- Substitute $\{y_{irk}\}$ into the formula (10), then the primal model is a linear programming model. Finally, the algorithm can be terminated.

5. Numerical experiments

In this section, some numerical results are provided to test the performance of the proposed method. In our experiment, all algorithms are implemented in C# language and run on PC with Intel Core i7-4770 3.4 GHz CPU, and windows 10 operation system (64 bit). In all algorithms, upper bounds, lower bounds, duality gaps, iteration numbers and running time are used to evaluate the performance of the conditional surrogate subgradient method by comparing with the improved surrogate subgradient method that is a combination of surrogate strategy and improved subgradient level method (Mao et al., 2014). For the improved subgradient level method, it is an efficient and effective method to solve the LD problem, which has been successfully applied to solve the SCC scheduling problems (Mao et al., 2014, 2015). For simplicity, the conditional surrogate subgradient method and improved surrogate subgradient method are named by CSS and ISS, respectively. For each instance, the lower bound (LB) and the upper bound (UB) are obtained from the LD problem and the constructive heuristic method, respectively. In this paper, all linear programming problems and quadratic programming problems are solved by using the standard software package (cplex 12.6) and the duality gap (Millar and Yang, 1994; Wolosewicz et al., 2015) is calculated by

$$\text{duality gap}=\frac{UB-LB}{UB+LB}\times 200\%. \tag{126}$$

The parameters adopted in the surrogate subgradient algorithms are: the stopping criterion with $\varepsilon_1=1e-5$; the initial parameters $\varepsilon_2=1e-3$, $\delta_1=(P(\eta_0)+F(\eta_0))/5$, $U=\sum_{i\in\Omega}(\sum_{j=1}^{S}P_{ij}+\sum_{j=1}^{S-1}T_{j,j+1})$, $n=\lceil lg\|g(\eta_0)\|^2/U\rfloor$, $\lambda_1=10^n g_\lambda(0)/\|g_\lambda(0)\|^2$, $\mu_1=10^n g_\mu(0)/\|g_\mu(0)\|^2$, $g(\eta_0)=[g_\mu(0), g_\lambda(0)]$, $W=4$, $t=0.8$, $\delta=10^4$, $\sigma_{max}=(P(\eta_0)+F(\eta_0))/\|g(\eta_0)\|$, $M=1.5$ and $p=0.01$. The initial feasible solution t_{ij}^0 is obtained by solving a linear programming problem of the relaxed problem for the given multipliers $\lambda=0$ and $\mu=0$, and the initial feasible solutions x_{ijk}^0 and y_{irk}^0 can obtained by using the constructive heuristic method.

The coefficients used in the model are: the objective function coefficients $C_1=10$, $C_j=10+20(j-1)(1<j<S)$, $D_1=10$, $D_2=110$; the due date $d_{b(k-1)+m+1}=d_{b(k-1)+m}+\sum_{i=n_{k,1}}^{n_{k,m}}P_{i,S}+T_B$, $d_{i_0}=\sum_{j=1}^{S-1}P_{i_0,j}+\sum_{j=1}^{S-1}T_{j,j+1}$, $i_0=s(b(k-1))+1$, $\bar{k}_m=b(k-1)+m$, $n_{k,1}=s(\bar{k}_m)-s(\bar{k}_m-1)$, $1\le m<b(k)-b(k-1)$, $1\le k\le M_S$, $Su_n\in[75, 80]$, the integer transportation times $T_{j,j+1}$ and the integer processing times $P_{i,j}$ are uniformly generated from Bertsekas (2011), Ding and Bie (2017) and Nishi and Hiranaka (2013), Tseng (2001), respectively.

For the sake of simplicity, we denote stages as S, machines as M, batches as B and jobs as J. For each pair of the batch and job levels, we randomly generate 10 instances to test the performance of these two algorithms. In the experiment, there are 400 test instances to compare the performance of the conditional surrogate subgradient algorithm. In Tables 1–4, the average of duality gaps, running time, iteration, lower bounds and upper bounds are presented to analyze the performance of the proposed surrogate subgradient method. From Tables 1–4, it is easy to obtain the following conclusions:

(1) As viewed from the duality gaps, all the duality gaps of the CSS method and ISS method do not exceed 5%, which imply that the surrogate subgradient methods can produce a high-quality schedule for the SCC scheduling problem. Beside, it is obvious

Table 2
The numerical results of CSS and ISS about the average, lower bounds and upper bounds (Svs.M = 3vs.4).

Bvs.J	Lower bounds		Upper bounds	
	CSS	ISS	CSS	ISS
4vs.11	4312198	4312196	4416314	4416314
4vs.12	5014581	5014581	5158858	5158858
4vs.13	5793466	5793466	5964923	5964923
4vs.14	6575164	6575156	6782956	6782956
4vs.15	7484200	7484191	7673764	7675159
4vs.16	8396496	8396350	8619561	8619561
4vs.17	9370882	9370615	9639696	9639696
4vs.18	10420395	10420384	10714446	10718435
4vs.19	11561097	11561085	11823587	11823587
4vs.20	12642303	12642290	13045006	13045006
5vs.11	6558702	6556445	6730252	6730252
5vs.12	7640738	7640728	7850358	7850358
5vs.13	8831415	8831405	9090072	9094214
5vs.14	10108472	10108458	10373683	10373924
5vs.15	11485109	11485095	11742651	11743238
5vs.16	12917547	12917528	13218739	13218739
5vs.17	14433448	14433433	14753870	14753870
5vs.18	16056261	16056245	16366690	16366690
5vs.19	17755348	17755332	18147475	18147475
5vs.20	19512425	19512405	19974610	19974610
Avg.	10343512	10343369	10604376	10604893

Table 3
The numerical results of CSS and ISS about the average, duality gaps, running time and iteration (Svs.M = 4vs.4).

Bvs.J	Duality gaps (%)		Running time (s)		Iteration	
	CSS	ISS	CSS	ISS	CSS	ISS
4vs.11	**4.4**	4.401	18.72	51.61	70	232
4vs.12	**4.538**	4.786	23.76	78.02	68	269
4vs.13	4.363	4.363	25.87	76.77	63	219
4vs.14	**4.375**	4.376	30.12	88.39	64	225
4vs.15	**4.587**	4.606	34.43	102.24	63	219
4vs.16	**4.275**	4.276	40.41	118.43	63	216
4vs.17	**4.572**	4.638	45.05	127.92	64	217
4vs.18	4.14	4.14	51.29	149.75	62	214
4vs.19	4.02	4.02	55.53	162.64	62	214
4vs.20	4.093	4.093	61.78	174.34	64	216
5vs.11	**3.63**	3.638	32.3	91.55	65	218
5vs.12	**3.995**	4	39.36	122.09	65	240
5vs.13	3.856	3.856	40.37	113.93	63	215
5vs.14	**3.962**	3.963	46.92	133.23	63	213
5vs.15	**3.991**	3.992	54.37	155.81	62	213
5vs.16	3.545	3.545	59.22	175.44	60	212
5vs.17	3.317	3.317	66.27	193.7	60	212
5vs.18	**3.766**	3.767	78.35	205.65	67	210
5vs.19	**3.547**	3.548	91.06	297.24	71	279
5vs.20	**3.352**	3.354	108.82	328.37	72	266
Avg.	**4.0162**	4.03395	50.2	147.36	65	226

The bold values represent that the results of CSS method are better than those of ISS method in terms of the duality gaps.

to see that there are about 60% of the CSS method superior to the ISS method.

(2) In terms of running time and iteration, all the running times and iteration numbers of the CSS method can make at least 60% and 65% reduction as compared to the ISS method, respectively. For example, the running time and iteration of the CSS method can reduced by $\frac{78.02-23.76}{78.02} \times 100\% = 70\%$ and $\frac{269-68}{269} \times 100\% = 75\%$ for the test instance 4vs.4vs.4vs.12, respectively. Therefore, this CSS method can improve the computational efficiency.

(3) From the perspective of the lower bounds and upper bounds, the results of the CSS method have a slight advantage over the ISS method.

In order to compare the performance of the conditional surrogate subgradient method and improved surrogate subgradient method clearly, the running times and iteration of these surrogate subgradient methods are reported at various problem sizes (problem sizes = machine × stage × batch × job) in Figs. 2 and 3, respectively. From Figs. 2 and 3, it is clear to see that the results of the CSS method are much more competitive than the ISS method in terms of the running time and iteration.

In a word, one can have a conclusion that the presented surrogate subgradient method can yield better results than the improved surrogate subgradient method from the analysis of the numerical results.

6. Conclusions

In this paper, we study a new mixed integer nonlinear mathematical model for a real-world scheduling problem arising from the SCC process. Due to its complexity, Lagrange relaxation approach is adopted to solve this problem, which can yield near-optimal

Table 4
The numerical results of CSS and ISS about the average, lower bounds and upper bounds (Svs.M = 4vs.4).

Bvs.J	Lower bounds		Upper bounds	
	CSS	ISS	CSS	ISS
4vs.11	4856901	4856894	5075454	5075454
4vs.12	5601427	5601416	5861641	5870412
4vs.13	6464827	6464819	6753412	6753412
4vs.14	7337041	7337031	7665323	7665323
4vs.15	8239595	8239582	8626736	8628341
4vs.16	9285836	9285820	9691469	9691469
4vs.17	10251524	10251505	10731310	10738385
4vs.18	11395283	11395263	11877380	11877380
4vs.19	12526413	12526391	13040457	13040457
4vs.20	13724231	13724210	14297577	14297577
5vs.11	7273932	7273919	7543114	7543647
5vs.12	8442174	8442158	8786467	8786893
5vs.13	9693909	9693896	10075298	10075298
5vs.14	11026444	11026428	11472335	11472335
5vs.15	12467188	12467168	12975075	12975075
5vs.16	13962246	13962223	14466187	14466187
5vs.17	15553638	15553618	16078471	16078471
5vs.18	17230483	17230457	17891935	17891935
5vs.19	19010043	19009756	19696432	19696432
5vs.20	20847169	20846971	21558097	21558097
Avg.	11257726	11257687	11706662	11707582

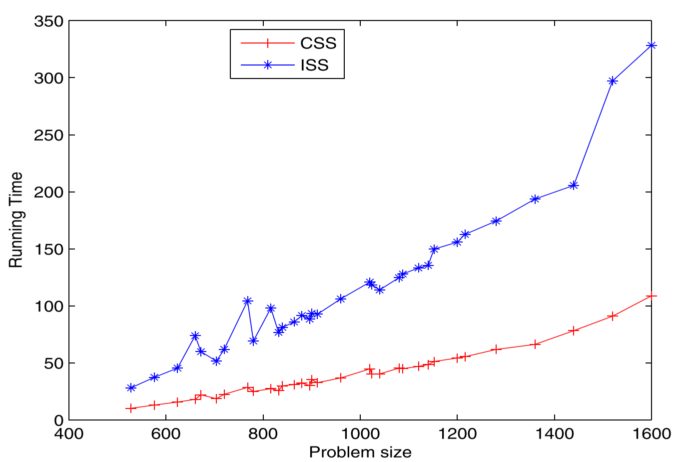

Fig. 2. Performance profiles about the running time.

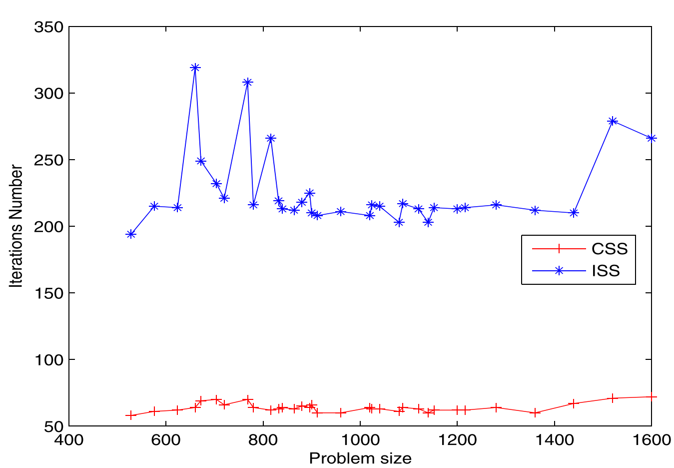

Fig. 3. Performance profiles about the number of iterations.

schedules within a reasonable computational time. By relaxing the complicated constraints, the relaxed problem can be decomposed into three subproblems by using the concave–convex procedure. To improve the computational efficiency, we devise an improved conditional surrogate subgradient algorithm to optimize the LD problem that can overcome the zigzagging phenomena and reduce the calculation time for finding a suitable subgradient. Moreover, a simple heuristic algorithm is proposed to construct a feasible schedule by using the information of the relaxed problem. The numerical results are reported to indicate that the proposed surrogate subgradient method is superior to the improved surrogate subgradient method in terms of the running time and iteration numbers.

Acknowledgements

The authors would like to thank the editor and anonymous referees for their valuable comments and suggestions, which improve this paper greatly. This work was partly supported by the National Natural Science Foundation of China (61333006, 61473074, 51634002).

References

Atighehchian, A., Bijari, M., Tarkesh, H., 2009. A novel hybrid algorithm for scheduling steel-making continuous casting production. Comput. Oper. Res. 36 (8), 2450–2461.
Bertsekas, D.P., 1999. Nonlinear Programming. Athena Scientific, Belmont.
Bertsekas, D.P., 2011. Incremental proximal methods for large scale convex optimization. Math. Program. 129 (2), 163–195.
Bragin, M.A., Luh, P.B., Yan, J.H., Yu, N.P., Stern, G.A., 2015. Convergence of the surrogate Lagrangian relaxation method. J. Optim. Theory Appl. 164 (1), 173–201.
Bellabdaoui, A., Teghem, J., 2006. A mixed-integer linear programming model for the continuous casting planning. Int. J. Prod. Econ. 104 (2), 260–270.
Beltran, C., Heredia, F.J., 2002. Unit commitment by augmented Lagrangian relaxation: testing two decomposition approaches. J. Optim. Theory Appl. 112 (2), 295–314.
Buil, R., Piera, M.Á., Luh, P.B., 2012. Improvement of Lagrangian relaxation convergence for production scheduling. IEEE Trans. Autom. Sci. Eng. 9 (1), 137–147.
Chen, H.X., Luh, P.B., 2003. An alternative framework to Lagrangian relaxation approach for job shop scheduling. Eur. J. Oper. Res. 149 (3), 499–512.
D'Antonio, G., Frangioni, A., 2009. Convergence analysis of deflected conditional approximate subgradient methods. SIAM J. Optim. 20 (1), 357–386.
Ding, T., Bie, Z.H., 2017. Parallel augmented Lagrangian relaxation for dynamic economic dispatch using diagonal quadratic approximation method. IEEE Trans. Power Syst. 32 (2), 1115–1126.
Fu, Y.M., Diabat, A., 2015. A Lagrangian relaxation approach for solving the integrated quay crane assignment and scheduling problem. Appl. Math. Model. 39 (3), 1194–1201.

Gupta, J.N.D., 1988. Two-stage, hybrid flowshop scheduling problem. J. Oper. Res. Soc. 39 (4), 359–364.
Harjunkoski, I., Grossmann, I.E., 2001. A decomposition approach for the scheduling of a steel plant production. Comput. Chem. Eng. 25 (11), 1647–1660.
Horst, R., Phong, T.Q., Thoai, N.V., Vries, J.D., 1991. On solving a DC programming problem by a sequence of linear programs. J. Glob. Optim. 1 (2), 183–203.
Horst, R., Thoai, N.V., 1999. DC programming: overview. J. Optim. Theory Appl. 103 (1), 1–43.
Hao, J.H., Liu, M., Jiang, S.L., Wu, C., 2015. A soft-decision based two-layered scheduling approach for uncertain steelmaking continuous casting process. Eur. J. Oper. Res. 244 (3), 966–979.
Jiang, S.L., Liu, M., Hao, J.H., Qian, W.P., 2015. A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry. Comput. Ind. Eng. 87, 518–531.
Jiang, S.L., Liu, M., Hao, J.H., 2016. A two-phase soft optimization method for the uncertain scheduling problem in the steelmaking industry. IEEE Trans. Syst. Man Cybern.: Syst. 47 (3), 416–431.
Li, J., Xiao, X., Tang, Q., Floudas, C.A., 2012. Production scheduling of a large-scale steelmaking continuous casting process via unit-specific event-based continuous-time models: short-term and medium-term scheduling. Ind. Eng. Chem. Res. 51 (21), 7300–7319.
Li, J.Q., Pan, Q.K., Mao, K., Suganthan, P.N., 2014. Solving the steelmaking casting problem using an effective fruit fly optimisation algorithm. Knowl. Based Syst. 72, 28–36.
Lanckriet, G.R., Sriperumbudur, B.K., 2009. On the convergence of the concave–convex procedure. Adv. Neural Inf. Process. Syst., 1759–1767.
Li, J.Q., Pan, Q.K., Mao, K., 2016a. A hybrid fruit fly optimization algorithm for the realistic hybrid flowshop rescheduling problem in steelmaking systems. IEEE Trans. Autom. Sci. Eng. 13 (2), 932–949.
Li, J.Q., Pan, Q.K., Duan, P.Y., 2016b. An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping. IEEE Trans. Cybern. 46 (6), 1311–1324.
Li, L., Huo, J., Tang, O., 2011. A hybrid flowshop scheduling problem for a cold treating process in seamless steel tube production. Int. J. Prod. Res. 49 (15), 4679–4700.
Li, Z.K., Ierapetritou, M.G., 2010. Production planning and scheduling integration through augmented Lagrangian optimization. Comput. Chem. Eng. 34 (6), 996–1006.
Larsson, T., Patriksson, M., Strömberg, A.B., 1996. Conditional subgradient optimization – theory and applications. Eur. J. Oper. Res. 88 (2), 382–403.
Liu, C., Shahidehpour, M., Wang, J., 2010. Application of augmented Lagrangian relaxation to coordinated scheduling of interdependent hydrothermal power and natural gas systems. IET Gener. Transm. Distrib. 4 (12), 1314–1325.
Lin, J.H., Liu, M., Hao, J.H., Gu, P., 2016. Many-objective harmony search for integrated order planning in steelmaking-continuous casting-hot rolling production of multi-plants. Int. J. Prod. Res., http://dx.doi.org/10.1080/00207543.2016.1232498.
Long, J.Y., Zheng, Z., Gao, X.Q., 2016. Dynamic scheduling in steelmaking-continuous casting production for continuous caster breakdown. Int. J. Prod. Res., http://dx.doi.org/10.1080/00207543.2016.1268277.
Missbauer, H., Hauber, W., Stadler, W., 2009. A scheduling system for the steelmaking-continuous casting process. A case study from the steel-making industry. Int. J. Prod. Res. 47 (15), 4147–4172.
Mao, K., Pan, Q.K., Pang, X.F., Chai, T.Y., 2014. A novel Lagrangian relaxation approach for a hybrid flowshop scheduling problem in the steelmaking-continuous casting process. Eur. J. Oper. Res. 236 (1), 51–60.
Mao, K., Pan, Q.K., Chai, T.Y., Luh, P.B., 2015. An effective subgradient method for scheduling a steelmaking-continuous casting process. IEEE Trans. Autom. Sci. Eng. 12 (3), 1140–1152.
Millar, H.H., Yang, M.Z., 1994. Lagrangian heuristics for the capacitated multiitem lot-sizing problem with backordering. Int. J. Prod. Econ. 34 (1), 1–15.
Nishi, T., Hiranaka, Y., Inuiguchi, M., 2010. Lagrangian relaxation with cut generation for hybrid flowshop scheduling problems to minimize the total weighted tardiness. Comput. Oper. Res. 37 (1), 189–198.
Nishi, T., Hiranaka, Y., 2013. Lagrangian relaxation and cut generation for sequence-dependent setup time flowshop scheduling problems to minimise the total weighted tardiness. Int. J. Prod. Res. 51 (16), 4778–4796.
Pan, Q.K., Wang, L., Mao, K., Zhao, J.H., Zhang, M., 2013. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. IEEE Trans. Autom. Sci. Eng. 10 (2), 307–322.
Pan, Q.K., 2016. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. Eur. J. Oper. Res. 250 (3), 702–714.
Pacciarelli, D., Pranzo, M., 2004. Production scheduling in a steelmaking-continuous casting plant. Comput. Chem. Eng. 28 (12), 2823–2835.
Razaviyayn, M., Hong, M.Y., Luo, Z.Q., 2013. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. 23 (2), 1126–1153.
Sriperumbudur, B.K., Lanckriet, G.R.G., 2012. A proof of convergence of the concave–convex procedure using Zangwill's theory. Neural Comput. 24 (6), 1391–1407.
Sun, L.L., Luan, F.J., Ma, T.M., 2015. A three-level batching strategy for the planning of steelmaking-continuous casting production. IFAC-PapersOnLine 48 (25), 144–149.
Sun, L.L., Yu, S.P., 2015. Scheduling a real-world hybrid flow shop with variable processing times using Lagrangian relaxation. Int. J. Adv. Manuf. Technol. 78 (9-12), 1961–1970.
Tang, L.X., Liu, J., Rong, A., Yang, Z., 2000. A mathematical programming model for scheduling steelmaking-continuous casting production. Eur. J. Oper. Res. 120 (2), 423–435.
Tang, L.X., Luh, P.B., Liu, J.Y., Fang, L., 2002. Steel-making process scheduling using Lagrangian relaxation. Int. J. Prod. Res. 40 (1), 55–70.
Tang, L.X., Liu, G.L., 2007. A mathematical programming model and solution for scheduling production orders in Shanghai Baoshan iron and steel complex. Eur. J. Oper. Res. 182 (3), 1453–1468.
Tang, L.X., Wang, G., 2008. Decision support system for the batching problems of steelmaking and continuous-casting production. Omega 36 (6), 976–991.
Tang, L.X., Wang, X., 2010. An improved particle swarm optimization algorithm for the hybrid flowshop scheduling to minimize total weighted completion time in process industry. IEEE Trans. Control Syst. Technol. 18 (6), 1303–1314.
Tang, L.X., Luo, J.X., Liu, J.Y., 2013. Modelling and a tabu search solution for the slab reallocation problem in the steel industry. Int. J. Prod. Res. 51 (14), 4405–4420.
Tang, L.X., Zhao, Y., Liu, J., 2014. An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. IEEE Trans. Evol. Comput. 18 (2), 209–225.
Tseng, P., 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory Appl. 109 (3), 475–494.
Vallada, E., Ruiz, R., 2011. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. Eur. J. Oper. Res. 211 (3), 612–622.
Wolosewicz, C., Dauzére-Pérés, S., Aggoune, R., 2015. A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. Eur. J. Oper. Res. 244 (1), 3–12.
Wang, S.H., 2003. An improved stepsize of the subgradient algorithm for solving the Lagrangian relaxation problem. Comput. Electr. Eng. 29 (1), 245–249.
Ye, Y., Li, J., Li, Z., Tang, Q., Xiao, X., Floudas, C.A., 2014. Robust optimization and stochastic programming approaches for medium-term production scheduling of a large-scale steelmaking continuous casting process under demand uncertainty. Comput. Chem. Eng. 66, 165–185.
Yu, S.P., Pan, Q.K., 2012. A rescheduling method for operation time delay disturbance in steelmaking and continuous casting production process. J. Iron Steel Res. Int. 19 (12), 33–41.
Yu, S.P., Chai, T.Y., Tang, Y., 2016. An effective heuristic rescheduling method for steelmaking and continuous casting production process with multirefining modes. IEEE Trans. Syst. Man Cybern.: Syst. 46 (12), 1675–1688.
Zhao, Y., Jia, F.Y., Wang, G.S., Wang, L.J., 2011. A hybrid tabu search for steelmaking-continuous casting production scheduling problem. 2011 International Symposium on Advanced Control of Industrial Processes, 535–540.