IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 16, 2020, accepted September 23, 2020, date of publication September 28, 2020, date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027018

Energy-Efficient Integration Optimization of Production Scheduling and Ladle Dispatching in Steelmaking Plants

DAYONG HAN[1,2], QIUHUA TANG[1,2], ZIKAI ZHANG[1,2], AND JUN CAO[3]
[1]Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China
[2]Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan 430081, China
[3]State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Qiuhua Tang (tangqiuhua@wust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 51875421.

ABSTRACT The existing production scheduling mode ignores ladle dispatching resulting in the increase of energy consumption in ladle heating and instability in production. Hence, we study the energy-efficient integration optimization of production scheduling and ladle dispatching in this paper. Specifically, a mixed integer linear programming model is formulated to coordinate the time-dependent correlations between them and quantify the energy consumption of them. Moreover, an enhanced migrating birds optimization algorithm (EMBO) is proposed to tackle this NP-hard integration optimization problem. In this proposed algorithm, a three-level rule-based heuristic decoding is designed to achieve the optimal solutions at the given production sequence; well-designed neighborhood structures are appended to intensify exploration; a simulated annealing-based acceptance criterion is hired to escape from local optima. Additionally, a novel competitive mechanism for birds regrouping is developed to increase the population diversity by information exchange between the left and right lines of V-formation. Mass experimental results demonstrate that the proposed EMBO observably outperforms all the compared algorithms, and the proposed integration optimization decreases the energy-consumption by 1.21% in the context of constant production efficiency.

INDEX TERMS Hybrid flow-shop, integration optimization, ladle dispatching, migrating birds optimization algorithm, production scheduling.

I. INTRODUCTION

The iron and steel sector accounts for no less than 18% of the total industrial energy consumption on a global scale and is regarded as one of the most energy-intensive manufacturing processes [1], [2]. The steelmaking and continuous-casting (SCC) process is a key phase in the whole steel manufacturing process, and the production management of SCC plays a determinant role in energy saving. In another word, small improvements in energy efficiency in SCC plants may translate into tremendous gains in overall energy savings and cost reductions [3]–[5]. Hence, there has recently been growing research interest in energy savings in SCC plants.

Production scheduling and ladle dispatching are two principal sub-systems in SCC production management [6], [7]. As illustrated in Figure 1, the molten iron (red color) is transformed into molten steel with a given chemical composition via steelmaking and refining processes, and further transformed into slabs with specific strength and dimensions via continuous-casting process. Production scheduling determines not only the allocation of machines at each stage but also the timing for performing corresponding tasks [8], [9]. And then, ladle dispatching receives the schedule of SCC

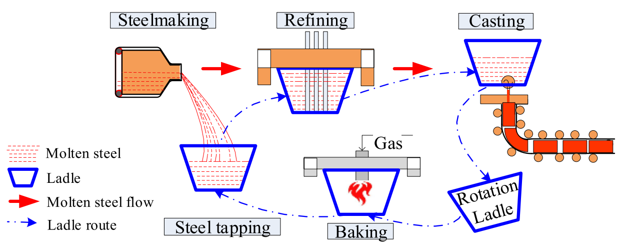


FIGURE 1. Production scheduling and ladle dispatching in SCC.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Zakarya.

production as an input and accordingly determines the assignment of ladles (blue color), the exclusive transportation devices of the molten steel. In the turnover period of each ladle, an empty ladle (no molten steel in the ladle) awaits in the waiting area till receiving the molten steel from the converter. Next, the ladle filled with molten steel is transported to the refining furnace for refining and then to the continuous caster for casting. After pouring molten steel into a caster and emptying the remains at slag place, the empty ladle returns to the waiting area. It is worth noting that in the waiting area, all the ladles need to be heated to the prescribed temperature before they are enabled again. Hence, an unreasonable schedule of ladle dispatching may result in a longer heating time and further bring about waste of energy consumption.

It should be pointed out that ladle resources are treated as negligible parameters in existing studies on the SCC scheduling problem [2], [4], which is obviously impractical. The inadequate quantity of ladles may delay the manufacturing cycle time and even cause the production disorder while an excessive number of ladles in process may cause unnecessary energy loss. It is essential to make a decision on a reasonable number of ladles to be utilized on the spot. However, since the production schedule is now regarded as the input of ladle dispatching, this sequential decision process causes serious problems as the following.

(1) For the reason that it costs more than two hours to heat a cooling ladle from the environmental temperature to the needed, a large variety of disruptions leading to frequent changes in the production schedule cannot be timely handled by ladle dispatching.

(2) To avoid the production disorder due to the lack of ladles, the current solution of reserving a relatively large number of ladles not only causes huge startup energy consumption, but also consumes considerable energy consumption to keep the ladles at the prescribed temperature [1], [3], [10].

Hence, the integration optimization of production scheduling and ladle dispatching (IPS-LD) demonstrates brand-new characteristics of high temperature and high energy consumption, and hence should be investigated thoroughly and systematically. Tan, *et al.* selected a limited number of ladles from the available ones and scheduled them with the goal of minimizing the total gas consumption [11]. Based on this, Tan, *et al.* further proposed a high-temperature-ladle matching rule to further reduce the total gas consumption [10]. Huang, *et al.* considered the ladle scheduling as the key problems of temperature drop in the transportation of hot metal and investigated the influencing factors of steel ladles exchange during the steelmaking and continuous casting process [12]. However, it can be seen from above that ladle dispatching is conducted independently during the optimization procedure. That is to say, the ladle dispatching problem is not included in the production scheduling in most recent studies on SCC production scheduling. It is usually impractical to timely obtain the ladle for charges in process under complex practical environments such as taking account of baking time for ladles. In sum, although important in practice and more effective and efficient policies are highly desirable, the research on IPS-LD is still in the infant stage. And this is also the main motivation behind this paper.

On the other hand, the SCC scheduling problem has received considered attention in the past decades. Various algorithms have been developed to obtain optimal or near-optimal solutions [13]–[15]. These algorithms include heuristics [16], [17], meta-heuristics [18], [19] and exact algorithms [20]. Generally, exact algorithms such as branch and cut algorithm [21] and a branch and bound algorithm [22] may solve the small-scale integration cases with simple features to optimality. It is well known that the computational complexity exponentially increases as the problem scale grows, and hence exact algorithms cannot solve the large-sized instance within an acceptable time. Heuristic algorithms focus on problem-specific features and may solve large-scale problems in extremely short computation time [23]–[25]. However, the quality of the derived solutions might not be satisfactory since it is difficult to combine all the features into a simple heuristic algorithm. By contrast, meta-heuristic algorithms can obtain high-quality solutions within short computational time via global and local searches [26]–[28]. Among the metaheuristics, migrating birds optimization algorithm (MBO), as a new meta-heuristic algorithm inspired by the V-shaped flight formation of migrating birds, has been proved to be effective on energy conservation. This algorithm is unique where the benefit mechanism is utilized to replace the poor-quality solution and accelerate the evolution process greatly [29]–[31]. Meanwhile, this algorithm has shown superiority over other algorithms in the related SCC [32], [33]. Hence this work selects MBO to tackle the proposed IPS-LD. To tackle the integration optimization of production scheduling and ladle dispatching effectively and efficiently, it is recommended to combine the known expertise of heuristic algorithms and optimization abilities of meta-heuristics together. This is also the main contribution behind this paper. Therefore, an enhanced migrating birds optimization algorithm (EMBO) with a problem-specific three-level heuristic decoding mechanism and several improvements, is designed to tackle the large-scaled integration optimization problem. Therefore, in this paper, considering characters of IPS-LD, our focus is on developing problem-oriented approach to tackle this proposed integration problem. The contributions are presented as follows.

(1) The internal correlations between production scheduling and ladle dispatching are represented with time-dependent functions; and a mixed integer linear programming problem is further formulated for the integration optimization of them.

(2) The energy consumption of activating a new ladle or continuously baking the incumbents is quantitatively expressed and further minimized in order to achieve the overall energy savings of production and transportation in SCC plants.

(3) An enhanced migrating birds optimization algorithm (EMBO) with a problem-specific three-level heuristic decoding mechanism is designed to tackle the large-scaled integration optimization problem.

(4) Experimental results indicate that via the integration optimization of production scheduling and ladle dispatching, the energy consumption has been reduced by 1.21% while the productivity efficiency remains constant.

The rest of this paper is structured as follows. Section II describes and formulates the energy-efficient scheduling model for this integration optimization problem. Section III provides a problem-specific three-level heuristic technique for decoding. Section IV gives a brief illustration of the migrating birds optimization algorithm. Section V presents an enhanced migrating birds optimization algorithm with several improvements to tackle this problem. Section VI reports the results of numerical experiments and finally Section VII provides conclusions and future research venues.

II. PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

In this section, we first describe the integration optimization problem of production scheduling and ladle dispatching within a SCC plant, then model this problem as a mathematical optimization problem with two objectives of minimizing the total weighted completion time and energy consumption. Note that, these two objectives are the most significant measures for the improvement of production efficiency and energy savings within SCC plants.

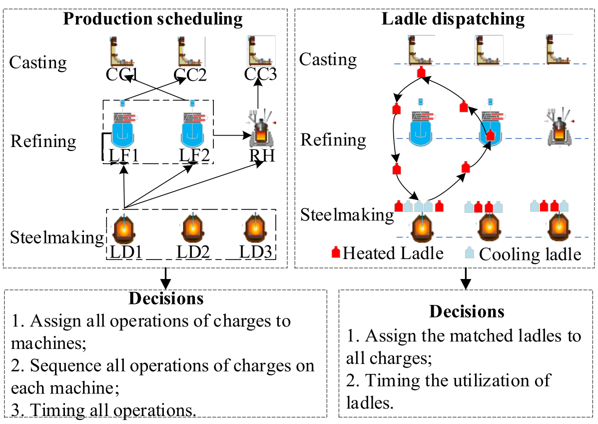


FIGURE 2. The integration of production scheduling and ladle dispatching.

A. PROBLEM DESCRIPTION

The SCC process is mainly comprised of three stages as exhibited in Figure 2: steelmaking, refining and continuous-casting. In the steelmaking stage, molten iron and scrap steel are poured into melting converter furnace (LD) and then transformed into molten steel. After steelmaking, molten steel is poured into a ladle and hence referred to as a charge, the smallest production unit, corresponding to job in the flow shop scheduling. Subsequently, the molten steel in a ladle is moved to refining equipment Ladle-Furnace (LF) to raise its temperature and adjust its chemical compositions, or even undergo the secondary refining process of Ruhrstahl-Hausen (RH) furnace for hydrogen removal. Finally, refined molten steel is transported to the continuous casters (CC) where several charges with the same ingredients and specifications are processed consecutively as a batch (or cast).

In respect to the ladle utilization, only the ladle satisfying the given molten steel's requirements on the composition and temperature (also called a matched ladle) can be utilized as a transportation device. This ladle is first loaded with molten steel at the completion time of steelmaking. Subsequently, the loaded ladle is transported to the refining process, and then to the turntable till the molten steel inside is overturned to the continuous caster. Afterwards, almost-empty ladle is turned over completely and quickly for dumping the steel slag, and finally is returned to the baking area near the converter. If this empty ladle will be reused soon after, it must keep heating immediately in the baking area and prepare for the next use. Otherwise, it may stay there, cool to environmental temperature and be reheated from the environmental temperature just before the next use. And the reheating process from the environmental temperature to the prescribed normally costs two hours or more with additional waste of energy consumption by natural gas.

Concerning the correlation of production operations and ladle transportation, it is obvious from Figure 2 that the completion time of steelmaking operation equals to the start time of a utilized ladle, and the start time of casting is approximately equivalent to the completion time of the ladle. In addition, a ladle can be reused only after it comes back to the baking area from the completion of the previous use.

B. MATHEMATICAL FORMULATION

For the proposed integration problem, assumptions are presented as follows:

1) All parameters are deterministic.

2) The sequence of the charges belonging to the same cast is predefined.

3) Setup time is technologically required before the start of the next cast on a continuous caster. The length of setup time is independent upon cast sequences and material properties.

4) Machine malfunction will not occur.

5) The residual life of ladles is given, which limits the maximal number of charges to be transported.

The following notations describe indices, sets, elements, parameters and variables.

INDICES

i	The stage indexes.
j	The charge (job) indexes.
l	The cast index (also called batch index).
m	The machine indexes.
p	The ladle indexes.

SETS

I_j The stage set of charge j, $I_j = \{i | i = 1, 2 \cdots I_j\}$, where I_j is the number of stages and indicates the continuous casting stage.

J The set of all charges and $J = \{j | j = 1, 2 \ldots j\}$.

L The set of all casts and $L = \{l | l = 1, 2 \ldots L\}$.

P The set of all ladles, $P = \{p | p = 1, 2 \ldots P\}$, where P is the total number of alternative ladles;

P_j The set of available ladles for charge j, $P_j \subseteq P$, $for\ all\ j$.

$LJ_{(l,j)}$ Set of charges in the lth cast, $l = \{1, 2 \cdots L\}$, where L is the total number of casts; $LJ_{(l1,j)} \cap LJ_{(l2,j)} = \emptyset\ for\ l1 \neq l2,\ and\ \bigcup_l LJ_{(l,j)} = J$.

M_i The set of machines in stage i.

J_l^s The set of the start charge in the lth cast.

J_l^e The set of the last charge in the lth cast.

PARAMETERS

U A sufficient large positive number.

$r_{i,m}$ The release time of the mth machine.

r_j The release time of charge j.

$pt_{j,m}$ The processing time for the charge j on machine m.

s_p Move time of ladle p from casters to the baking area.

s_u The setup time of casters between adjacent casts to prepare for the next cast.

t_p The residual life of ladle p.

r_0 The weight coefficient for the objective of the total completion time.

ei_m Per unit processing energy consumption when machine m is in process.

er_p Per unit energy consumption to bake ladle p.

eq_p Per unit energy consumption to enable ladle p for the next use.

VARIABLES

$X_{i,j,m}$ Binary variable. Takes value 1 if charge j is being processed at stage i on machine m and 0 otherwise.

$Y_{i,j,j'}$ Binary variable. If charge j is processed before charge j' at stage i, $y_{i,j,j'} = 1$; otherwise, $y_{i,j,j'} = 0$.

$Q_{j,P}$ Binary variable. Takes value 1 if charge j is transported by ladle p and 0 otherwise.

$Z_{p,j,j'}$ Binary variable. If charge j is transported just before charge j' by ladle p and 0 otherwise.

K_P Binary variable. Takes value 1 if ladle $p$isutilized and 0 otherwise.

$Wt_{j,j',p}$ Continuous variable, waiting time between two adjacent charges j and j' on ladle p.

$Ts_{i,j}$ Continuous variable, start time of charge j at stage i.

$Tf_{i,j}$ Continuous variable, completion time of charge j at stage i.

$Tb_{j,p}$ Continuous variable, the start time of charge j on ladle p.

$Te_{j,p}$ Continuous variable, the end time of charge j on ladle p.

TEC Free variable, the total weighted energy consumption.

C_{max} Positive variable, the maximum completion time, also be known as makespan.

E_1 Positive variable, the processing energy of machines to fulfill the given processing tasks.

E_2 Positive variable, the idle energy consumption for transporting ladles between two adjacent stages.

E_3 Positive variable, the baking energy consumption for baking ladles continuously at the needed temperature between two adjacent uses.

E_4 Positive variable, the startup energy for reheating a ladle to the prescribed temperature from the environment temperature.

With notations above, the integration optimization of production scheduling and ladle dispatching (IPS-LD) is formulated as follows. Note that, all charges must be allocated to and sequenced on suitable machines and delivery ladles, and be scheduled under the constraints of technological requirements. In addition, the objective of this paper is to find a schedule that minimizes the weighted completion time and the total energy consumption simultaneously.

1) ALLOCATION AND SEQUENCING CONSTRAINTS

Each charge must be allocated to exactly one machine at any stage within its process route for processing. Particularly, in the casting stage all the charges in a cast must be processed sequentially on the same caster.

$$\sum\nolimits_{m \in M_i} X_{i,j,m} = 1, \quad \forall i \in I_j, j \tag{1}$$

$$X_{i,j,m} = X_{i,j',m}, \quad \forall i = s, (j, j') \in LJ_{(l,j)},\ m \in M_i \tag{2}$$

Exactly one ladle satisfying technological requirements must be selected and allocated to a charge for transportation, and the remaining usage times of this ladle must be less than its residual life.

$$\sum\nolimits_{p \in \mathbf{P}_j} Q_{j,P} = 1, \quad \forall j \tag{3}$$

$$\sum\nolimits_{j} Q_{j,P} \leq t_p, \quad \forall p \tag{4}$$

For two different charges at a given stage, one charge may start before the other and vice versa. Even if two charges start at the same time on different machines belonging to this stage, one of them is supposed to be in advance of the other.

$$Y_{i,j,j'} + Y_{i,j',j} = 1, \quad \forall i, j, j' \neq j \tag{5}$$

All charges allocated to a ladle for transportation should be sequenced in an order.

$$Z_{p,j,j'} + Z_{p,j',j} \leq 1, \quad \forall p, j, j' \neq j \tag{6}$$

$$Z_{p,j,j'} + Z_{p,j',j''} + Z_{p,j'',j} \leq 2, \quad \forall p, j, j \neq j' \neq j'' \tag{7}$$

2) TIMING CONSTRAINTS

For two consecutive operations of a charge, the following one can start only when its preceding has terminated.

$$Ts_{i+1,j} \geq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m}, \quad \forall j, m \in M_i, m' \in M_{i+1}, (i, i+1) \in I_j \quad and\ i < s. \tag{8}$$

At the casting stage, all charges in a cast must be processed consecutively till the completion of the last charge in this cast as shown in equations (9-10). In these equations, U means a sufficient large positive number and it is utilized in the constraints to ensure that the constraint takes effect only when $Y_{i,j,j'} = 1$ are satisfied, otherwise, these equations are relaxed. And between any two adjacent casts allocated to the same casters, a setup time should be reserved for changing equipment.

$$Ts_{i,j'} \leq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} + U \cdot \left(1 - Y_{i,j,j'}\right), \quad \forall m \in M_i,\ i = s,\ \left(j, j'\right) \in LJ_{(l,j)},\ and\ j \prec j' \tag{9}$$

$$Ts_{i,j'} \geq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} - U \cdot \left(1 - Y_{i,j,j'}\right), \quad \forall m \in M_i, i = s, (j, j') \in LJ_{(l,j)},\ and\ j \prec j' \tag{10}$$

$$Ts_{i,j'} \geq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} + s_u - U \cdot \left(3 - X_{i,j,m} - Y_{i,j',m} - Z_{i,j,j'}\right), \quad \forall j \in J_l^e, j' \in J_l^s, \quad i = s,\ and\ j \prec j' \tag{11}$$

With respect to processing machine capacity, a machine can process at most one charge at a time.

$$Ts_{i,j'} - Ts_{i,j} - \sum\nolimits_{m' \in M_i} X_{i,j,m'} \cdot pt_{j,m'} + U \cdot \left(3 - X_{i,j,m} - X_{i,j',m} - Y_{i,j,j'}\right) \geq 0, \quad \forall i \in I_j, j, j', m \in M_i \tag{12}$$

$$Ts_{i,j'} - Ts_{i,j} + U \cdot \left(3 - X_{i,j,m} - X_{i,j',m'} - Y_{i,j,j'}\right) \geq 0, \quad \forall i \in I_j, j, j', \left(m, m'\right) \in M_i, m \neq m' \tag{13}$$

In regard to ladle utilization, if two charges are allocated to a ladle for transportation, the transportation operation of the following charge can start only after the preceding one has been completed.

$$Tb_{j',p} - Tb_{j,p} - \sum\nolimits_{i>1, i<I_j-1} \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} + U \cdot \left(3 - Q_{j,P} - Q_{j',P} - Z_{p,j,j'}\right) \geq s_p, \quad \forall j, j', p \tag{14}$$

3) CORRELATION CONSTRAINTS

Provided that the *pth* ladle is assigned to charge *j*, the start time for transporting charge *j* equals to the completion time of this charge at the steelmaking stage. Otherwise, these two constraints are relaxed.

$$Tb_{j,p} \leq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} + U \cdot \left(1 - Q_{j,P}\right), \quad \forall i = 1, j, p \in P_j \tag{15}$$

$$Tb_{j,p} \geq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} - U \cdot \left(1 - Q_{j,P}\right), \quad \forall i = 1, j, p \in P_j \tag{16}$$

Similarly, as long as the *pth* ladle is assigned to charge *j*, the end time for transporting charge *j* equals to the start time of this charge at the continuous casting stage.

$$Te_{j,p} \leq Ts_{i,j} + U \cdot \left(1 - Q_{j,P}\right), \quad \forall i = s, j, p \in P_j \tag{17}$$

$$Te_{j,p} \geq Ts_{i,j} - U \cdot \left(1 - Q_{j,P}\right), \quad \forall i = s, j, p \in P_j \tag{18}$$

4) OBJECTIVE FUNCTIONS

For most steelmaking companies, makespan is a common and essential index which accurately reflects the productivity.

$$C_{max} \geq Ts_{i,j} + \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m}, \quad \forall i = s, j \tag{19}$$

The total energy consumption comprises four aspects: processing energy consumption, idle energy consumption, baking energy consumption, startup energy consumption. Among them, the processing energy consumption is the electricity energy consumed by processing machines.

$$E_1 = ei_m \times \sum\nolimits_{j=1}^{n} \sum\nolimits_{i \in I_j} \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} \tag{20}$$

The idle energy consumption of a charge is lost by transportation and waiting time between any two stages.

$$E_2 = er_p \times \left\{ \sum\nolimits_{j=1}^{n} \left[\sum\nolimits_{p} \left(Te_{j,p} - Tb_{j,p}\right) - \sum\nolimits_{i>1|i \in I_j}^{i<s} \sum\nolimits_{m \in M_i} X_{i,j,m} \cdot pt_{j,m} \right] \right\}, \tag{21}$$

If two successive charges are allocated to a ladle for transportation, after completing the preceding charge, the ladle will stay at the baking area and wait for the next use. During the waiting period, the ladle to be utilized will keep heated by gas and hence result in the baking energy consumption of ladles.

$$E_3 = er_p \times \sum\nolimits_{t|t<T_p} \sum\nolimits_{p} Wt_{p,j,j'}, \tag{22}$$

where,

$$Wt_{p,j,j'} \geq Tb_{j',p} - Te_{j,p} - U \cdot \left(3 - Q_{j,P} - Q_{j',P} - Z_{p,j,j'}\right), \quad \forall j, j', p \in P_j \tag{23}$$

$$Wt_{p,j,j'} \leq Tb_{j',p} - Te_{j,p} + U \cdot \left(3 - Q_{j,P} - Q_{j',P} - Z_{p,j,j'}\right), \quad \forall j, j', p \in P_j \tag{24}$$

The startup energy consumption is involved when a ladle is first enabled after a long-term stay at the environment temperature. Note that, the startup energy consumption of ladles is huge and inevitable.

$$E_4 = eq_p \times K_P, \quad \forall p \tag{25}$$

where,

$$U \cdot K_P \geq \sum\nolimits_{j} Q_{j,P}, \quad \forall p \tag{26}$$

Therefore, the objective function is minimizing the weighted summation of the maximum completion time and the total energy consumption so as to improve the production efficiency and energy saving simultaneously.

$$\min : TEC = r_0 \cdot C_{max} + (1 - r_0) \cdot \left\{ ei_m \times E_1 + er_p \times (E_2 + E_3) + eq_p \times E_4 \right\} \quad (27)$$

Therefore, using Equations (1-27), the proposed IPS-LD is formulated as a mixed integer linear programming model. This model can be used to exactly solve small-sized instances of the problem to optimality by the software such as GAMS/Cplex. However, it is known that production scheduling of SCC is NP-hard [34]. Hence, the proposed IPS-LD must be strong NP-hardness, and an effective and well-designed meta-heuristic is proposed to tackle this proposed integration problem.

III. THREE-LEVEL HEURISTIC BASED SOLUTION GENERATION

Compared with pure production scheduling of SCC, this integration problem involves more decisions and mutual correlation between production scheduling and ladle dispatching. To present the solution procedure clearly, we first design the way of generating the optimal solution under a given production sequence at the first stage. This specially designed approach based on the acquired expertise is denoted as the three-level heuristic-based solution generation method. In the method, the upper two levels, the forward and backward heuristic mechanisms, produce a feasible production schedule; the third level provides a feasible ladle dispatching plan under the limitation of the given production schedule.

A. FIFO-BASED FORWARD HEURISTIC

As mentioned in Equation "27", the first scheduling goal is to obtain the objective of makespan as small as possible. The FIFO-based forward heuristic is thus proposed to ensure that each charge may be processed as early as possible. As long as a production sequence at the first stage is given, this heuristic helps to allocate charges to processing machines and sequence all charges allocated to a machine in an order.

This heuristic is implemented from the first stage to the last according to breadth first principle. In each stage, all the charges are first allocated to processing machines and then those of them allocated to a machine are sequenced in an order. Particularly, in the first stage, each charge will be assigned to the first available machine according to a given production sequence, and then a charge that has an earlier completion time of the first stage is endowed with a higher priority according to the FIFO rule (First in, First out), and hence is ranked first in the sequence.

Compared with the first two stages, the allocation and sequencing in the casting stage follows the FIFO rules too and shows some unique characteristics. The difference is the fixed machine assignment: as long as the first charge in a cast has been allocated to a caster, all other charges in this cast must be allocated to the caster too.

With all things above, the FIFO-based forward heuristic is designed and provided in Algorithm 1.

Algorithm 1 FIFO-Based Forward Heuristic

Input: Casts and charges, release times of machines and charges ($r_{i,m}$, r_j), a production sequence at the first stage.
Output: Machine assignment, charge sequence, completion time, makespan.
//Forward heuristic for steelmaking and refining stages//

For $i = 1$ *to* $s - 1$ **do**
  While $j \leq |\boldsymbol{J}|$ **do**
    Select charge j sequentially according to FIFO, where $j^* = \min_j r_j$;
Assign charge j^* to the earliest available machine m^*, where $r_{i,m^*} = \min_{m \in \boldsymbol{M}_i} r_{i,m}$;
    Schedule charge j^* on the assigned machine, set $ST_{i,j^*} = max\{r_{i,m^*}, r_{j^*}\}$;
    Update $r_{i,m^*} = ST_{i,j^*} + PT_{i,j^*}$;
  End while
End for.
// Forward heuristic for casting stage//
While $l \leq L$ **do**
  Select the earlier available caster for cast l, $m^* = \arg_{m \in M_s} \min\{r_{s,m}\}$;
  Compute the casting starting time of cast l, $Tsi_{s,\boldsymbol{LJ}_{(l,J_l^s)}} = \max\{Tf_{s-1,\boldsymbol{LJ}_{(l,J_l^s)}}, r_{s,m^*} + su\}$;
  Set $t = J_l^s + 1$;
  While $t \leq \boldsymbol{J}_l^e$ **do**
    Compute the casting starting time of other charges, $Tsi_{s,\boldsymbol{LJ}_{(l,t)}} = \max\{Tsi_{s,\boldsymbol{LJ}_{(l,t-1)}} + PT_{\boldsymbol{LJ}_{(l,t-1)},m^*}, Tf_{s-1,\boldsymbol{LJ}_{(l,j)}}\}$;
    Set $t = t + 1$;
  End while
  Set $l = l + 1$;
End while

Using the aforementioned forward heuristic, the machine assignment, charge sequence on each machine, and makespan are obtained. It should be noted that the continuous casting constraints are neglected unfortunately here, and hence the timing of all charges needs to be reconsidered to ensure the continuity of casting in a cast and reduce the unnecessary energy consumption.

B. DEPTH-FIRST BACKWARD HEURISTIC

To satisfy the constraints of continuous casting and reduce the useless energy consumption, the depth-first backward heuristic adjusts the timing of all charges one by one. This adjusting process is performed according to the descending order of the completion times in the last stage of all charges. For each charge, the timing is recalculated from the continuous casting stage to the refining and then to the steelmaking according to depth-first principle.

It is worth noting that at the continuous casting stage, all the charges except the last one in a cast must move right so as to satisfy the continuous casting constraint. Due to this, the difference between the start time of the current stage and the completion time of the previous stage is enlarged;

and accordingly, the idle energy consumption has increased unfortunately. To reduce idle energy consumption, the timing of this charge at the refining and steelmaking stages must move right, too. In this process, the charge sequence from the FIFO heuristic will be kept unchanged. This adaption can be described by Algorithm 2.

Algorithm 2 Depth-First Backward Heuristic

Input: Machine assignment, charge sequence, completion time, makespan.
Output: Production schedule.

//Continuous casting stage//
Sort all casts in the descending order of the completion time of the last charge in these casts, generate a cast list π of the cast stage, let π_l denote that cast l is in the π_l position in the list;
While $\pi_l \leq L$ **do**
  Confirm start time of the last charge in cast l, $Ts_{s,J_l^e} = \max\{Ts_{s,J_l^e-\mathbf{1}} + pt_{J_l^e-1,m^*}, Tf_{s-1,J_l^e}\}$;
  Set $t = J_l^e - 1$;
  While $t \geq J_l^s$ **do**
    Adjust the start time of charge t, $Ts_{s,t} = Ts_{s,t+1} - pt_{t,m^*}$, to ensure this charge can be cast continuously;
    Set t = t − 1;
  End while
    Set $\pi_l = \pi_l + 1$;
End while
//Refining and steelmaking stages//
Sort all charges in the descending order of their completion times, generate a cast list π of all charges, let π_j denotes that charge j is in the π_j position in the list;
Set $\pi_j = 1$;
While $\pi_j \leq J$ **do**
  For $i = s - 1 to 1$ **do**
    Adjust start time of charge j on the given machine m, $Ts_{i,j} = \min\{r_{i,m}, Ts_{i,j}\} - pt_{j,m}$;
    Update $r_{i,m} = Tsi_{i,j}$;
  End for
    Set $\pi_j = \pi_j + 1$;
End while

C. RULES-BASED LADLE DISPATCHING HEURISTIC

For the purpose of reducing energy consumption, the primary task of ladle dispatching ($Q_{j,P}$) is to ensure that the total number of ladles in use is as small as possible and all the ladles in use have a high utilization rate. Meanwhile, many factors should be considered. First, only the matched ladle can be utilized for the transportation of a charge, i.e., $p \in \mathbf{P}_j$. And, if more than one ladle can be utilized for transporting a charge, special measures are needed to choose a suitable ladle for the purpose of reducing energy consumption. Hence, we separate all the ladles into two types: heating ladles and cooling ladles. The former is reheated immediately as long as it returns to the baking area while the latter cools to the environmental temperature without restriction. For each type of ladles, we propose a rule to choose a suitable ladle respectively.

TABLE 1. The Characteristics of the given charges.

Cast {Charges in cast}	Processing time LD	LF	RH	CC	Available Ladle style for casts
1{1}	70	45	-	61	L1, L2
2{2,3,4}	70	45	-	56	L1, L2
3{5,6,7,8}	70	45	40	55	L3, L4

"-" denotes casts 1 and 2 will not undergo the RH stage;

Rule 1: Among all heating ladles, give higher priority to the ladle that has been baked for a shorter period i.e. the ladle with a shorter waiting time since the last utilize.

$$p^* = arg_p \min\left\{U * \left(1 - K_p\right) + Wt_{p,j,j'}\right\}. \tag{28}$$

Proof: If K_p equals 1, the ladle p must be in heating status. In this case, the reuse of this ladle may reduce the huge startup energy consumption. Additionally, among the ladles heated in the baking area, the shorter the baking time is, the larger the recirculation times is, and the less the consumed energy will be. For a long time period, the ladle with larger baking time will be gradually out of use and hence the number of ladles in use will be reduced, too.

Rule 2: Among all cooling ladles, endow the ladle p^* having the bigger residual life t_p with the higher priority.

$$p^* = arg_p \max\left\{t_p\right\}. \tag{29}$$

Proof: It is known that in the ladle transportation the temperature of the molten steel may drop owning to two types of heat loss: the heat loss due to evaporation from the surface of the molten steel and that for heat conduction by the shell and lining of the ladle. Among them, the heat loss by ladle lining is about 40% $\sim$ 50% of the total; and, the larger the residual life t_p is, the less energy consumption that the refractory material of ladle lining needs to achieve the prescribed temperature [35]. To reduce the heat loss by ladle lining, the ladle with the bigger residual life should be reused first.

Based on the above two rules, the detailed steps for ladle dispatching are given as follows.

D. AN ILLUSTRATIVE EXAMPLE

This illustrative example comes from a real SCC production plant [36]. At the steelmaking and casting stages, there are two parallel machines (LDs and CCs) respectively. And at the refining stage, there are two process routes, (LF) and (LF-RH). Among them, the LF process has two parallel machines while the RH process has only one machine. For clarity, a small industrial case is taken as an example. In this case, only three casts including eight charges are considered, and the detailed characteristics of these charges and ladles are represented in Table 1.

Algorithm 3 Rules-Based Ladle Dispatching Heuristic

Input: Production schedule, release time of ladles r_p.
Output: Ladle schedule.

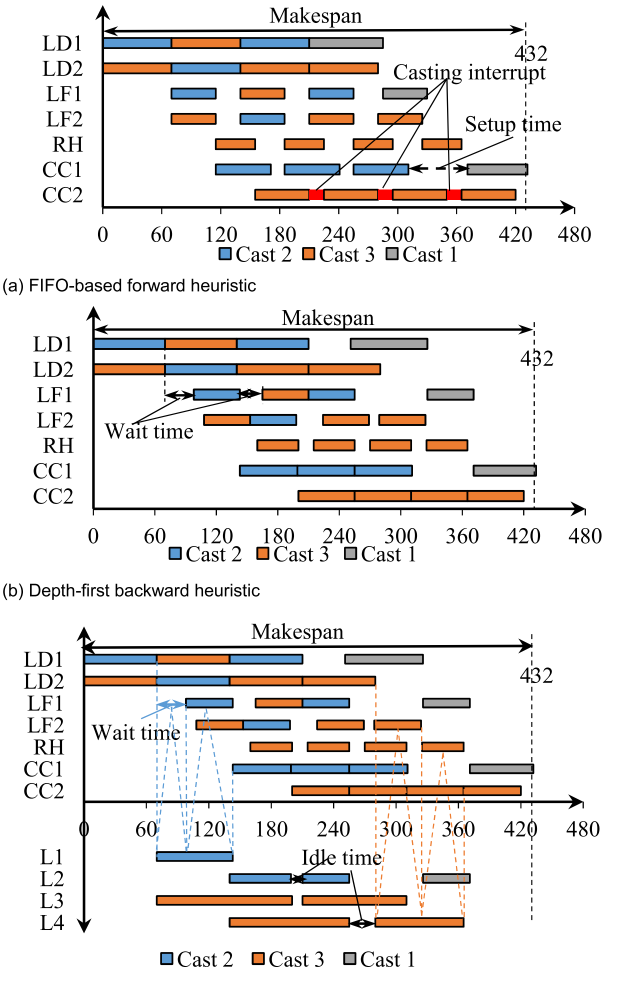
Sort all charges in the increasing order of completion times at the steelmaking stage, and generate a charge list π of all the charges, let π_j denote that charge j is at the π_j position of the list;
While $\pi_j \leq J$, **do**
    Find all heating ladles satisfying technological requirements, set $\mathbf{\Phi_j}$=$\mathbf{P}_j \cap \{p|r_p < Tf_{1,j}\}$;
  If $\mathbf{\Phi_j} \neq \phi$
    Select the ladle p^* with **Rule 1**;
  Else
    Select the ladle p^* with **Rule 2;**
  End if
    Update the start/end time of ladle p^* for transporting charge j, $Tb_{j,p^*} = Tf_{1,j}$, $Te_{j,p^*} = Ts_{s,j}$;
    Update the release time of ladle p^*: $r_{p^*} = Te_{j,p^*}$;
End while

Provided that a given production sequence at the first stage is {2, 5, 3, 6, 4, 7, 1, 8}, using the FIFO-based forward heuristic, every charge in each stage is allocated to the first available machine as long as its previous process has been completed as shown in Figure 3(a). It can be seen that the continuous-casting constraint between adjacent charges in a cast is interrupted and the current solution is infeasible.

Then, the depth-first backward heuristic is utilized to move some charges right from the continuous casting stage to the refining stage and then to the steelmaking as illustrated in Figure 3(b). As a result, all the charges in a cast can be cast continuously and the idle energy consumption between any two successive stages can be reduced to minimum. Finally, all the charges are sorted according to the ascending order of completion times at the first stage, and then one by one, exactly one of all the matched ladles is allotted to a charge according to the rules-based ladle dispatching heuristic as represented by Figure 3(c). Finally, a feasible and optimal solution is generated for the given production sequence.

IV. THE BASIC MBO

Migrating birds optimization (MBO) is a recent high-performing meta-heuristic algorithm inspired by the migrating bird's flight in a V-shaped formation, which has obtained competing performances in kinds of combinatorial optimization problems [37]–[39]. The main feature of MBO is sharing their neighbors with others, promoting the evolution of the whole swarm. This algorithm starts with birds (population) initialization, where these birds are put on a hypothetical formation. In this formation, a head bird leads other birds following on the right and left lines. Specifically, first, α solutions are generated in the feasible solution space in a random manner. Then, the solutions are arbitrarily placed on a hypothetical 'V' formation containing one leader solution,



(a) FIFO-based forward heuristic
(b) Depth-first backward heuristic
(c) Rules-based ladle dispatching heuristic

FIGURE 3. Overview of three-level heuristic-based solution generation.

(α-1)/2 solutions in the left line and (α-1)/2 solutions in the right line. It should be noted that when the leader is tired, it will move to the tail of the following birds and another bird will take its place to lead the population. The details for the basic MBO are shown as follows.

//Initialization
  Initialize α birds;
    % α is the number of swarm size
  Put birds on a hypothetical V formation;
//Evolution of leader and following birds
  While Termination criterion is not satisfied **do**
  For $d = 0$ to γ **do**
    % γ is the parameter
  Improve the leader;
  Improve the following birds;
  End For
//Leader changing
  Update the best bird;
  Select leader bird;
  End While

It should be noted that inspired by this special formation, the MBO algorithm does not employ the concepts of the constant angle and depth, but has a hypothetical V-shaped population formed by a leader solution and other solutions in left and right lines following the leader and introduces a benefit mechanism corresponding to the WTS. The MBO starts with a number of solutions placed on the 'V' population arbitrarily. Then, an evolving loop involving a number of tours or iterations, and each tour evolves beginning with the leader and progressing along the left and right lines in parallel by exploring their neighborhood. Particularly, the benefit mechanism that the solutions may share their best unused neighbors with the following solutions through a special neighbor shared set is applied in the evolutionary process. The unused neighbors are referred to the neighbors that are not used to update its current solution. Finally, when a loop is finished, the leader is to be changed, and another loop starts. The above procedure is conducted repeatedly until the termination condition is met.

V. THE PROPOSED EMBO FOR IPS-LD

In this section, we present an enhanced MBO (EMBO) with stronger optimization capabilities to solve the proposed IPS-LD. We first determine the solution representation and employ the decoding rule based on "three-level heuristic-based solution generation". Then, we design several modifications to enhance the performance of the MBO, including the neighborhood search strategies, dynamic solution acceptance criteria, and novel competitive mechanism. With these modifications, the proposed algorithm is expected to capture the balance between the exploration and exploitation abilities, and it performs well in solving IPS-LD. The details are given below. Note that, This algorithm has four parameters: the number of individuals in the swarm (α), the number of neighbors around the leader and following bird (β), the number of neighbors to be shared with the following birds (χ), and the number of iterations before replacing the leader, which is also called the number of tours (γ). The time complexity is $O(\alpha \times \gamma \times \beta \times \frac{\alpha-1}{2} \log_2 \frac{\alpha-1}{2})$.

A. SOLUTION REPRESENTATION AND POPULATION INITIALIZATION

The encoding scheme plays an important part in designing effective algorithms. In the EMBO, an individual (migrating bird) in the flock is a solution to the integration problem. As mentioned in Section II, as long as the production sequence of charges at the first stage is given, the variables including the machine assignment, charge sequence, completion time and makespan can be deduced sequentially using the three-level heuristic-based solution generation. Meanwhile, since the ladle schedule is generated based on a feasible production schedule, the ladle dispatching needs not to be considered in the coding process. Therefore, a permutation-based representation, which expresses the ascending order of release times of charges at the first stage, is enough for encoding the integration problem. That is, each solution is represented by a string of integers in which each integer denotes a charge number and the length of this string equals to the number of charges. Based on the encoding scheme, a solution can be generated by the three-level heuristic, which is described in Section III.

//Flowchart of the EMBO algorithm
Input: Data of the integration problem; parameters of EMBO.
Output: Leader bird with best solution

1: Generate α birds with heuristic-based or random initialization;
2: Put birds on a hypothetical V formation;
3: **If** termination criterion is not satisfied **do**
4: **For** d=0 to γ **do**
// Leader improvement
5: Generate β neighbors for leader bird with neighborhood operators;
6: Improve the leader bird and form the left/right sharing neighbor set respectively with χ neighbors;
// Follower improvement
7: **For** each following bird X **do**
8: Generate β-χ neighbors with the neighborhood operator;
9: Form neighborhood $\Lambda(X)$ with β-χ neighbors and χ shared neighbors;
10: **If** the best neighbor X' in $\Lambda(X)$ is better than X **then**
11: $X = X$';
12: **Else If** random(0,1)$\leq e^{-(\Delta/temp)}$ **then**
13: $X = X$';
14: **End If**
15: **End For**
16: **End If**
// Birds regrouping
17: Update the best bird;
18: Select the leader bird;
19: Line up the followers;
20: **End For**

It is worth noting that at the casting stage, all the ladles belonging to a cast must be processed continuously and be in accordance with the predetermined order of charges in a cast. Moreover, an ineffective coding may result in the interruption of this predetermined order of charges in a cast and further influence the continuity of the casting process. Hence, Rule 3 is proposed to guarantee that all the ladles in a cast can be cast continuously.

Rule 3: The number of casts in process is less than that of casters.

Proof: As mentioned in Section III. (A), all the charges are allocated and sequenced one by one according to breadth first principle in the casting stage. Supposing that the number of casts in process is larger than that of casters at the very moment, the charges belonging to the incumbent cast can be cast on time. However, the others must wait until the *completion of the incumbent cast. Obviously, the idle energy consumption will be largely increased due to wait time of other charges.*

Based on Rule 3, a heuristic-based initialization is employed to guarantee the feasibility of solutions and expedite the optimization process as shown in the following.

Algorithm Initialization Heuristic

Input: A random sequence of casts
Output: A charge processing sequence

Step 1: Generate randomly a sequence of casts.
Step 2: Create a set $A[j]$ with all assignable charges according to the predetermined order of charges in the casts.
Step 3: Randomly select a charge from set $A[j]$ satisfying Rule 3; assign the charge to the sequence; and eliminate the charge from set $A[j]$.
Step 4: Check there are new assignable charges or not. If yes, append them into the set $A[j]$; otherwise, keep the set unchanged.
Step 5: If the set $A[j]$ is empty, output the resulted sequence; otherwise, return to step 2.
Step 6: Next, we obtain the solution representation that is the same as the charge processing sequence selected by the order in sequence $\pi[j]$

For clearly, Table 2 presents the detailed procedure of generating a feasible sequence for the small industrial case in Section 4.4. The resulted feasible sequence is represented as $\{2, 5, 3, 6, 4, 7, 1, 8\}$.

TABLE 2. The Procedure of a feasible sequence.

Remained casts	Assignable charges	Selected charge	Casts in process	Charges set
{1,2,3}	{1,2,5}	2	{2}	2
{1,2,3}	{1,3,5}	5	{2,3}	2,5
{1,2,3}	{1,3,6}	3	{2,3}	2,5,3
{1,2,3}	{1,4,6}	6	{2,3}	2,5,3,6
{1,3}	{1,4,7}	4	{2,3}	2,5,3,6,4
{3}	{1,7}	7	{1,3}	2,5,3,6,4,7
{3}	{1,8}	1	{1,3}	2,5,3,6,4,7,1
{3}	{8}	8	{3}	2,5,3,6,4,7,1,8

Since MBO is a swarm-based algorithm and a swarm of individuals evolves in parallel, we need to generate a population of initial solutions first and put these birds on a distinctive 'V' formation. In this paper, to improve the efficiency of the EMBO, we use the solution generated by the heuristics as the leader. The rest of the solutions are randomly generated. And the rest of solutions in the population are divided into two parts as the followers in the left line P_l and right line P_r.

B. COMBINATIVE NEIGHBORHOOD STRUCTURES

In the basic MBO, the leader bird is updated as long as the current best neighbor shows better performance, and the remaining neighbor solutions are utilized for the evolution of the following birds. In this case, most neighbors of the following birds are inherited from the leader. Obviously, the quality of neighbors of the leader bird has a great influence on the algorithm, and its neighborhood structure should be more exquisite. Hence, we adopt the generally utilized insert and swap operators to generate the neighbor solutions of the following birds, and more importantly, we propose greedy insert and swap operators based on greedy selection to improve the neighborhood of the leader bird.

Insert: remove a charge randomly and reinsert it at any other position except the original one.

Swap: exchange two randomly selected charges.

Greedy insert: select a charge randomly, reinsert it back into the sequence at each position except the original one and obtain multiple sequences, and choose the sequence with the minimum objective value of makespan.

Greedy swap: select a charge randomly, swap it with each other charges in the sequence and obtain multiple sequences, and choose the sequence with the minimum objective value of makespan.

Note that, since the two new neighborhood operators are both for the neighborhood improvement of the leader, we define a random number in (0,1) to choose one operator for fairness. If this number is less than 0.5, choose one operator and otherwise another.

C. NEWLY ACCEPTANCE CRITERION

The basic MBO conducts a greedy selection by which the current solution is substituted if and only if its best neighbor has an improvement. This acceptance criterion might lead to the situation that the incumbent bird remains unchanged. For this reason, escaping from the local optima is in urgent need [40]. And, simulated annealing (SA) is an effective heuristic for achieving good solutions to difficult problems. It mimics the process of cooling solids. Different from common descent algorithms allowing better solution to survive, it also accepts a neighborhood with lower performance at a predefined probability. Hence, we introduce the SA strategy into EMBO with an attempt to escape from the local optimum.

If the best neighbor N_{best} outperforms the incumbent bird X, replace the latter directly; otherwise, the leader bird keeps unchanged but the following birds are replaced by the best neighbor with the probability of $e^{-(\Delta T/temp)}$, where the constant temperature $temp$ is calculated using Equation (30).

$$temp = T \times \sum_i \sum_j PT_{i,j}/(|L| * n * \rho) \tag{30}$$

Clearly, this new acceptance criterion accepts the worse solution with a certain probability, and hence enhances the diversity of population and avoids being trapped in the local optima.

D. COMPETITIVE MECHANISM

In the basic MBO, the birds are put on hypothetical V-shaped formation arbitrarily, and some promising birds may emerge in the tail and have few opportunities to share their neighbors. However, the basic MBO puts birds on V-shaped formation arbitrarily without the differentiation of their performance, and conducts the improvement process of following birds from the bird next to the leader to that at the tail. Pitifully, those promising birds which emerge in the tail may have few opportunities to share their neighbors.

Hence, this study is based on the competitive mechanism proposed by Zhang *et al.* [39] to remedy this possible drawback. In the modified competitive mechanism, first, the leader is moved to the tail of the left or the right line alternatively and the best one in the selected line becomes the new leader. Then, two individuals from the same position of the left and right lines are selected randomly and respectively, and are exchanged to promote the information exchange of the birds on two lines. Subsequently, all the birds in each line are queued again according to the descending order of objective values. The bird with the best fitness is removed to the first position, the bird with the second-best fitness is removed to the second position, and finally the bird with the worst fitness is removed to the last position.

Clearly, this competitive mechanism guarantees that promising birds locate in the front of the line and have more opportunities to share their neighbors. In this study, this mechanism is executed after the selection of leader bird to adjust the position of each following bird in the flock lines.

VI. COMPUTATIONAL EXPERIMENTS

To test the performance of the proposed EMBO, EMBO is compared with four other published algorithms. Although there exist a number of optimization algorithms, they might not be able to solve the considered problem directly and hence this research mainly re-implements the methods applied to the SCC and other production scheduling problems. The methods considered for comparative study are: simulated annealing algorithm (SA) [41], genetic algorithm (GA) [42], teaching–learning-based optimization algorithm (TLBO) [43], artificial bee colony algorithm (ABC) [27]. The main operators of GA, SA, TLBO, and ABC are selected based on the reported ones in [44]. The termination criterion for each case is set as an elapsed CPU time which is set to $|L| \times n \times \rho$ milliseconds, where ρ is a parameter. In order to observe the performance of the algorithms from short to large computational time, ρ is set to 10 and 20, respectively.

The following experimental study is conducted based on a real SCC factory in China, which involves 2 EAFs, 2 LFs, 1 RH and 2 CCs. Limited by the production capacity, the maximum number of casts in a scheduling horizon is about 34 and each cast contains at most 5 ladles. With this, the number of casts, the number of charges belonging to each cast, the number of stages for a ladle, the residual life of each ladle and the processing time are generated using uniform distribution between [1, 34], [1, 5], [3, 4], [5, 80] and [40, 75] respectively. Due to the space limitation, the details of the date are given as on-line materials. It must be pointed out that each ladle in a cast has the exact same processing route.

The proposed mathematical model is programmed with GAMS/CPLEX 23.0 and all the algorithms are codes using C++ programming language, and all the experiments are implemented on a PC with a 2.3 GHZ Intel Core i5 processor in a WIN7 Operation System (64-bit).

A. PARAMETER CALIBRATION

Since parameter settings significantly influence computational results, we utilize the Taguchi method to study the influence of parameters. For EMBO, there are five parameters or controlled factors: the number of initial solutions (α), the number of neighbor solutions to be considered (β), the number of neighbor solutions to be shared (χ), the number of tours (γ) and the initial temperature (T). Based on the parameter calibration method reported in [44], the Taguchi method of design of experiment (DOE) is applied to select the parameter values. Specifically, the largest case with 34 casts and 115 charges is selected and is solved ten times by any combination of the parameter levels, with the termination criterion of $|L| \times n \times 10$ milliseconds.

For all the experiments are conducted, the relative percentage deviation or RPD is selected as the response variable using,

$$\mathrm{RPD} = (\mathrm{TEC_{Some}} - \mathrm{TEC_{Best}})/\mathrm{TEC_{Best}} \times 100. \quad (31)$$

Here, $\mathrm{TEC_{Some}}$ is the function value achieved by a given parameter combination and $\mathrm{TEC_{Best}}$ is the best function value obtained by all combinations for the same instance. We utilize orthogonal array to arrange the experiments which then are run to obtain the corresponding response value. As shown in Table 3, there are $L_{16}(4^5)$ experiment combinations and each experiment is run 10 times. Then, the average RPD is regarded as the final response value. Clearly, the algorithm with a lower RPD has a better performance.

TABLE 3. Orthogonal array of EMBO.

No.	Levels of parameters					Response value
	α	β	χ	γ	T	
1	9	5	3	50	0.4	0.12
2	9	8	4	100	0.5	0.12
3	9	9	5	150	0.45	0.12
4	9	12	6	200	0.6	0.08
5	11	5	4	150	0.6	0.11
6	11	8	3	200	0.45	0.11
7	11	9	6	50	0.5	0.07
8	11	12	5	100	0.4	0.04
9	13	5	5	200	0.5	0.13
10	13	8	6	150	0.4	0.09
11	13	9	3	100	0.6	0.08
12	13	12	4	50	0.45	0.11
13	15	5	3	100	0.45	0.10
14	15	8	5	50	0.6	0.10
15	15	9	4	200	0.4	0.08
16	15	12	6	150	0.5	0.08

After obtaining the response values, the Taguchi method of design of experiment (DOE), a powerful parametric statistical inference tool, is carried out to check the normality, homoscedasticity and independence of the residuals. Detailed

TABLE 4. The mean response values of signal-noise ratio (SNR).

Level	α	β	χ	γ	T
1	18.59	17.67	18.35	19.77	20.26
2	20.63	19.3	18.92	19.58	18.29
3	19.27	20.18	19.29	19.47	19.36
4	19.54	20.88	21.47	19.21	20.13
Delta	2.04	3.21	3.12	0.56	1.98
Rank	3	1	2	5	4

results indicating the significance rank of each parameter are illustrated in Table 4. It is observed that the parameter β, the number of neighbor solutions to be shared, has the largest *delta* (3.21), indicating that β has the greatest influence on the proposed EMBO. If ranking the parameters in the decreasing order of the *delta* values, the sequence is β, χ, α, T, and γ, where the former parameter has greater influence.

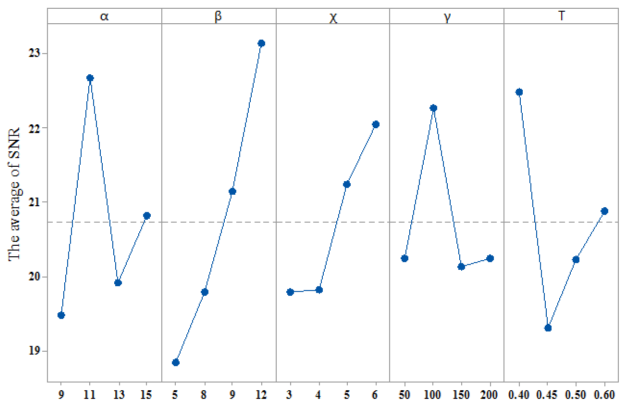


FIGURE 4. SNR main effects plot.

Meanwhile, according to the response value in Table 3, the signal-noise ratio (SNR) main effects plot of the five parameters is illustrated in Figure 4. It demonstrates that the proposed EMBO performs the best when the combination is: ($\alpha = 11$, $\beta = 12$, $\chi = 6$, $\gamma = 100$, $T = 0.4$).

B. SIGNIFICANCE OF INTEGRATION OPTIMIZAITON

1) MICRO-ANALYSIS OF ENERGY-SAVING NATURE

To present clearly the energy-saving nature of the integration method, a typical instance is hired as an illustrative example. It contains five casts with (5, 3, 4, 5, 1) charges in the respective cast sequentially. Note that for clarity, only three to five ladles can be utilized for transportation. The corresponding Gantt charts are presented in Figure 5.

Figure 5(a) shows the optimal makespan is 1000 minutes when three ladles are involved; while it remains at 800 minutes even if more ladles are appended as shown in Figure 5(b-c). In a word, reasonable ladle dispatching plays a significant impact to optimize the production scheduling to a large extent. In nature, the lack of ladles may result in a long delay of the transportation of the molten steel, and of course, the makespan will be extended largely. In order to guarantee productivity, the number of ladles to be utilized is usually preset as a large enough number. As long as the number of ladles is large enough, the optimal makespan remains unchanged. However, as shown by the results of four and five ladles respectively in Figure 5(b-c), the average circulating number of ladles for the former is 4.5 while that for the latter is 3.6; and, the total baking time is 1005 minutes for the former, while that is 1620 minutes for the latter. These results demonstrate that the utilization of an additional ladle not only decreases the average circulating number, and more importantly, it increases the total baking time since a ladle to be utilized cannot wait without baking. Consequently, the incurred energy consumption is increased dramatically as the number of additional ladles grows.

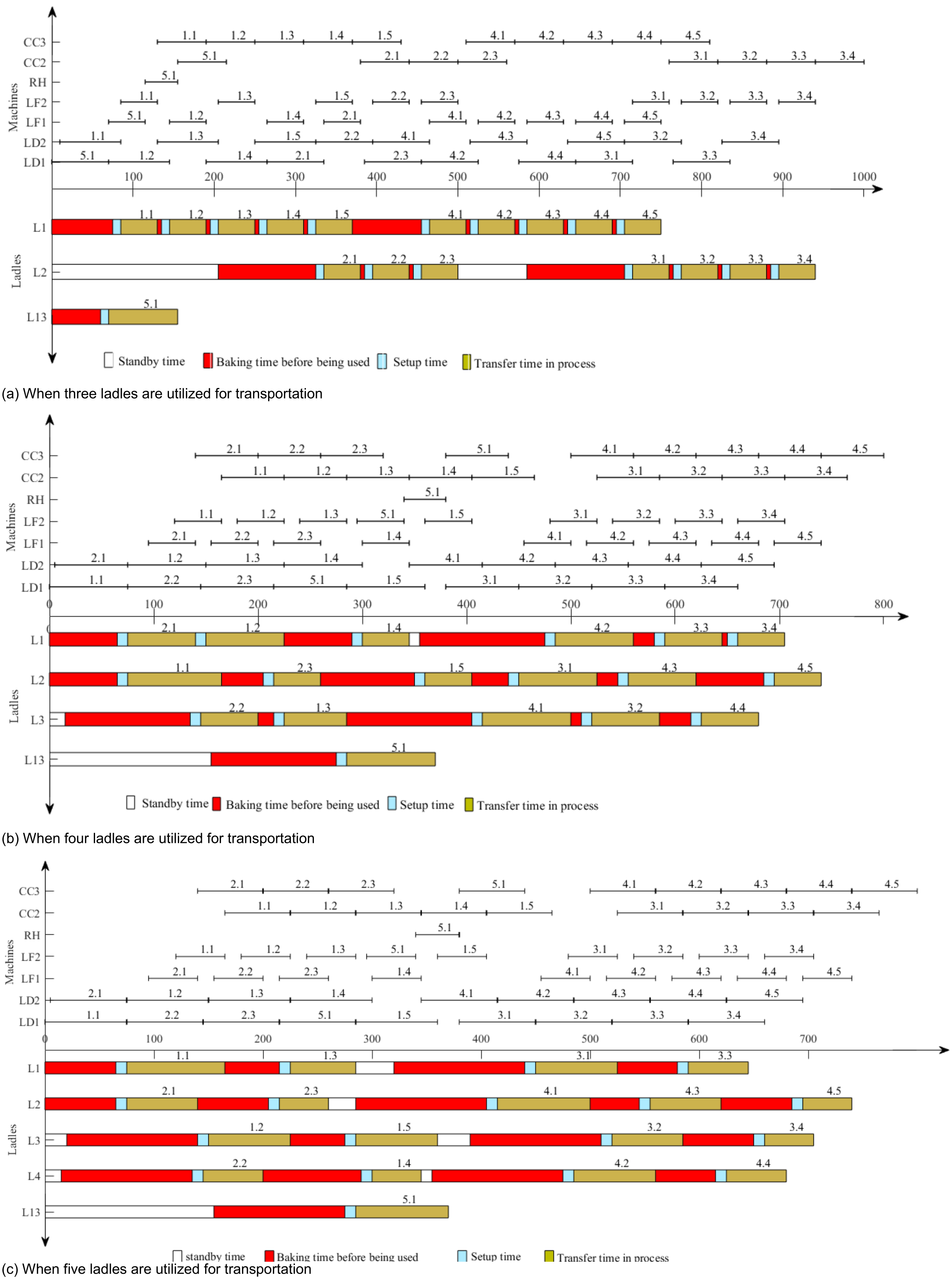
Therefore, to reduce energy consumption, it is recommended to optimize the number of ladles to be utilized and the circulating of ladles according to the production requirements. In another word, incorporating the ladle dispatching into the production scheduling may be an energy-efficient measures to obtain productivity and energy saving simultaneously.

2) MACRO-ANALYSIS OF INTEGRATION OPTIMISATION

Further to confirm the energy-efficient performance of the integration optimization of production scheduling and ladle dispatching, the proposed integration model (IPS-LD) is compared with several separation methods. With respect to the separation methods so far performed on the spot, the optimal production schedule is determined at first similar to [36]; and based on this, the ladles are preheated according to different patterns. In the first pattern denoted as PS-AB, all the ladles are preheated regardless of the initial release times of ladles. In the second pattern denoted as PS-HB, half of the provided ladles are selected for baking while the others stay at the environmental temperature. In the third pattern denoted as PS-JB, all ladles are heated just before the time to be into service.

Hence, in order to verify the optimality performance of the proposed IPS-LD, Table 5 reports the comparison results among these integration and separation methods. In this table, columns (1-2) present the instance number and the number of casts and charges. Note that, each of the tested instances is solved 10 times by the integration or separation methods respectively at the CPU time limit set as $|L| \times n \times \rho$ millisecond ($\rho = 10$). This results in a total of $20 \times 10 \times 1 \times 4 = 800$ experiments. Columns (3-13) report the obtained TEC, Avg and RD calculated by the integration or separation methods. Here, TEC is computed according to "27", *Avg* signifies the mean absolute percentage error of 10 times, and RD reports the relative difference (%) of the incumbent method compared with the proposed IPS-LD, or called the improvement rate of the proposed IPS-LD from the viewpoint of the IPS-LD. This relative difference is calculated with,

$$\mathrm{RD} = (1 - \mathrm{TEC}_{\mathrm{some}}/\mathrm{TEC}_{\mathrm{IPS-LD}}) \times 100, \tag{32}$$



(a) When three ladles are utilized for transportation

(b) When four ladles are utilized for transportation

(c) When five ladles are utilized for transportation

FIGURE 5. The Gantt chart for instance 10 with different available ladles.

in which TEC_{some} is the obtained TEC by any one of the separation methods. And, the average relative difference of all the related cases is reported in the last row.

It can be seen that the value of TEC by IPS-LD is significantly less than those by PS-HB, PS-JB, and PS-AB, indicating that on energy saving, the integration method

TABLE 5. The results for energy consumption comparison with the separation methods.

No.	Casts × Charges	Separation									Integration	
		PS-HB			PS-JB			PS-AB			IPS-LD	
		TEC	Avg	RD	TEC	Avg	RD	TEC	Avg	RD	TEC	Avg
11	10×30	5037	0	1.04	5050	0	1.3	5042	0	1.14	4985	0
12	10×30	5030	0	0.88	5045	0	1.18	5042	0	1.12	4986	0
13	10×30	4982	0	0.83	5002	0	1.23	4987	0	0.93	4941	0
14	10×30	5029	0	0.94	5046	0	1.28	5035	0	1.06	4982	0
15	10×30	5030	0	0.9	5044	0	1.18	5037	0	1.04	4985	0
16	15×47	7786	0.21	0.59	7809	0.23	0.89	7777	0.29	0.48	7740	0.33
17	15×47	7834	0.27	0.35	7866	0.38	0.76	7869	0.34	0.79	7807	0.43
18	15×47	7822	0.43	0.8	7839	0.21	1.02	7831	0.52	0.91	7760	0.35
19	15×47	7840	0.29	0.84	7891	0.38	1.49	7858	0.34	1.07	7775	0.28
20	15×47	7825	0.35	1.43	7852	0.46	1.78	7775	0.19	0.78	7715	0.26
21	20×61	10099	0.34	1.24	10132	0.21	1.57	10089	0.25	1.14	9975	0.54
22	20×61	10175	0.52	1.31	10204	0.12	1.6	10191	0.68	1.47	10043	0.62
23	20×61	10132	0.24	1.05	10158	0.24	1.31	10142	0.34	1.15	10027	0.44
24	20×61	10126	0.17	0.88	10157	0.32	1.19	10150	0.27	1.12	10038	0.37
25	20×61	10095	0.33	0.83	10136	0.27	1.24	10106	0.28	0.94	10012	0.29
26	25×76	12561	0.17	0.94	12605	0.19	1.29	12577	0.54	1.07	12444	0.17
27	25×76	12595	0.32	0.91	12628	0.25	1.18	12612	0.41	1.05	12481	0.37
28	25×76	12604	0.28	0.59	12642	0.31	0.89	12590	0.38	0.48	12530	0.18
29	25×76	12451	0.23	0.34	12503	0.28	0.76	12508	0.42	0.8	12409	0.11
30	25×76	12518	0.31	0.8	12546	0.46	1.02	12532	0.28	0.91	12419	0.36
Average (*RD*)				0.87			1.21			0.97		

TABLE 6. Algorithm performance for small-scaled problems.

Instance *No.*	Casts × Charges	MIP			EMBO	
		TEC	Gap (%)	CPU time (s)	TEC	CPU time (s)
1	3×8	1478*	0	118.71	1478	1.00
2	3×8	1466*	0	115.52	1466	1.00
3	3×8	1451*	0	141.10	1451	1.00
4	3×8	1487*	0	120.61	1487	1.00
5	3×8	1481*	0	123.43	1481	1.00
6	5×18	3069*	0	1187.54	3069	1.00
7	5×18	3076*	0	1150.11	3076	1.00
8	5×18	3084	0.34	1200.00	3076	1.10
9	5×18	3068	0.69	1200.00	3047	1.00
10	5×18	3211	1.17	1200.00	3174	1.12
Average				638.79		1.02

outperforms considerably all the separation methods. Meanwhile, among the separation methods, the PS-JB allows all ladles to be heated just before the service shows worse performance on energy saving than PS-AB and PS-HB in which all or half ladles are preheated. This demonstrates that since the energy consumption is relatively high for warming up a ladle, heating a ladle just before the service causes large energy consumption. And, the average relative difference of the PS-HB is slightly smaller than that of the PS-AB. This confirms that if the number of ladles is relatively large, preheating all ladles no matter if they are needed or not is definitely a waste of energy, and conversely, the recommended method is to preheat the ladles to be utilized just in time and keep most of them in frequent recirculation.

Table 5 also shows that among the four methods for the results, the integration method (IPS-LD) is the top 1, the half-preheated (PS-HB) is the second, the all-preheated (PS-HB) is the third and the way of heating just before the service (PS-JB) is the last. Particularly, compared with PS-JB, the improvement rate of the integration method achieves 1.21%, implying a significant decrease in energy consumption within SCC plants. This finding proves statistically the superiority of the integration optimization of production scheduling and ladle dispatching over the separation methods on the energy savings. Therefore, the ladle utilization is recommended to be integrated in the production scheduling.

In conclusion, the traditional notion of preheating more ladles than the needed is indeed helpful for keeping high productivity but is quite expensive in energy consumption. To decrease energy consumption in SCC production, it is necessary and sufficient to determine when and which ladles to be utilized when making schedules for production. The integration optimization of production scheduling and ladle dispatching is an effective measure for productivity improvement and energy saving.

C. ALGORITHM PERFORMANCE OF EMBO

1) COMPARING WITH CPLEX

To test the validation of the proposed EMBO, the mixed integer programming model (MIP) in Section II is solved in small-scaled instances utilizing the GAMS/CPLEX solver, and the derived results are compared with those by the proposed EMBO. For each instance, the solution procedure by GAMS/CPLEX terminates if an optimal solution is obtained or computation time reaches 1200*s*. Table 6 presents the results solved by two methods in 10 small-scaled instances. In this table, columns 1-2 present the instance number and the number of casts and charges. Columns 3-7 report the obtained best solutions and the consumed CPU times by each method

TABLE 7. Algorithm performance for medium- and large-scaled problems results when $\rho = 10$.

Instance *No.*	TEC_{Best}	GA	SA	TLBO	ABC	MBO	$EMBO_{NS}$	$EMBO_{NC}$	$EMBO_{SC}$	EMBO
		Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg
11	4985	0/0.23	0/0.21	0/0.23	0/0.18	0/0.13	0/0	0/0	0/0	**0/0**
12	4986	0/0.43	0/0.59	0.31/0.75	0/0.34	0/0.35	0/0.07	0/0.23	0/0	**0/0**
13	4941	0/0.37	0/0.31	0/0.22	0/0.3	0/0.19	0/0.12	0/0.06	0/0.1	**0/0**
14	4982	0.31/0.59	0.31/0.37	0.31/0.53	0.25/0.47	0/0.36	0/0.13	0/0.09	0/0.08	**0/0**
15	4985	0/0.65	0/0.66	0/0.5	0/0.52	0/0.17	0/0.23	0/0.24	0/0.11	**0/0**
16	7740	0.53/1.41	0.53/1.41	1.06/1.68	0.42/1.13	0.84/1.28	0/0.34	0/0.36	0/0.21	**0/0.33**
17	7778	0.52/1.16	1.1/1.29	0.94/1.25	0.42/0.93	1.05/1.28	0.37/0.6	0.47/0.82	0.29/0.87	**0/0.43**
18	7748	0.63/1.05	0.84/1.35	0.58/1.28	0.5/0.84	0.58/1	0.36/0.32	0.05/0.48	0.07/0.53	**0/0.35**
19	7775	0.31/0.53	0.31/0.51	0.31/0.57	0.25/0.42	0.31/0.44	0.3/0.38	0.16/0.3	0.18/0.35	**0/0.28**
20	7715	0.89/1.36	0.89/1.33	1.47/1.62	0.71/1.09	0.95/1.28	0.18/0.36	0/0.29	0.02/0.34	**0/0.26**
21	9965	0.78/1.1	0.95/1.33	1.03/1.49	0.62/0.88	0.46/1.34	0.22/0.63	0.12/0.57	0.14/0.62	**0/0.54**
22	10037	0.78/1.2	1.03/1.31	0.82/1.27	0.62/0.96	0.66/0.84	0.68/0.51	0.49/0.73	0.51/0.78	**0/0.62**
23	10012	0.21/1.02	0.99/1.44	1.2/1.47	0.17/0.82	0.37/1.17	0.17/0.89	0.08/0.68	0.1/0.73	**0/0.44**
24	10019	0.41/0.7	0.7/1.04	0.49/0.94	0.33/0.56	0.37/0.61	0.31/0.32	0.21/0.39	0.23/0.44	**0/0.37**
25	10002	0.2/0.57	0.57/1.02	0.61/1.02	0.16/0.46	0.25/0.58	0.22/**0.25**	0.16/0.27	0.18/0.32	**0**/0.29
26	12437	0.34/0.93	1.17/1.64	1.01/1.67	0.27/0.74	0.6/0.84	0.07/0.2	0/0.26	0.04/0.31	**0/0.17**
27	12462	0.9/1.25	1.27/1.7	1.58/1.97	0.72/1	0.64/1.1	0.17/0.35	0/**0.33**	0.02/0.38	**0**/0.37
28	12468	0.67/1.28	1.14/1.46	1.14/1.62	0.54/1.02	1.05/1.44	0.51/0.66	0.07/0.37	0.09/0.42	**0/0.18**
29	12395	0.6/1.2	1.67/1.91	1.14/1.7	0.48/0.96	0.67/1.04	0.13/0.4	0/0.28	0.02/0.33	**0/0.1**
30	12400	0.8/1.35	1/1.7	1.44/1.86	0.64/1.08	0.44/1.22	0.17/0.44	0.07/**0.28**	0.09/0.33	**0**/0.36
31	15852	0.59/1.11	1.13/1.51	1.1/1.58	0.47/0.89	0.67/0.98	0/0.36	0.15/0.37	0.11/**0.32**	**0**/0.38
32	15667	0.75/1.35	1.18/1.5	1.53/1.81	0.6/1.08	0.43/0.93	0.08/0.53	0.03/0.38	0.05/0.43	**0/0.26**
33	15737	0.7/1.03	1.02/1.45	1.29/1.56	0.56/0.82	0.59/0.86	0.31/0.62	0.21/0.46	0.23/0.51	**0/0.35**
34	15710	0.73/1.2	1.05/1.64	1.26/1.64	0.58/0.96	0.75/0.92	0.13/0.74	0.03/0.47	0.05/0.52	**0/0.2**
35	15724	0.62/0.94	1.18/1.34	0.83/1.28	0.5/0.75	0.75/0.99	0.05/0.42	0.05/0.28	0.07/**0.23**	**0**/0.44
36	18946	0.48/0.72	0.82/1.02	1.02/1.2	0.38/0.58	0.43/0.58	0.15/0.44	0.06/0.29	0.08/0.34	**0/0.15**
37	18756	0.52/0.77	0.66/0.99	0.86/1.13	0.42/0.62	0.32/0.6	0.05/0.25	0/0.27	0.02/0.32	**0/0.24**
38	18637	0.32/0.59	0.34/0.85	0.89/1.18	0.26/0.47	0.57/0.85	0.14/0.39	0/**0.16**	0.02/0.21	**0**/0.19
39	18619	0.3/0.64	0.7/0.87	0.59/0.96	0.24/0.51	0.34/0.62	0.11/0.26	0.02/0.13	0.04/**0.12**	**0**/0.14
40	18678	0.3/0.72	0.64/0.9	0.55/1.1	0.24/0.58	0.55/0.77	0.25/0.42	0/0.32	0.02/0.37	**0/0.17**

respectively. Note that, the data with the signal $*$ signifies the optimal solution.

It can be seen that all the algorithms have found the optimal solutions for instances 1-7, which provides a complete validation of the proposed EMBO. Moreover, taking instance 8 as an example, the GAMS/CPLEX cannot find the optimal solution within 1200s while the proposed EMBO finds a much better solution within 2s. This proves the prominent superiority of the proposed EMBO as the problem scale increases. And, the average running time of GAMS/CPLEX is 638.79s while that of EMBO is 1.02s. In summary, the EMBO is capable of obtaining the optimal solutions for small-scaled problems and achieving solutions with higher quality in much less time for the large-scaled problems.

2) COMPARING WITH OTHER ALGORITHMS

To investigate the effectiveness of the proposed EMBO on medium- and large-scaled problems, the obtained results are first compared with those solved by the five existing algorithms: GA, SA, TLBO, ABC and MBO. Further, they are contrasted with the results calculated by three variants of the proposed EMBO: $EMBO_{SC}$ that removes the appended neighborhood, $EMBO_{NC}$ that removes SA-based acceptance criterion and $EMBO_{NS}$ that removes competitive mechanism. Each algorithm solves each of the instances for 10 times at two termination criteria respectively (elapsed CPU time of $|L| \times n \times \rho$ milliseconds and $\rho = 10$ or 20) for each experiment. Hence, a total of 5400 experiments are collected and analyzed. Tables 7-8 present the detailed results by these algorithms. In these tables, TEC_{Best} is the best value obtained by all algorithms for an instance, and Min/Avg signifies the minimum/average relative percentage deviation to the best solution respectively.

It is clear through Tables 7-8 that all the algorithms improve their results when there is more computational effort involved. In terms of the minimum RPD values, the proposed EMBO performs steadily and well, and outperforms all the algorithms for all the instances under two termination criteria. Concerning the average RPD, the proposed EMBO achieves the smallest average RPD in 23 and 27 scenarios of the respective 30 instances when $\rho = 10$ and 20. This demonstrates that in contrast with the comparison algorithms, the proposed EMBO may achieve solutions with better performance and higher robustness. In addition, we also utilize the ANOVA to analyze difference among other algorithms and modifications based on MBO. The least significant differences intervals of the results are depicted in Figures 6-7. These figures clearly illustrate that the proposed EMBO outperforms all the comparing algorithms from a statistical viewpoint.

TABLE 8. Algorithm performance for medium- and large-scaled problems results when $\rho = 20$.

Instance *No.*	TEC_{Best}	GA	SA	TLBO	ABC	MBO	$EMBO_{NS}$	$EMBO_{NC}$	$EMBO_{SC}$	EMBO
		Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg	Min/Avg
11	4985	0/0.13	0/0.12	0/0.34	0/0	0/0.13	0/0	0/0	0/0	**0/0**
12	4986	0/0.35	0/0.44	0/0.87	0/0.07	0/0.35	0/0.07	0/0.23	0/0	**0/0**
13	4941	0/0.19	0/0.06	0/0.33	0/0	0/0.19	0/0.12	0/0.06	0/0.1	**0/0**
14	4982	0/0.36	0/0.31	0.31/0.81	0/0.03	0/0.36	0/0.13	0/0.09	0/0.08	**0/0**
15	4985	0/0.17	0/0.34	0/0.46	0/0.03	0/0.17	0/0.23	0/0.24	0/0.11	**0/0**
16	7740	0.84/1.28	0.79/1.54	0.79/1.53	0/0.54	0.84/1.28	0/0.34	0/0.36	0/0.21	**0/0.17**
17	7778	1.05/1.28	1.21/1.42	0.79/1.2	0.37/0.86	1.05/1.28	0.37/0.6	0.17/0.82	0.09/0.87	**0/0.73**
18	7748	0.58/1	1/1.19	0.94/1.33	0.16/0.42	0.58/1	0.36/0.32	0.05/0.38	0.07/0.53	**0/0.31**
19	7775	0.31/0.44	0.31/0.46	0.26/0.48	0/0.38	0.31/0.44	0.3/0.38	0.16/0.3	0.18/0.35	**0/0.24**
20	7715	0.95/1.28	0.42/1.07	0.63/1.27	0/0.16	0.95/1.28	0.18/0.36	0/0.15	0.02/0.34	**0/0.14**
21	9963	0.46/1.34	1.45/1.75	1.66/1.83	0.12/0.63	0.46/1.34	0.22/0.63	0.12/0.57	0.14/0.62	**0/0.25**
22	10035	0.66/0.84	0.86/1.12	0.86/1.1	0.18/0.61	0.36/0.84	0.28/0.61	0.09/0.53	0.11/0.58	**0/0.41**
23	10010	0.37/1.17	1.41/1.55	1.41/1.65	0.17/0.89	0.37/1.17	0.17/0.89	0.08/0.68	0.1/0.73	**0/0.54**
24	10017	0.37/0.61	0.57/0.83	0.37/0.78	0.21/0.32	0.37/0.61	0.31/0.62	0.21/0.39	0.13/0.34	**0/0.28**
25	10000	0.25/0.58	0.61/0.93	0.61/0.89	0.22/0.25	0.25/0.58	0.22/0.35	0.16/0.27	0.18/0.34	**0/0.23**
26	12435	0.6/0.84	1.44/1.73	1.04/1.6	0.07/0.2	0.6/0.84	0.07/0.29	0/0.26	0.04/0.21	**0/0.13**
27	12460	0.64/1.1	0.84/1.49	1.07/1.53	0.17/0.35	0.64/1.1	0.17/0.55	0/0.37	0.02/0.38	**0/0.34**
28	12466	1.05/1.44	1.42/1.77	0.78/1.79	0.51/0.66	1.05/1.44	0.51/0.66	0.06/0.47	0.09/**0.42**	**0**/0.46
29	12393	0.67/1.04	1.24/1.57	1.14/1.64	0.13/0.4	0.67/1.04	0.13/0.47	0/0.38	0.02/0.33	**0/0.33**
30	12398	0.44/1.22	1.31/1.6	1.47/1.86	0.17/0.44	0.44/1.22	0.17/0.44	0.07/0.28	0.09/0.33	**0/0.26**
31	15850	0.67/0.98	1.05/1.31	1.15/1.42	0/0.66	0.67/0.98	0/0.46	0.13/0.37	0.15/0.42	**0/0.34**
32	15665	0.43/0.93	0.88/1.19	1.12/1.38	0.09/0.3	0.43/0.93	0.08/0.3	0.03/0.38	0.05/0.43	**0/0.2**
33	15735	0.59/0.86	0.83/1.24	0.94/1.44	0.11/**0.32**	0.59/0.86	0.21/0.72	0.11/0.56	0.13/0.51	**0**/0.52
34	15708	0.75/0.92	0.94/1.29	1.07/1.46	0.13/0.34	0.75/0.92	0.13/0.54	0.03/0.47	0.05/0.52	**0/0.28**
35	15722	0.75/0.99	0.91/1.18	1.05/1.33	0.1/0.22	0.75/0.99	0.09/0.42	0.05/0.28	0.07/0.33	**0/0.23**
36	18944	0.43/0.58	0.59/0.89	0.48/0.93	0.19/0.14	0.43/0.58	0.15/0.44	0.06/0.29	0.08/0.34	**0/0.08**
37	18754	0.32/0.6	0.34/0.76	0.8/0.98	0.15/0.25	0.32/0.6	0.05/0.35	0/0.27	0.02/0.32	**0/0.22**
38	18635	0.57/0.85	0.8/1.02	0.91/1.28	0.14/**0.39**	0.57/0.85	0.14/0.89	0/0.46	0.02/0.61	**0**/0.5
39	18617	0.34/0.62	0.77/0.93	0.68/1.03	0.11/0.26	0.34/0.62	0.11/0.36	0.02/0.23	0.04/0.28	**0/0.17**
40	18624	0.55/0.77	0.55/1.06	0.91/1.26	0.25/**0.42**	0.55/0.77	0.25/0.65	0/0.52	0.02/0.54	**0**/0.43

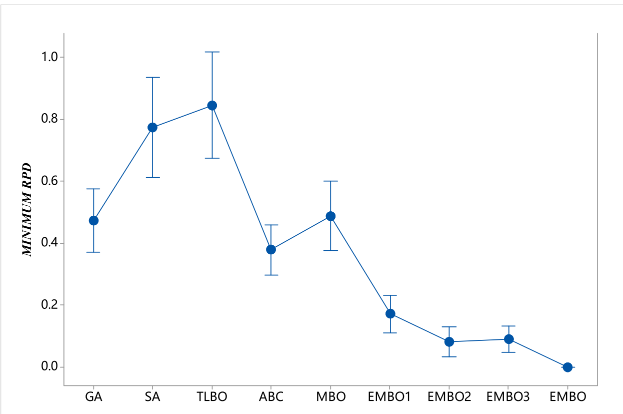


FIGURE 6. 95% Confidence interval for the minimum RPD of EMBO and comparing algorithms at $\rho = 10$.

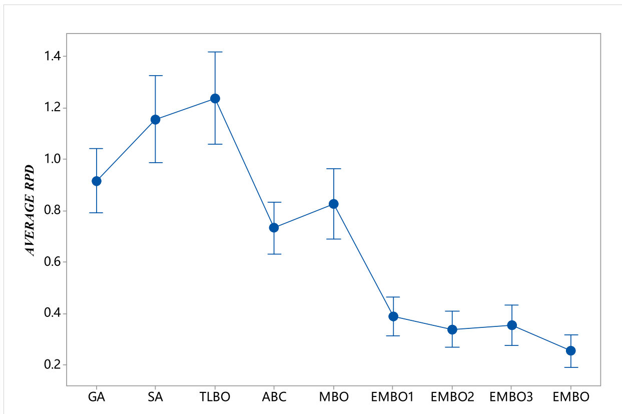


FIGURE 7. 95% Confidence interval for the average RPD of EMBO and comparing algorithms at $\rho = 10$.

To find out why EMBO behaves better, further comparison experiments are conducted between EMBO and its three variants. The results also show clearly that the proposed EMBO outperforms all its variants. Specifically, the $EMBO_{NS}$ has the worst performance for most of the problems in Tables 7-8, indicating that the SA-based acceptance criterion and competitive mechanism have better ability in escaping from local optima and enhancing the population diversity. Under this circumstance, the two appended neighborhood structures further balance the exploitation and exploration. This observation reveals that the designed modifications, including two appended neighborhood structures, SA-based acceptance criterion and competitive mechanism, are effective to enhance the optimization capability of the EMBO. Hence, it is concluded that the proposed EMBO is very effective in solving the energy-efficient scheduling problem considered in this work.

In summary, the integration of production scheduling and ladle dispatching is an imperative and efficient measure to promote productivity improvement and energy saving simultaneously in SCC plants. And, the proposed EMBO with some reasonable modifications solves the integration problem effectively and efficiently.

VII. CONCLUSION AND FUTURE RESEARCH

This paper addresses the integration problem of production scheduling and ladle dispatching (IPS-LD) with the objective of the total completed time and the energy compulsion related to the ladle baking. To solve this problem, a mixed integer linear programming model is formulated to coordinate the relationship of production scheduling and ladle dispatching, which includes time-dependent functions at the joint-point of them and the energy consumption quantification of processing and transportation. Furthermore, an enhanced migrating birds optimization method (EMBO) with the acquired expertise and stronger optimization capability is developed to solve this NP-hardness problem. And the experimental study involving 6200 comparison experiments draws three conclusions:

(1) Due to characteristics of high temperature and high energy consumption in production and transportation within SCC plants, a reasonable integration of production scheduling and ladle dispatching is an effective measure for productivity improvement and energy saving. The proposed integration optimization decreases energy consumption by 1.21% in the context of constant productivity efficiency.

(2) The acquired expertise and three proposed modifications, including the three-level heuristic-based solution generation, two novel neighborhood structures, a new simulated annealing-based acceptance criterion and a novel competitive mechanism, improves the performance of EMBO in reducing the energy consumption of the integration problem.

(3) The proposed EMBO outperforms the state-of-the-art algorithms in the small-, middle- and large-scaled integration problem.

Future research might extend the integration model into more generalized hybrid flow-shop problem. In addition, as the proposed algorithm produces competing performance, it could be employed to reduce energy consumption in other scheduling problems such as assembly line balancing and job-shop scheduling. Moreover, the proposed MBO might be criticized by having many parameters needed to calibrated and future study might develop the self-adapted MBO to tune the parameters automatically.

REFERENCES

[1] K. He and L. Wang, "A review of energy use and energy-efficient technologies for the iron and steel industry," *Renew. Sustain. Energy Rev.*, vol. 70, pp. 1022–1039, Apr. 2017, doi: 10.1016/j.rser.2016.12.007.

[2] J. Li, P. Duan, H. Sang, S. Wang, Z. Liu, and P. Duan, "An efficient optimization algorithm for resource-constrained steelmaking scheduling problems," *IEEE Access*, vol. 6, pp. 33883–33894, 2018, doi: 10.1109/Access.2018.2840512.

[3] B. Lu, G. Chen, D. Chen, and W. Yu, "An energy intensity optimization model for production system in iron and steel industry," *Appl. Thermal Eng.*, vol. 100, pp. 285–295, May 2016, doi: 10.1016/j.applthermaleng.2016.01.064.

[4] W. Xu, L. Tang, and E. N. Pistikopoulos, "Modeling and solution for steelmaking scheduling with batching decisions and energy constraints," *Comput. Chem. Eng.*, vol. 116, pp. 368–384, Aug. 2018, doi: 10.1016/j.compchemeng.2018.03.010.

[5] Z. Xu, Z. Zheng, and X. Gao, "Energy-efficient steelmaking-continuous casting scheduling problem with temperature constraints and its solution using a multi-objective hybrid genetic algorithm with local search," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106554, doi: 10.1016/j.asoc.2020.106554.

[6] E. Mousa, C. Wang, J. Riesbeck, and M. Larsson, "Biomass applications in iron and steel industry: An overview of challenges and opportunities," *Renew. Sustain. Energy Rev.*, vol. 65, pp. 1247–1266, Nov. 2016, doi: 10.1016/j.rser.2016.07.061.

[7] A. Matzke, T. Volling, and T. S. Spengler, "Upgrade auctions in build-to-order manufacturing with loss-averse customers," *Eur. J. Oper. Res.*, vol. 250, no. 2, pp. 470–479, Apr. 2016, doi: 10.1016/j.ejor.2015.11.003.

[8] S.-L. Jiang, M. Liu, J.-H. Lin, and H.-X. Zhong, "A prediction-based online soft scheduling algorithm for the real-world steelmaking-continuous casting production," *Knowl.-Based Syst.*, vol. 111, pp. 159–172, Nov. 2016, doi: 10.1016/j.knosys.2016.08.010.

[9] D. Armellini, P. Borzone, S. Ceschia, L. Di Gaspero, and A. Schaerf, "Modeling and solving the steelmaking and casting scheduling problem," *Int. Trans. Oper. Res.*, vol. 27, no. 1, pp. 57–90, Sep. 2018, doi: 10.1111/itor.12595.

[10] Y. Tan, T. C. E. Cheng, and M. Ji, "A multi-objective scatter search for the ladle scheduling problem," *Int. J. Prod. Res.*, vol. 52, no. 24, pp. 7513–7528, Dec. 2014, doi: 10.1080/00207543.2014.939238.

[11] Y. Tan, W. Zhen, S. Wang, Z. Wei, and S. Liu, "Optimization algorithm for ladle scheduling based on the VRPTW-AT model," *J. Syst. Eng.*, vol. 20, no. 1, pp. 94–100, 2013, doi: 10.3969/j.issn.1000-5781.2013.01.013.

[12] B. F. Huang, N. Y. Tian, Z. Shi, and Z. W. Ma, "Steel ladle exchange models during steelmaking and continuous casting process," *J. Iron Steel Res. Int.*, vol. 24, no. 6, pp. 617–624, Jun. 2017, doi: 10.1016/S1006-706x(17)30093-6.

[13] S. Yu, T. Chai, and Y. Tang, "An effective heuristic rescheduling method for steelmaking and continuous casting production process with multirefining modes," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 12, pp. 1675–1688, Dec. 2016, doi: 10.1109/Tsmc.2016.2604081.

[14] X. Wang, K. Xing, X. Li, and J. Luo, "An estimation of distribution algorithm for scheduling problem of flexible manufacturing systems using Petri nets," *Appl. Math. Model.*, vol. 55, pp. 776–788, Mar. 2018, doi: 10.1016/j.apm.2017.11.018.

[15] H. Cui, X. Luo, and Y. Wang, "Scheduling of steelmaking-continuous casting process using deflected surrogate Lagrangian relaxation approach and DC algorithm," *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106271, doi: 10.1016/J.Cie.2020.106271.

[16] F. Xiong, K. Xing, and F. Wang, "Scheduling a hybrid assembly-differentiation flowshop to minimize total flow time," *Eur. J. Oper. Res.*, vol. 240, no. 2, pp. 338–354, Jan. 2015, doi: 10.1016/j.ejor.2014.07.004.

[17] M. Shakeri, M. Y. H. Low, S. J. Turner, and E. W. Lee, "An efficient incremental evaluation function for optimizing truck scheduling in a resource-constrained crossdock using metaheuristics," *Expert Syst. Appl.*, vol. 45, pp. 172–184, Mar. 2016, doi: 10.1016/j.eswa.2015.09.041.

[18] Q.-K. Pan, "An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling," *Eur. J. Oper. Res.*, vol. 250, no. 3, pp. 702–714, May 2016, doi: 10.1016/j.ejor.2015.10.007.

[19] M. H. F. Zarandi and F. Dorry, "A hybrid fuzzy PSO algorithm for solving steelmaking-continuous casting scheduling problem," *Int. J. Fuzzy Syst.*, vol. 20, no. 1, pp. 219–235, Jan. 2018, doi: 10.1007/s40815-017-0331-0.

[20] K. Mao, Q.-K. Pan, X. Pang, and T. Chai, "A novel Lagrangian relaxation approach for a hybrid flowshop scheduling problem in the steelmaking-continuous casting process," *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 51–60, Jul. 2014, doi: 10.1016/j.ejor.2013.11.010.

[21] D. Oliveira and A. Pessoa, "An improved branch-cut-and-price algorithm for parallel machine scheduling problems," *INFORMS J. Comput.*, vol. 32, no. 1, pp. 90–100, Jan. 2020, doi: 10.1287/ijoc.2018.0854.

[22] J. Gmys, M. Mezmaz, N. Melab, and D. Tuyttens, "A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem," *Eur. J. Oper. Res.*, vol. 284, no. 3, pp. 814–833, Aug. 2020, doi: 10.1016/j.ejor.2020.01.039.

[23] J. G. Peng, M. Z. Liu, X. Zhang, and L. Ling, "Hybrid heuristic algorithm for multi-objective scheduling problem," *J. Syst. Eng. Electron.*, vol. 30, no. 2, pp. 327–342, Apr. 2019, doi: 10.21629/Jsee.2019.02.12.

[24] J.-P. Yang, B.-L. Wang, Q. Liu, M. Guan, T.-K. Li, S. Gao, W.-D. Guo, and Q. Liu, "Scheduling model for the practical steelmaking-continuous casting production and heuristic algorithm based on the optimization of 'furnace-caster matching' mode," *ISIJ Int.*, vol. 60, no. 6, pp. 1213–1224, Jun. 2020, doi: 10.2355/isijinternational.ISIJINT-2019-423.

[25] Z. Wei, T. Zhu, T. He, and S. Liu, "A fast heuristic algorithm for ladle scheduling based on vehicle routing problem with time windows model," *ISIJ Int.*, vol. 54, no. 11, pp. 2588–2597, 2014, doi: 10.2355/isijinternational.54.2588.

[26] J.-Q. Li, Q.-K. Pan, and K. Mao, "A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems," *Eng. Appl. Artif. Intell.*, vol. 37, pp. 279–292, Jan. 2015, doi: 10.1016/j.engappai.2014.09.015.

[27] K. Peng, Q. Pan, and B. Zhang, "An improved artificial bee colony algorithm for steelmaking–refining–continuous casting scheduling problem," *Chin. J. Chem. Eng.*, vol. 26, no. 8, pp. 1727–1735, Aug. 2018, doi: 10.1016/j.cjche.2018.06.008.

[28] A. Sbihi and M. Chemangui, "A genetic algorithm for the steel continuous casting with inter-sequence dependent setups and dedicated machines," *RAIRO-Oper. Res.*, vol. 52, nos. 4–5, pp. 1351–1376, Oct. 2018, doi: 10.1051/ro/2018023.

[29] Y. Han, J.-Q. Li, D. Gong, and H. Sang, "Multi-objective migrating birds optimization algorithm for stochastic lot-streaming flow shop scheduling with blocking," *IEEE Access*, vol. 7, pp. 5946–5962, 2019, doi: 10.1109/ACCESS.2018.2889373.

[30] B. Zhang, Q.-K. Pan, L. Gao, X.-L. Zhang, H.-Y. Sang, and J.-Q. Li, "An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming," *Appl. Soft Comput.*, vol. 52, pp. 14–27, Mar. 2017, doi: 10.1016/j.asoc.2016.12.021.

[31] T. Meng, Q.-K. Pan, J.-Q. Li, and H.-Y. Sang, "An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem," *Swarm Evol. Comput.*, vol. 38, pp. 64–78, Feb. 2018, doi: 10.1016/j.swevo.2017.06.003.

[32] L. Gao and Q.-K. Pan, "A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem," *Inf. Sci.*, vol. 372, pp. 655–676, Dec. 2016, doi: 10.1016/j.ins.2016.08.046.

[33] D. Oz, "An improvement on the migrating birds optimization with a problem-specific neighboring function for the multi-objective task allocation problem," *Expert Syst. Appl.*, vol. 67, pp. 304–311, Jan. 2017, doi: 10.1016/j.eswa.2016.09.035.

[34] A. Allahverdi and H. Aydilek, "Total completion time with makespan constraint in no-wait flowshops with setup times," *Eur. J. Oper. Res.*, vol. 238, no. 3, pp. 724–734, Nov. 2014, doi: 10.1016/j.ejor.2014.04.031.

[35] A. Hou, S. Jin, H. Harmuth, and D. Gruber, "A method for steel ladle lining optimization applying thermomechanical modeling and taguchi approaches," *JOM*, vol. 70, no. 11, pp. 2449–2456, Nov. 2018, doi: 10.1007/s11837-018-3063-1.

[36] Y. Ye, J. Li, Z. Li, Q. Tang, X. Xiao, and C. A. Floudas, "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.*, vol. 66, pp. 165–185, Jul. 2014, doi: 10.1016/j.compchemeng.2014.02.028.

[37] M. N. Janardhanan, Z. Li, G. Bocewicz, Z. Banaszak, and P. Nielsen, "Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times," *Appl. Math. Model.*, vol. 65, pp. 256–270, Jan. 2019, doi: 10.1016/j.apm.2018.08.016.

[38] E. Duman, M. Uysal, and A. F. Alkaya, "Migrating birds optimization: A new Metaheuristic approach and its performance on quadratic assignment problem," *Inf. Sci.*, vol. 217, pp. 65–77, Dec. 2012, doi: 10.1016/j.ins.2012.06.032.

[39] Z. Zhang, Q. Tang, D. Han, and Z. Li, "Enhanced migrating birds optimization algorithm for U-shaped assembly line balancing problems with workers assignment," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7501–7515, Nov. 2019, doi: 10.1007/s00521-018-3596-9.

[40] Q.-K. Pan and Y. Dong, "An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimisation," *Inf. Sci.*, vol. 277, pp. 643–655, Sep. 2014, doi: 10.1016/j.ins.2014.02.152.

[41] C. Chen and L. K. Tiong, "Using queuing theory and simulated annealing to design the facility layout in an AGV-based modular manufacturing system," *Int. J. Prod. Res.*, vol. 57, no. 17, pp. 5538–5555, Sep. 2019, doi: 10.1080/00207543.2018.1533654.

[42] S. Lin, F. Luan, Z. Han, X. Lü, X. Zhou, and W. Liu, "Genetic algorithm based on duality principle for bilevel programming problem in steelmaking production," *Chin. J. Chem. Eng.*, vol. 22, no. 7, pp. 742–747, Jul. 2014, doi: 10.1016/j.cjche.2014.05.006.

[43] Z. Li, X. Zhang, J. Qin, and J. He, "A reformative teaching–learning-based optimization algorithm for solving numerical and engineering design optimization problems," *Soft Comput.*, vol. 24, no. 20, pp. 15889–15906, Oct. 2020, doi: 10.1007/s00500-020-04918-4.

[44] Z. Zhang, Q. Tang, Z. Li, and L. Zhang, "Modelling and optimisation of energy-efficient U-shaped robotic assembly line balancing problems," *Int. J. Prod. Res.*, vol. 57, no. 17, pp. 5520–5537, Sep. 2019, doi: 10.1080/00207543.2018.1530479.

DAYONG HAN was born in Xiaogan, China, in 1990. He received the master's degree in mechanical design and theory from the Wuhan University of Science and Technology, in 2017, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include intelligent optimization algorithms and exact algorithms within flow-shop scheduling problems.

QIUHUA TANG was born in Lichuan, Hubei, China, in 1970. She received the B.S. degree in process and equipment of machinery manufacturing from Northeastern University, in 1992, and the master's degree in mechanical design and theory and the Ph.D. degree from the Wuhan University of Science and Technology.

Since 1992, she has been working with the Wuhan University of Science and Technology, and was promoted to Professor, in 2009. Her research interests include production process planning and scheduling, manufacturing process monitoring and control, and modern optimization methods and algorithms. She has published more than 100 papers in academic journals and conferences. She has undertaken three general research projects supported by the National Natural Science Foundation of China (NSFC), and won the 2016 honorable mention in mechanical engineering discipline of NSFC, in December 2017. She also received awards, like Wuhan Excellent Paper Award and Excellent Paper of Chinese Mechanical Engineering Society.

ZIKAI ZHANG was born in Xiangyang, Hubei, China, in 1994. He received the B.S. degree in industrial engineering from the Wuhan University of Science and Technology, in 2016, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include production process control and intelligent algorithms.

JUN CAO was born in Chongqing, China, in 1980. He received the B.S. and master's degrees in mechanical design and theory from the Wuhan University of Science and Technology. He is currently pursuing the Ph.D. degree in mechanical engineering with the Huazhong University of Technology. His research interests include production process control, flexible job shop scheduling, and intelligent algorithms.