J. Iron Steel Res. Int. (2021) 28:520–529
https://doi.org/10.1007/s42243-021-00578-2

ORIGINAL PAPER

Hybrid optimization for charge planning problem in twin strands continuous casting production

Jian Yi[1,2] · Shu-jin Jia[2] · Bin Du[1,2]

Received: 21 January 2020 / Revised: 7 July 2020 / Accepted: 8 July 2020 / Published online: 21 April 2021

Abstract

Charge planning is one of batching problems for steelmaking and continuous casting production, and its optimization will be conducive to subsequent cast planning. Charge planning problem in the twin strands continuous casting production was studied, where casting width of the odd strand might be different from that of the even strand. Considering the different widths in the twin strands, the resulting counterweights and the constraints of steelmaking and continuous casting, a multi-objective optimization model was established to minimize the number of charges, the number of scale pairs, the surplus and the upgrading costs of steel grades. Furthermore, a hybrid optimization algorithm combined with heuristic and mutation-based estimation of distribution algorithm was proposed to solve the model. Experiments were conducted on several groups of test data collected from practical production orders of Baosteel. The computational results demonstrate that the proposed algorithm can generate better solutions than the manual method. The proposed model and algorithm proved to be effective and practical.

Keywords Steelmaking · Continuous casting · Production planning · Charge · Cast · Estimation of distribution algorithm

1 Introduction

Steelmaking and continuous casting (SCC) are two critical parts in steel manufacturing that are closely related [1, 2]. In the SCC production process, effective planning and scheduling are crucial to raise productivity and have been widely studied [3]. Tang and Wang [4] divided the SCC production into two types of decisions: scheduling decisions and batching decisions. The scheduling decisions are production scheduling problems, while the batching decisions are production planning problems, which include charge planning and cast planning. Charge planning refers to that orders are converted into dummy slabs by the production design, and then, dummy slabs are consolidated into charges according to the converter's capacity for smelting. Cast planning refers to that charges are grouped into casts according to the rules of SCC.

There have been many studies on production scheduling problems [5–14] and production planning problems [15–24] for the SCC process. The methods to study the SCC production planning mainly include mathematical programming [15], exact algorithm [16, 17], heuristic algorithm [4, 18, 19], intelligence optimization [20–22] and hybrid optimization [23, 24].

As for mathematical programming, Bellabdaoui and Teghem [15] presented a mixed-integer linear programming model for the continuous casting planning and used standard software packages to solve it. As for exact algorithm, Tang et al. [16] studied an integrated charge batching and casting width selection problem and developed a column generation-based branch-and-price solution approach. Yi et al. [17] established a multi-objective model for the continuous casting production planning problem and presented a column generation-based optimization algorithm. As for heuristic algorithm, Tang and Wang [4] investigated two batching problems for steelmaking and continuous casting production, formulated them as integer programming models and then developed two heuristic algorithms for the corresponding batching

✉ Jian Yi
yijian@baosteel.com

[1] College of Information Science and Engineering, Northeastern University, Shenyang 110819, Liaoning, China

[2] Institute of Intelligent Manufacturing, Baosteel Central Research Institute, Shanghai 201900, China

problems. Dong et al. [18] proposed a multi-objective model for the integrated tundish planning problem and then developed two improved variable neighborhood local search methods. Song [19] generated an integrated formulation for the cast design problem by introducing multilayer network representation and derived a heuristic algorithm. As for intelligence optimization, Liu et al. [20] formulated a multi-objective order-planning model for manufacturing steel sheets, transferred the multiple objectives into a single one by the weighted-sum approach and designed a specific particle swarm optimization algorithm. Li et al. [21] proposed an effective fruit fly optimization algorithm to solve the steel-making casting problem. Lin et al. [22] constructed a multi-objective optimization model for the integrated production planning in the steel industry and then proposed a novel approach based on a modified interval multi-objective optimization evolutionary algorithm. As for hybrid optimization, Zhu et al. [23] proposed a novel optimization model combined with parallel-backward inferring algorithm and genetic algorithm to improve the efficiency and performance for production planning in steelmaking and continuous casting process. Sun et al. [24] proposed a three-level batching policy consisting of charge planning, tundish planning and cast planning and then developed a hybrid optimal strategy combined with the iterated local search, variable neighborhood search and ant colony optimization.

This paper focuses on charge planning problem and uses a hybrid optimization method to solve it. Charge planning problem is important in the SCC production, which should consider the requirements of converters and casters as well. For a converter, it is required that the ordered slabs combined in a charge have the same steel grade and the total mass of ordered slabs cannot exceed the upper bound of the converter's capacity. If the total mass of ordered slabs in a charge is less than the lower bound of the converter's capacity, the open ordered slabs are filled in the charge. The open ordered slabs are also called the surplus [17], with which there are no corresponding orders. For a caster, it is required that the thickness should be the same and the width range should be intersected if the ordered slabs are consolidated together into a charge. This paper extends the problem by considering the twin strands continuous casting production and allowing different casting widths between the odd strand and the even strand.

2 Charge planning problem in twin strands continuous casting production

2.1 Twin-strand caster

The caster can be classified as a single-strand caster and a multi-strand caster. One typical multi-strand caster is the twin-strand caster (Fig. 1), whose two strands are called the odd strand and the even strand. The two strands may have different casting widths denoted by W^{odd} and W^{even}, but they must keep the balance of time in the continuous casting production. It means that the two strands should start and end simultaneously in the continuous casting production. In this case, the total mass of slabs in the odd strand and the total mass of slabs in the even strand in a charge are defined as the counterweights denoted by G^{odd} and G^{even}.

Generally, a caster has many scales because its available casting widths are scaled by the multiple of 50 mm. For the odd strand and the even strand of a twin-strand caster, the number of scales is the same. The casters at Baosteel have 24 scales, and their casting widths vary from 900 to 2050 mm. For scale $r \in \{1, \ldots, 24\}$, the corresponding casting width is $(850 + 50r)$ mm.

2.2 Counterweight calculation

It is assumed that G is the capacity of converter, while $G^{\min}$ and $G^{\max}$ are the lower bound and the upper bound of converter's capacity, respectively. For each charge, the total mass of slabs in the odd strand and the one in the even strand are the same if the casting widths of two strands are the same. Otherwise, they are not the same. To maintain the balance of time between two strands, the counterweights need to be calculated. The counterweights are related not only to the casting widths of two strands, but also to the casting speeds. The casting speed relies on the casting width, and each casting width can correspond to a fixed casting speed. The counterweights are calculated as follows:

$$\begin{cases} G^{\mathrm{odd}} = \dfrac{W^{\mathrm{odd}} \cdot v^{\mathrm{odd}}}{W^{\mathrm{odd}} \cdot v^{\mathrm{odd}} + W^{\mathrm{even}} \cdot v^{\mathrm{even}}} \cdot G \\ G^{\mathrm{even}} = \dfrac{W^{\mathrm{even}} \cdot v^{\mathrm{even}}}{W^{\mathrm{odd}} \cdot v^{\mathrm{odd}} + W^{\mathrm{even}} \cdot v^{\mathrm{even}}} \cdot G \end{cases} \tag{1}$$

where v^{odd} and v^{even} are casting speeds of the odd strand and the even strand, respectively.

2.3 Problem description

The dummy slabs with the same thickness derived from n orders are required to be consolidated into charges that are arranged to produce in the twin-strand caster. For order $i \in \{1, \ldots, n\}$, the set of its steel grades is marked as S_i, the casting wide range is $W_i^{\min}$–$W_i^{\max}$, the single slab mass is G_i, and the demand for slabs is D_i. For each charge, casting width of the odd strand may be different from that of the even strand; thus, it needs to determine in which charge, at which strand and on what width every dummy slab should

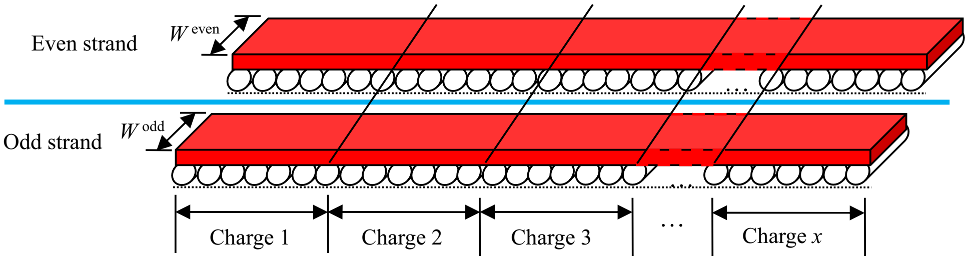

Fig. 1 Schematic of a twin-strand caster

be consolidated. The goal is to minimize the surplus, the number of casting widths and the upgrading costs of steel grades. Specifically, for the dummy slabs that are consolidated into the same charge, the requirements to be met are as follows:

(1) Steel grades must be the same;
(2) The total mass of slabs is not more than the upper bound of the converter's capacity;
(3) The width range should be intersected and can correspond to one scale of the caster at least if arranged on the same strand;
(4) The counterweights are required.

3 Mathematical modeling

3.1 Pattern

To facilitate modeling, a pattern is firstly defined which is the set of charges with the same steel grade, the same casting width of the odd strand, and the same casting width of the even strand. It is assumed that there are m steel grades for these n orders and q scales for the twin-strand caster, and then, the number of patterns is mq^2, where $m = \left|S^{\text{all}}\right|$ and $S^{\text{all}} = \bigcup_{i=1}^{n} S_i$. Because of the symmetry between the odd strand and the even strand, the number of patterns can be reduced to $mq(q+1)/2$ and all patterns can be listed.

Each pattern has a unique steel grade, odd strand's casting width and even strand's casting width. For example, pattern $((u-1)\cdot q(q+1)/2 + r(r-1)/2 + r')$ indicates that steel grade S_u^{all} is used in this pattern, casting width of the odd strand is $(850+50r)$ and casting width of the even strand is $(850+50r')$, where $u = 1, 2, \ldots, m$, $r = 1, 2, \ldots, q$, and $r' = 1, 2, \ldots, r$. For this pattern, the following data can be calculated in advance.

(i) According to Eq. (1), its counterweights can be predetermined:

$$\begin{cases} G^{\text{odd}}_{(u-1)\cdot q(q+1)/2+r(r-1)/2+r'} = \dfrac{(850+50r)\cdot v_r^{\text{odd}}}{(850+50r)\cdot v_r^{\text{odd}} + (850+50r')\cdot v_{r'}^{\text{even}}} \cdot G^{\max} \\ G^{\text{even}}_{(u-1)\cdot q(q+1)/2+r(r-1)/2+r'} = \dfrac{(850+50r')\cdot v_{r'}^{\text{even}}}{(850+50r)\cdot v_r^{\text{odd}} + (850+50r')\cdot v_{r'}^{\text{even}}} \cdot G^{\max} \end{cases} \tag{2}$$

(ii) Let $E^{\text{odd}}_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'}$ and $E^{\text{even}}_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'}$ indicate whether order i can be consolidated in the odd strand and the even strand of this pattern, respectively. They can be predetermined as follows:

$$E^{\text{odd}}_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'} = \begin{cases} 1, \text{if } W_i^{\min} \le 850+50r \le W_i^{\max} \\ 0, \text{otherwise} \end{cases} \tag{3}$$

$$E^{\text{even}}_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'} = \begin{cases} 1, \text{if } W_i^{\min} \le 850+50r' \le W_i^{\max} \\ 0, \text{otherwise} \end{cases} \tag{4}$$

(iii) Let $C_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'}$ denote the upgrading cost of steel grade if the order i is consolidated in this pattern, which can be predetermined as follows:

$$C_{i,(u-1)\cdot q(q+1)/2+r(r-1)/2+r'} = \begin{cases} c, \text{if } S_u^{\text{all}} \in S_i \\ \infty, \text{if } S_u^{\text{all}} \notin S_i \end{cases} \tag{5}$$

where c is the index of S_u^{all} in S_i when $S_u^{\text{all}} \in S_i$.

Therefore, for each pattern $t \in \{1, \ldots, mq(q+1)/2\}$ and each order $i \in \{1, \ldots, n\}$, G_t^{odd}, G_t^{even}, E_{it}^{odd}, E_{it}^{even} and C_{it} can all be obtained according to Eqs. (2)–(5). G_t^{odd} and G_t^{even} are the counterweights of pattern t in the odd strand and the even strand, respectively. E_{it}^{odd} and E_{it}^{even} are binary flags indicating whether order i can be consolidated in the odd strand and the even strand of pattern t, respectively. C_{it} is the upgrading cost of steel grade when a slab of order i is consolidated in pattern t.

3.2 Scale pair

Scales r and r' are corresponding to casting widths of the odd strand and the even strand, respectively. It is assumed that casting width of the odd strand is always no less than that of the even strand, i.e., $r \geq r'$. And then, (r, r') is called scale pair, where $r = 1, 2, ..., q$, and $r' = 1, 2, ..., r$. It is obvious that the number of scale pairs is $q(q+1)/2$. Suppose that F_{tk} indicates whether scale pair k is used in pattern t. If scale pair k is used in pattern t, then $F_{tk} = 1$; otherwise, $F_{tk} = 0$. Therefore, F_{tk} can be determined in advance for all patterns and all scale pairs, where $t = 1, 2, ..., mq(q+1)/2$, and $k = 1, 2, ..., q(q+1)/2$. It is expected that the least scale pairs should be used in this problem.

3.3 Mathematical model

The model can be described as follows:

$$\min \sum_{t \in O} \alpha_t \tag{6}$$

$$\min \sum_{k \in H} \beta_k \tag{7}$$

$$\min \sum_{t \in O} \chi_t \tag{8}$$

$$\min \sum_{i \in N} \sum_{t \in O} C_{it} \cdot (x_{it}^{\text{odd}} + x_{it}^{\text{even}}) \tag{9}$$

s.t.

$$\sum_{t \in O} (x_{it}^{\text{odd}} + x_{it}^{\text{even}}) = D_i, i \in N \tag{10}$$

$$x_{it}^{\text{odd}} \leq M \cdot E_{it}^{\text{odd}}, i \in N, t \in O \tag{11}$$

$$x_{it}^{\text{even}} \leq M \cdot E_{it}^{\text{even}}, i \in N, t \in O \tag{12}$$

$$y_t \leq \sum_{i \in N} (x_{it}^{\text{odd}} + x_{it}^{\text{even}}) \leq M \cdot y_t, t \in O \tag{13}$$

$$\beta_k \leq \sum_{t \in O} y_t \cdot F_{tk} \leq M \cdot \beta_k, k \in H \tag{14}$$

$$G_t^{\text{odd}} \cdot (\alpha_t^{\text{odd}} - 1) \leq \sum_{i \in N} G_i \cdot x_{it}^{\text{odd}} \leq G_t^{\text{odd}} \cdot \alpha_t^{\text{odd}}, t \in O \tag{15}$$

$$G_t^{\text{even}} \cdot (\alpha_t^{\text{even}} - 1) \leq \sum_{i \in N} G_i \cdot x_{it}^{\text{even}} \leq G_t^{\text{even}} \cdot \alpha_t^{\text{even}}, t \in O \tag{16}$$

$$\alpha_t^{\text{odd}} - \alpha_t^{\text{even}} = \varepsilon_t^{\text{odd}} - \varepsilon_t^{\text{even}}, t \in O \tag{17}$$

$$\varepsilon_t^{\text{odd}} + \varepsilon_t^{\text{even}} \leq 1, t \in O \tag{18}$$

$$\alpha_t \geq \alpha_t^{\text{odd}}, t \in O \tag{19}$$

$$\alpha_t \geq \alpha_t^{\text{even}}, t \in O \tag{20}$$

$$G^{\min} \cdot \alpha_t \leq \chi_t + \sum_{i \in N} G_i \cdot (x_{it}^{\text{odd}} + x_{it}^{\text{even}}) \leq G^{\max} \cdot \alpha_t, t \in O \tag{21}$$

where $O = \{1, ..., mq(q+1)/2\}$, $H = \{1, ..., q(q+1)/2\}$, $N = \{1, ..., n\}$, and M is a very large number, while x_{it}^{odd}, x_{it}^{even}, y_t, α_t^{odd}, α_t^{even}, α_t, $\varepsilon_t^{\text{odd}}$, $\varepsilon_t^{\text{even}}$, β_k, and χ_t are the decision variables. x_{it}^{odd} and x_{it}^{even} are nonnegative integer variables representing the number of dummy slabs when order i is consolidated in the odd strand and the even strand of pattern t, respectively. y_t is a binary variable, which is equal to 1 if pattern t is used, otherwise 0. α_t^{odd} and α_t^{even} are nonnegative integer variables representing the number of charges in the odd strand and the even strand of pattern t, respectively. α_t is a nonnegative integer variable representing the number of charges in pattern t. $\varepsilon_t^{\text{odd}}$ and $\varepsilon_t^{\text{even}}$ are nonnegative integer variables. If $\alpha_t^{\text{odd}} > \alpha_t^{\text{even}}$, then $\varepsilon_t^{\text{odd}} = \alpha_t^{\text{odd}} - \alpha_t^{\text{even}}$ and $\varepsilon_t^{\text{even}} = 0$. If $\alpha_t^{\text{odd}} < \alpha_t^{\text{even}}$, then $\varepsilon_t^{\text{odd}} = 0$ and $\varepsilon_t^{\text{even}} = \alpha_t^{\text{even}} - \alpha_t^{\text{odd}}$. If $\alpha_t^{\text{odd}} = \alpha_t^{\text{even}}$, then $\varepsilon_t^{\text{odd}} = 0$ and $\varepsilon_t^{\text{even}} = 0$. β_k is a binary variable, which is equal to 1 if scale pair k is used, otherwise 0. χ_t is a nonnegative real variable indicating the surplus of pattern t.

Objectives (6)–(9) are to minimize the number of charges, the number of scale pairs, the surplus and the upgrading costs of steel grades, respectively. Constraint (10) is the demand for dummy slabs of each order. Constraints (11) and (12) indicate whether the dummy slab of order i can be consolidated in the odd strand and the even strand of pattern t, respectively. Constraint (13) represents that a pattern is used if any dummy slab is consolidated in its odd strand or even strand. Constraint (14) indicates whether a scale pair is used. Constraints (15) and (16) calculate the number of charges in the odd strand and the number of charges in the even strand for each pattern, respectively, while constraint (17) is the difference between the two numbers, and constraint (18) ensures that the absolute value of the difference is not more than 1. Constraints (19) and (20) guarantee that the number of charges for each pattern is the maximum between the two numbers. Constraint (21) calculates the surplus for each pattern.

4 Solution approach

The estimation of distribution algorithm (EDA) [25] is an evolutionary algorithm based on probabilistic statistics, which is very suitable for solving high-dimensional problems by using a probability model to describe the distribution information of candidate solutions in solution space. This paper uses a heuristic method to produce the set of patterns and then designs a mutation-based EDA to solve the model.

4.1 Heuristic

The number of patterns involved in the model is $mq(q+1)/2$. However, for a specific batch of orders, the number of patterns can be reduced on account of order structure. A heuristic method is proposed below.

(1) For each order i, the casting width range is converted:

$$\begin{cases} r_i^{\min} = \left\lceil \dfrac{(W_i^{\min} - 850)}{50} \right\rceil \\ r_i^{\max} = \left\lfloor \dfrac{(W_i^{\max} - 850)}{50} \right\rfloor \end{cases}, i \in N \tag{22}$$

where the symbol $\lceil x \rceil$ represents the minimum integer that is no less than x; and $\lfloor x \rfloor$ represents the maximum integer that is no more than x.

(2) All the orders are grouped into the subsets according to their steel grades.

(3) For each subset of orders $N_u = \{i | S_u^{\text{all}} \in S_i, i \in N\}$ with identical steel grade S_u^{all}, the set of available scales is obtained: $R_u = \bigcup\limits_{i' \in N_u} \{r | r_{i'}^{\min} \leq r \leq r_{i'}^{\max}\}$. Set $R^{\text{all}} = \bigcup\limits_{k=1}^{m} R_u$.

(4) The real number of patterns is calculated: $o = \sum\limits_{u=1}^{m} |R_u|(|R_u| + 1)/2$. Set $O = \{1, ..., o\}$.

(5) The real number of scale pairs is calculated: $h = \left|R^{\text{all}}\right|\left(\left|R^{\text{all}}\right| + 1\right)/2$. Set $H = \{1, ..., h\}$.

4.2 Mutation-based EDA

4.2.1 Encoding

The chromosome is designed as $\boldsymbol{x} = (x_{ij}), i \in N, j \in D_i$, where $x_{ij} \in \{1, ..., 2o\}$ contains the following gene information:

(1) If $1 \leq x_{ij} \leq o$, then slab j of order i is consolidated in the odd strand of pattern x_{ij};

(2) If $o < x_{ij} \leq 2o$, then slab j of order i is consolidated in the even strand of pattern $(x_{ij} - o)$.

Therefore, x_{ij} represents the pattern and strand information when slab j of order i is consolidated.

4.2.2 Probability vector

The probability vector is the core of EDA. Through the probability vector and its updating, the spatial distribution of solution and the evolution trend of the population are described. According to the chromosome, a probability vector is established as below: $\boldsymbol{P}(\boldsymbol{x}) = (P_v(x_{ij}))$, where $P_v(x_{ij})$ represents the probability of $x_{ij} = v$, $v \in \{1, ..., 2o\}$.

Set $O' = \{1, ..., 2o\}$. For order i and pattern t, if $E_{it}^{\text{odd}} = 0$ or $C_{it} = \infty$, it means $x_{ij} \in O' \backslash \{t\}$. Similarly, if $E_{it}^{\text{even}} = 0$ or $C_{it} = \infty$, it means $x_{ij} \in O' \backslash \{t + o\}$. Thus, the domain of x_{ij} is $B_i = O' \backslash \{t | E_{it}^{\text{odd}} = 0 \text{ or } C_{it} = \infty, t \in O\} \backslash \{t + o | E_{it}^{\text{even}} = 0 \text{ or } C_{it} = \infty,\ t \in O\}$. It is assumed that b_i is the number of elements in B_i. The probability vector in generation L is denoted by $(P_v^L(x_{ij}))$. In generation 0, $P_v^0(x_{ij})$ is initialized as follows:

$$P_v^0(x_{ij}) = \begin{cases} 1/b_i, \text{if } v \in B_i \\ 0, \text{if } v \notin B_i \end{cases}, i \in N, j \in D_i, v \in O' \tag{23}$$

In generation L, new individuals are generated by randomly sampling the probability vector based on roulette wheel selection method. The sampling process is listed below:

(1) For slab j of order i, $P_1^L(x_{ij})$, $P_2^L(x_{ij})$, …, $P_v^L(x_{ij})$, …, $P_{2o}^L(x_{ij})$ make up of the proportions in roulette wheel.

(2) According to the roulette wheel selection, a random number between 0 and 1 is generated to determine the value of x_{ij}.

(3) For all slabs and all orders, repeat steps (1) to (2).

After new individuals are generated to form the new population, the probability vector in generation $L + 1$ is constructed and updated by the following strategy:

$$\begin{cases} P_v^{L+1}(x_{ij}) = (1 - \lambda) \cdot P_v^L(x_{ij}) + \lambda \cdot \phi(v, x_{ij}) \\ P_v^{L+1}(x_{ij}) = P_v^{L+1}(x_{ij}) / \sum\limits_{v' \in O'} P_{v'}^{L+1}(x_{ij}) \end{cases}, i \in N, j \in D_i, v \in O' \tag{24}$$

where $\lambda \in (0, 1)$ is the learning rate, and $\phi(v, x_{ij})$ represents the frequency of $x_{ij} = v$ in the new population.

4.2.3 Mutation

Mutation in evolutionary algorithms is to introduce new genetic material and maintain a certain level of diversity in a population. After a new individual is generated by sampling process, the following mutation strategy is used.

(1) Generate a random number between 1 and n to pick a order, such as order i'.

(2) Generate a random number between 1 and $D_{i'}$ to pick a slab of order i', such as slab j'.

(3) Randomly select an element from $B_{i'}$ as the value of $x_{i'j'}$.

4.2.4 Fitness evaluation

For an individual, the value of its fitness is evaluated as below:

$$f = -\left[f_1 \cdot \sum_{t \in O} \alpha_t + f_2 \cdot \sum_{k \in H} \beta_k + f_3 \cdot \sum_{t \in O} \chi_k + f_4 \sum_{i \in N} \sum_{j \in D_i} C_{i, \psi(x_{ij})} \right] \tag{25}$$

where

$$\psi(x_{ij}) = \begin{cases} x_{ij}, & \text{if } 1 \le x_{ij} \le o \\ x_{ij} - o, & \text{if } o < x_{ij} \le 2o \end{cases}.$$

f_1, f_2, f_3 and f_4 are the weights of four objectives; β_k is given by constraint (10); and α_t and χ_t need to be calculated by Eqs. (26)–(29).

$$\alpha_t^{\text{odd}} = \left\lceil \frac{\sum_{i \in N} \sum_{j \in D_i} G_i \cdot \theta^{\text{odd}}(t, x_{ij})}{G_t^{\text{odd}}} \right\rceil, t \in O \tag{26}$$

$$\alpha_t^{\text{even}} = \left\lceil \frac{\sum_{i \in N} \sum_{j \in D_i} G_i \cdot \theta^{\text{even}}(t, x_{ij})}{G_t^{\text{even}}} \right\rceil, t \in O \tag{27}$$

$$\alpha_t = \sigma(\alpha_t^{\text{odd}}, \alpha_t^{\text{even}}) \cdot \max\{\alpha_t^{\text{odd}}, \alpha_t^{\text{even}}\}, t \in O \tag{28}$$

$$\chi_t = \max\{0, (\alpha_t \cdot G^{\min} - \sum_{i \in N} \sum_{j \in D_i} G_i \cdot \theta(t, x_{ij}))\}, t \in O \tag{29}$$

where

$$\theta^{\text{odd}}(t, x_{ij}) = \begin{cases} 1, \text{if } x_{ij} = t \\ 0, \text{otherwise} \end{cases},$$

$$\theta^{\text{even}}(t, x_{ij}) = \begin{cases} 1, \text{if } x_{ij} = t + o \\ 0, \text{otherwise} \end{cases},$$

$$\theta(t, x_{ij}) = \begin{cases} 1, \text{if } \psi(x_{ij}) = t \\ 0, \text{otherwise} \end{cases},$$

$$\sigma(\alpha_t^{\text{odd}}, \alpha_t^{\text{even}}) = \begin{cases} 1, \text{if } \left|\alpha_t^{\text{odd}} - \alpha_t^{\text{even}}\right| \le 1, \\ M, \text{otherwise} \end{cases}.$$

Note that the function $\sigma(\alpha_t^{\text{odd}}, \alpha_t^{\text{even}})$ is the penalty to violate constraint (18).

4.3 Algorithm procedure

Algorithm 1 lists the procedure of hybrid optimization algorithm combined with heuristic and mutation-based EDA. The stopping criterion is specified in terms of a maximum number of generations (i.e., $L^{\max}$). X^{local} is the local optimal individual, while X^{best} is the global optimal individual. a is the population size.

```
Algorithm 1
{
    Determine the set of patterns and the set of scale pairs by using the heuristic;
    Calculate the parameters $G_t^{odd}$, $G_t^{even}$, $C_{it}$, $E_{it}^{odd}$, $E_{it}^{even}$, $F_{tk}$ for each pattern;
    Initialize the probability vector $(P_v^0(x_{ij}))$ and $a$;
    $L = 0$;
    while ( $L \le L^{max}$ ){
        Randomly sample the probability vector $(P_v^L(x_{ij}))$ to generate $a$ individuals;
        Mutate $a$ individuals according to the mutation strategy;
        Evaluate the fitness for $2a$ individuals;
        Sort $2a$ individuals by their fitness in descending order;
        Select the front $a$ individuals as the new population;
        $X^{local}$ = the first individual of new population;
        $X^{best}$ = max_fitness ($X^{best}$, $X^{local}$);
        Construct $P_v^{L+1}(x_{ij})$ by the new population;
        $L++$;
    }
    Output the global optimal individual $X^{best}$;
}
```

5 Experiment and analysis

Experiments are conducted on four groups of test data collected from practical production orders of Baosteel, which are shown in Table 1. The proposed algorithm is coded in Microsoft Visual C++, and the computational experiments are performed on a PC with Intel Pentium-4 2.8 GHz CPU and 2 GB RAM.

5.1 Parameter setting

The technological parameters are as follows: $G^{\min} = 290$, $G^{\max} = 310$, and $q = 24$. With considering the importance of four objectives, the weighted parameters in the model are set as below: $f_1 = 100, f_2 = 10, f_3 = 0.1,$ and $f_4 = 1$. As for the algorithm parameters, $L^{\max}$ is set to 1000, while λ and a are undetermined. In order to analyze the influence of algorithm parameters, the following trials on Test1 are conducted:

(1) Learning rate is fixed as 0.1 and population size is set to 20, 50, 80 and 110, respectively.
(2) Population size is fixed as 80 and the learning rate is set to 0.004, 0.02, 0.1 and 0.5, respectively.

Test1 is computed 10 times with each type of parameters. Box plots based on these trials are shown in Fig. 2. Figure 2a is used to analyze the influence of population size on Test1 when learning rate is fixed, while Fig. 2b is used to analyze the influence of learning rate on Test1 when population size is fixed. Each rectangular box is a statistic for 10 computations on Test1 with determined λ and a, displaying the dispersion of results.

As seen from Fig. 2, increasing the population size can improve the algorithm performance, because larger population is conducive to a more comprehensive search for solution space. However, large population will cause an increase in computational complexity. When the population reaches a certain size, the marginal improvement will be trivial. Moreover, the learning rate should not be too high or too low. If the learning rate is too high, the population will converge to local optimal solution, which turns out to be premature. If the learning rate is too low, the convergence speed tends to be slow and the ability of searching for the optimum gets worse. Therefore, population size and learning rate should be set appropriately considering the performance and efficiency. In the next computations on four groups of test data, the two parameters are set as follows: $\lambda = 0.1$, and $a = 80$.

5.2 Computational results

Each test datum is computed 10 times with the proposed algorithm, and the best result is kept. To demonstrate the proposed algorithm, the manual method is also used to produce the result. The manual method for charge planning is a greedy procedure. Firstly, all slabs are separated into subsets according to their first steel grades (it is not easy to think about the upgrading costs of steel grades in the manual method because the combinations of steel grades are very complex), and the statistics of their mass are displayed in a matrix in which rows represent steel grades and columns represent widths. Secondly, the planner analyzes the order structure by observing the matrix, selects a subset of slabs with the same steel grade into a pool and determines two sets of casting widths covering them,

Table 1 Four groups of order data

Case	i	D_i	G_i/t	$(W_i^{\min} - W_i^{\max})$/mm	S_i
Test1	1	6	12.47	1132−1269	{JL6103E5, GL6102E1, …}
	…	…	…	…	…
	20	15	24.16	1723−1970	{JT6103D1, IL6205D2, …}
Test2	1	8	17.09	900−1095	{IV7105G1, JX1140E5, …}
	…	…	…	…	…
	50	9	30.52	1450−1730	{IV8106G2, JU1150H1, …}
Test3	1	25	19.87	951−1098	{IV4355E5, JU1150H1, …}
	…	…	…	…	…
	100	3	27.48	1495−1658	{IV4355E5, JU1250H1, …}
Test4	1	10	23.29	1214−1380	{EU6814A6, EU6810A6, …}
	…	…	…	…	…
	200	8	16.24	1819−2030	{GV4924E1, IV5937E1, …}

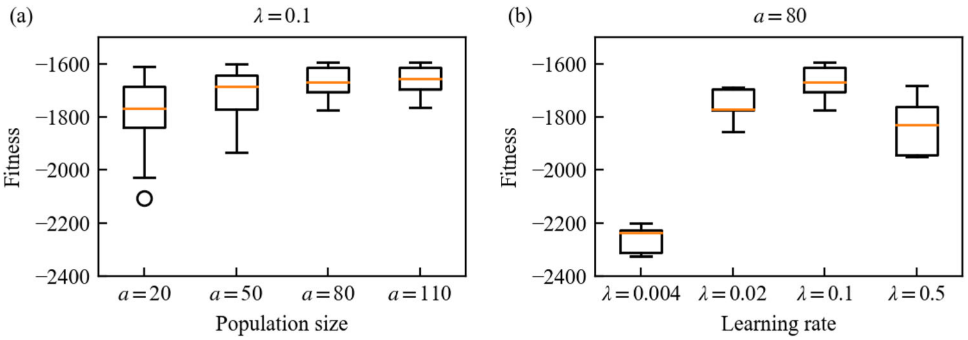

Fig. 2 Influence of algorithm parameters on Test1

corresponding to two strands. Finally, these selected slabs in the pool are consolidated into one or several charges (i.e., cast-lot) with determined casting widths of the odd strand and the even strand. Repeat these steps until all slabs are consolidated.

The computational results are reported in Table 2.

In Table 2, the column labelled Condition represents the limit condition for casting width of the odd strand and the even strand, which includes Sam. condition and Dif. condition. Sam. means the situation that does not make a distinction between the odd strand and the even strand when charge planning, i.e., for each charge, casting width of the odd strand is the same as that of the even strand. Dif. means another situation that makes a distinction between the odd strand and the even strand when charge planning, i.e., for each charge, casting width of the odd strand may be different from that of the even strand. Man_sam and Man_dif represent the manual method under the Sam. and Dif. conditions, respectively. Alg_sam and Alg_dif represent the proposed algorithm under the Sam. and Dif. conditions, respectively. Note that under the Sam. condition, the number of patterns in the model is mq theoretically, not $mq(q+1)/2$. In other words, the number of patterns under the Sam. condition is less than the one under the Dif. condition.

The following conclusions can be drawn from the results in Table 2:

(1) The objective values indicate that the proposed algorithm is obviously dominant compared to the manual method under an identical condition, whether it is the Sam. condition or Dif. condition. The number of charges and surplus obtained by the proposed algorithm are remarkably less than the ones obtained by the manual method. The number of scale pairs is also less than the one obtained by the manual method, which is helpful to reduce the width adjustment times when cast planning. Moreover, the proposed algorithm has better control of steel grades' upgrading costs.

(2) Regardless of the proposed algorithm or the manual method, the values obtained under the Dif. condition are superior to the ones obtained under the Sam. condition. The reason lies in that those orders without intersection in width range can be consolidated together into a charge by distributing in the odd strand and the even strand, which will inevitably reduce the number of charges and the number of scale pairs.

(3) The proposed algorithm can achieve the desired solution for different test data. Especially when the problem size is bigger, the effect will be more obvious. Of course, the computing time will increase accordingly.

Figure 3a–d displays the curves with the fastest convergence speed and the best global optimal solution when four groups of test data are computed by the proposed algorithm. As shown in Fig. 3, the proposed algorithm has a good convergence performance. Meanwhile, the increase in problem size slows down the convergence because its solution set expands.

6 Conclusions

1. Charge planning is one of batching problems for steelmaking and continuous casting production, and its optimization will be conducive to subsequent cast planning. This paper studies the charge planning problem in the twin strands continuous casting production.
2. Considering the different widths in the twin strands, the resulting counterweights and the constraints of steelmaking, a multi-objective optimization model is established to minimize the number of charges, the

Table 2 Results of proposed algorithm and manual method

Case	Condition	Method	Number of charges	Number of scale pairs	Surplus/t	Upgrading cost	Computing time/s
Test1	Sam.	Man_sam	16	7	512.92	29	≥120
		Alg_sam	15	6	233.83	22	3
	Dif.	Man_dif	15	6	233.83	22	≥120
		Alg_dif	15	5	233.83	22	5
Test2	Sam.	Man_sam	40	12	1108.24	59	≥300
		Alg_sam	38	11	495.50	46	19
	Dif.	Man_dif	39	11	801.47	53	≥300
		Alg_dif	38	10	495.50	46	24
Test3	Sam.	Man_sam	80	17	1326.52	87	≥600
		Alg_sam	78	16	761.91	73	46
	Dif.	Man_dif	79	15	1059.26	74	≥600
		Alg_dif	77	14	462.75	66	59
Test4	Sam.	Man_sam	166	19	3198.24	99	≥1200
		Alg_sam	162	18	2003.59	92	87
	Dif.	Man_dif	161	16	1701.92	91	≥1200
		Alg_dif	158	14	809.83	81	106

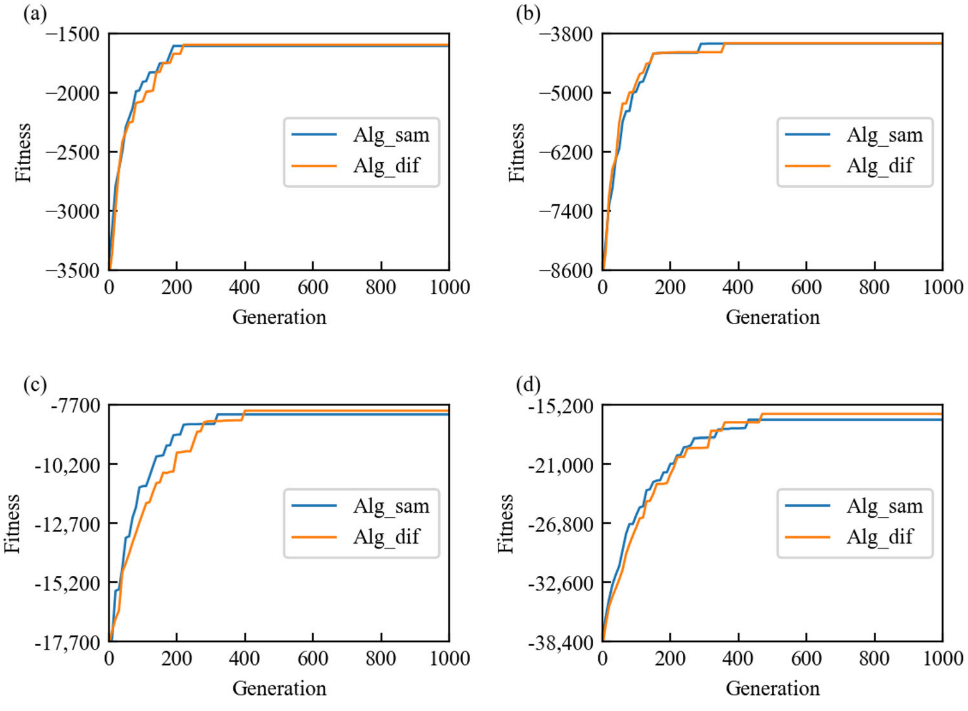

Fig. 3 Algorithm convergence curves. **a** Test1; **b** Test2; **c** Test3; **d** Test4

number of scale pairs, the surplus and the upgrading costs of steel grades.

3. A hybrid optimization algorithm combined with heuristic and mutation-based EDA is proposed to solve the model. Experiments are conducted on several groups of test data under the different conditions. The results indicate that the proposed algorithm can generate better solutions than the manual method.

Acknowledgements This work was supported by the National Key Research and Development Program of China (No. 2017YFB0304100).

References

[1] H. Missbauer, W. Hauber, W. Stadler, Int. J. Prod. Res. 47 (2009) 4147–4172.

[2] J.Y. Long, Z. Zheng, X.Q. Gao, Y.M. Gong, J. Iron Steel Res. Int. 21 (2014) No. S1, 44–50.

[3] L.X. Tang, J.Y. Liu, A.Y. Rong, Z.H. Yang, Eur. J. Oper. Res. 133 (2001) 1–20.

[4] L.X. Tang, G.S. Wang, Omega 36 (2008) 976–991.

[5] I. Harjunkoski, I.E. Grossmann, Comput. Chem. Eng. 25 (2001) 1647–1660.

[6] A. Atighehchian, M. Bijari, H. Tarkesh, Comput. Oper. Res. 36 (2009) 2450–2461.

[7] S.P. Yu, Q.K. Pan, J. Iron Steel Res. Int. 19 (2012) No. 12, 33–41.

[8] L.X. Tang, Y. Zhao, J.Y. Liu, IEEE Trans. Evol. Comput. 18 (2014) 209–225.

[9] J.H. Hao, M. Liu, S.L. Jiang, C. Wu, Eur. J. Oper. Res. 244 (2015) 966–979.

[10] Q.K. Pan, Eur. J. Oper. Res. 250 (2016) 702–714.

[11] Z. Zheng, J.Y. Long, X.Q. Gao, J. Iron Steel Res. Int. 24 (2017) 586–594.

[12] J. Mori, V. Mahalec, Comput. Chem. Eng. 101 (2017) 312–325.

[13] S.L. Jiang, Z. Zheng, M. Liu, Comput. Ind. Eng. 115 (2018) 582–594.

[14] S. Kammammettu, Z.K. Li, IFAC-PapersOnLine 51 (2018) 262–267.

[15] A. Bellabdaoui, J. Teghem, Int. J. Prod. Econ. 104 (2006) 260–270.

[16] L.X. Tang, G.S. Wang, Z.L. Chen, Oper. Res. 62 (2014) 772–787.

[17] J. Yi, S.J. Jia, B. Du, Q. Liu, J. Iron Steel Res. Int. 26 (2019) 242–250.

[18] H.Y. Dong, M. Huang, W.H. Ip, X.W. Wang, Int. J. Prod. Res. 50 (2012) 5747–5761.

[19] S.H. Song, Int. J. Prod. Res. 52 (2014) 1443–1454.

[20] S.X. Liu, J.F. Tang, J.H. Song, Int. J. Prod. Econ. 100 (2006) 30–43.

[21] J.Q. Li, Q.K. Pan, K. Mao, P.N. Suganthan, Knowl.-Based Syst. 72 (2014) 28–36.

[22] J.H. Lin, M. Liu, J.H. Hao, S.L. Jiang, Comput. Oper. Res. 72 (2016) 189–203.

[23] D.F. Zhu, Z. Zheng, X.Q. Gao, J. Iron Steel Res. Int. 17 (2010) No. 9, 19–24.

[24] L.L. Sun, F.J. Luan, T.M. Ma, IFAC-PapersOnLine 48 (2015) 144–149.

[25] P. Larrañaga, J.A. Lozano, Estimation of distribution algorithms: a new tool for evolutionary computation, Kluwer Press, Boston, USA, 2002.