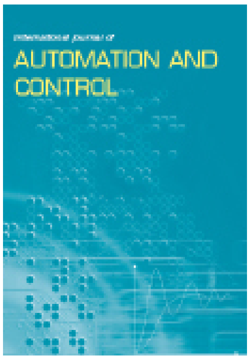


International Journal of Automation and Control

ISSN online: 1740-7524 - ISSN print: 1740-7516
https://www.inderscience.com/ijaac

Research on steelmaking-continuous casting cast batch planning based on an improved surrogate absolute-value Lagrangian relaxation framework

Congxin Li, Liangliang Sun

DOI: 10.1504/IJAAC.2025.10062754

Article History:

Received:	12 September 2023
Last revised:	04 December 2023
Accepted:	15 February 2024
Published online:	02 December 2024

Research on steelmaking-continuous casting cast batch planning based on an improved surrogate absolute-value Lagrangian relaxation framework

Congxin Li

School of Mechanical Engineering,
Shenyang Jianzhu University,
Shenyang, 110168, China
Email: licongxin2037@163.com

Liangliang Sun*

School of Mechanical Engineering,
Shenyang Jianzhu University,
Shenyang, 110168, China

and

School of Control Engineering,
Northeastern University at Qinghuangdao,
Qinghuangdao, 066004, China
Email: swinburnsun@163.com
*Corresponding author

Abstract: Cast batch planning (CBP) is the bottleneck of batch planning in the steelmaking-continuous casting-hot rolling (SM-CC-HR) section. With the rapid development of the market-oriented demand of steel enterprises to multiple species, small batches, and on-time delivery, the batch planning integrated production process has dramatically increased the flexibility of the CBP as well as the functional requirements of the time dynamic balance. Therefore, it is of great significance to research the method of CBP to improve production efficiency and reduce material and energy consumption. In this paper, based on the improved surrogate absolute-value Lagrangian relaxation (ISAVLR) framework, the heuristic method based on a multiplier iteration strategy with controllable gradient direction combined with a local search (LS) algorithm is proposed. The 'zigzagging' problem in the traditional Lagrangian relaxation (LR) is overcome and the solution efficiency is improved while the original problem is provided with tighter lower bounds. Finally, simulation experiments based on real production data verify the effectiveness of the proposed method.

Keywords: steelmaking-continuous casting; ISAVLR; improved surrogate absolute-value Lagrangian relaxation; CBP; cast batch planning; heuristic.

Reference to this paper should be made as follows: Li, C. and Sun, L. (2025) 'Research on steelmaking-continuous casting cast batch planning based on an improved surrogate absolute-value Lagrangian relaxation framework', *Int. J. Automation and Control*, Vol. 19, No. 1, pp.37–58.

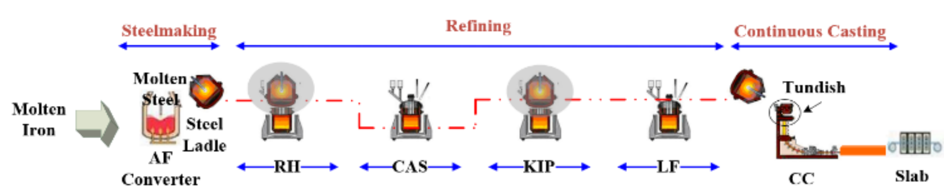
Biographical notes: Congxin Li received his BS in Measurement and Control Technology and Instrument from North University of China in 2017. The MS in Control Science and Engineering from Shenyang Jianzhu University in 2020. He is currently pursuing his PhD in Mechanical Engineering from Shenyang Jianzhu University. His research interests include industrial scheduling and batch planning.

Liangliang Sun received his PhD in Control Theory and Control Engineering from Northeastern University, China, in 2015. He is currently a Professor at Northeastern University at Qinghuangdao, China. His current research interests include intelligent manufacturing systems, planning, scheduling, and coordination of design, manufacturing, and service activities intelligence field.

1 Introduction

A typical integrated manufacturing system for steel production consists of three main successive stages: ironmaking, steelmaking-continuous casting, and rolling. In the ironmaking stage, blast charge ironmaking is utilised to reduce iron from iron-containing raw materials and convert it to molten iron in the blast charge. In the steelmaking-continuous casting stage, the iron in the blast charge is converted into slabs matching the specifications required by customer orders through three closely coordinated production processes: steelmaking, refining, and continuous casting, The schematic diagram of steelmaking-continuous casting is shown in Figure 1. In the rolling stage, it is divided into hot rolling and cold rolling. Slabs can be made into hot-rolled strip coils through hot rolling, and if they meet the customer's order requirements they can be directly delivered to the customer, if not, they will be delivered to the cold rolling stage for further processing. Steelmaking-continuous casting plays a role in the production process of steel manufacturing, which is an important stage in steel production and is also a bottleneck stage (Sun et al., 2021). Steelmaking-continuous casting is divided into batch planning and scheduling decisions (Tang et al., 2011). The cast batch planning (CBP) belongs to the steelmaking-continuous casting batch planning. The problem of CBP is to determine the optimal combination of charge production programs based on the charge steel grade, size, and delivery date approximation, taking into account the utilisation rate of resource allocation, production capacity, as well as process constraints. Therefore, under the market demand of multiple species, small batches, and on-time delivery, the preparation of a high-quality CBP can help steel enterprises ensure a quick response to customer demand at the same time, reduce enterprise production costs, and improve production efficiency.

Figure 1 Diagram of steelmaking-continuous casting (see online version for colours)



The difficulty of the CBP problem is reflected in both modelling and optimisation. From the perspective of mathematical modelling, the steel production process is accompanied by high-temperature and high-pressure chemical and physical changes, accompanied by a huge energy conversion (Xu et al., 2021), and the optimisation of the CBP needs to comprehensively consider the customer's needs, the rules for the implementation of the production process and the limitations of the capacity of the machine, to satisfy the dynamic balance of the logistics and time among the iron and steel production processes, and then make efficient use of energy and resources, however, the actual modelling process is difficult to comprehensively combine multiple performance indicators of the production process.

From the optimisation aspect of the analysis, with the increase in the number of orders, the number of charges also increases accordingly. In the process of reorganisation of customer diversity order data from multiple varieties and small batches into large-scale production batches of the enterprise, the difficulty of solving the model grows exponentially with the increase in the number of production equipment and the increase in the number of orders, resulting in difficulty in solving the problem with high quality within the required time constraints.

Recently, problems related to CBP problem have been widely explored by researchers using different approaches. Problem optimisation methods are mainly divided into three categories: Heuristic (Chang et al., 2000; Tang and Wang, 2008; Yi et al., 2012; Yang et al., 2014), intelligent algorithm (Yang et al., 2015; Tang and Luo, 2007; Xu and Wang, 2015; Xu et al., 2016; Wang et al., 2022) and optimisation method of operations research (Zhu et al., 2021) as shown in Table 1. However, the works of literature (Chang et al., 2000; Tang and Wang, 2008; Yi et al., 2012; Yang et al., 2014) only consider a part of indicators in actual production as optimisation objectives, which makes the problem optimisation incomplete and difficult to meet the actual production demand. Although the intelligent optimisation method (Yang et al., 2015; Tang and Luo, 2007; Xu and Wang, 2015; Xu et al., 2016; Wang et al., 2022) has been able to take into account the mathematical model structure and the exponential growth of data in the optimisation problem of CBP, however, given various physical attributes of orders and the complexity of steelmaking-continuous casting production flow, it is difficult to make the knowledge reserve of the decision-making system complete or consistent, thus unable to meet the actual production demand (Liu et al., 2016). A mixed integer programming (MIP) model was established by comprehensively considering various optimisation objectives in the production process and was solved based on the augmented Lagrangian relaxation (ALR) algorithm (Zhu et al., 2021). The standard Lagrangian relaxation (LR) convergence rate can be improved by the quadratic term penalty violation constraint in the ALR framework. However, the introduction of the quadratic term under the ALR framework transforms the problem into a nonlinear one, which in turn leads to model indivisibility. A study (Bragin et al., 2018) shows that optimisation methods in the surrogate absolute-value Lagrangian relaxation (SAVLR) framework can linearise the model exactly with few additional constraints. However, the error of the SAVLR function comparison to the quadratic function increases large with the level of constraint violation and does not capture the quadratic growth characteristics well. Based on the previous research work, the conclusion can be obtained, the steelmaking-continuous casting CBP problem due to its optimisation process of large scale, multiple objectives, multiple constraints, multiple coupling, and multiple stages characteristics, the existing CBP problem-solving method is difficult to overcome the problem of the complex process constraints, data size is huge, the computational complexity of the problem is high. Therefore, it is the main research content of this paper to propose optimisation model algorithms that are more suitable for the complexity of the problem and to improve the efficiency and quality of the solution to be more suitable for the actual production requirements. Table 1 shows the classification of CBP problem descriptions. Where abbreviations are defined as, integer programming (IP), multiple travelling salesman problem (MTSP), travelling salesman problem (TSP), generalised vehicle routing problem (GVRP), quadratic integer programming (QIP), MIP.

Table 1 Classification of CBP

Papers	*Model*	*Approach*
Chang et al. (2000)	IP	Heuristic
Tang and Wang (2008)	IP	Two-stage heuristic
Yi et al. (2012)	MTSP	Heuristic + K-opt Neighborhood Search + Estimation of Distribution
Yang et al. (2014)	TSP	Heuristic + Cross Entropy
Yang et al. (2015)	GVRP	Improved Dross Entropy + Reaching algorithm
Tang and Luo (2007)	QIP	Iterated Local Search
Xu and Wang (2015)	MIP	Improved Differential Evolution
Xu et al. (2016)	MIP	Subpopulation-based Differential Evolution
Wang et al. (2022)	MIP	Improved Non-dominated Sorting Genetic Algorithms + Local Search
Zhu et al. (2021)	QIP	Augmented Lagrangian Relaxation Algorithm

2 Problem description

CBP is the bottleneck of batch planning in the SM-CC-HR section. In the continuous casting (CC) process, firstly, take over the steelmaking stage of the charge loaded with steel, and then, in the continuous casting machine on the steel casting, formed with a certain specification and quality of the slab, and finally, transported to the downstream production stage for rolling. Therefore, CBP is a key stage that connects the entire steel production process. The requirements of an intermediate contract (e.g., hot slab) or a final contract (e.g., hot strip) determine the attributes of the steel grade, specification, and delivery date of the molten steel in the charges, which in turn determines the attributes of the charges. CBP is the process of determining the optimal combination scheme for charges, using charges with given attributes as input conditions, taking into account continuous CC constraints and tundish capacity constraints, depending on the degree of approximation of the attributes.

The objective of optimisation of CBP is considered in the following aspects. Firstly, the attributes of each charge are not consistent, however, the CC production process places certain restrictions on the attributes of the charges that make up the same cast, the violation of which will result in the inability to group casts or high additional costs. Secondly, the steel of all charges in the same cast will be injected into the tundish (the vessel that holds the steel on the continuous casting machine) when it is grouped, and the tundish has a certain service life, and whether or not it reaches its service life it needs to carry out regular maintenance on its high-temperature-resistant layer, which will incur a high cost (Ma et al., 2015), so to save costs, the optimisation process needs to consider not exceeding the tundish's service life and increase the utilisation rate of the tundish as much as possible. Finally, to increase customer satisfaction, all charges in the pre-selection pool are scheduled for production as much as possible. In summary, the CBP is optimised using the following performance indicators as objective functions: minimising the penalty for differences in charge attributes (steel grade, width, and delivery date) within the same cast, minimising the difference between the tundish lifespan and the number of charges produced by the tundish, and minimising the penalty due to charges not being selected, respectively. In addition, the optimisation process needs to satisfy the following production process constraints:

1 Steel grades of adjacent charges are required to be the same or similar in each CBP. Interlocking billets are produced in the continuous casting of charges of different steel grades. Interlocking billets are awarded to charges with low steel grades, and the additional cost of substituting good for bad is caused.

2 The charges in the same cast should have similar delivery times. The delivery time is one of the attributes of the charge, which is determined by the delivery time of the slab in the charge, and is the identification of urgent orders. Excessive differences in the delivery times of the charges in the same cast will lead to early or late deliveries of the charges in the same cast, which reduces customer satisfaction.

3 The charges in the same CBP shall have the same thickness. When processing charges with different thicknesses, the continuous casting machine must be stopped for several hours to adjust the equipment, and restarting the machine will incur additional costs (Long et al., 2018). To avoid the additional cost of adjustments, steel mills generally produce the same thickness of charges on consecutive days. In this paper, it is assumed that all charges to be group-cast have the same thickness attribute.

4 The charges should be arranged in a non-increasing order of width in the continuous-casting process. In addition, the frequency and range of jumps between charges are limited by production regulations. During the continuous casting process, the width of the slabs in the charges is adjusted only from wide to narrow due to production technology requirements. The width of adjacent slabs can only be adjusted once, to one of two discrete values of 50 mm or 100 mm.

5 The number of charges in a tundish must not exceed the lifespan of the tundish. The lifespan of a tundish is the maximum number of charges it can hold, and the length of the lifespan is affected by the grade composition of the molten steel it holds. When multiple charges are included in the CBP, multiple tundishes are used to meet the production demand. However, due to the high cost of replacing tundishes, it is necessary to maximise the capacity utilisation of the lifespan to reduce production costs.

6 Actual production quantities meet machine capacity constraints. In the production target, the following requirements for the next production cycle are given: the range of the number of charges, the range of the number of refining charges, the range of the weight of the hot roll materials in the hot rolling process, and the range of the total weight of slabs to be processed by the downstream units. The above ranges should be met by the charge properties in the charge batch planning.

The mathematical model of the CBP optimisation problem is as follows, and all symbol definitions for this model are presented as follow.

N	Number of charges
i, j	Job number of charges, $i = \{1,2,\ldots N\}$. $j = \{1,2,\ldots,N\}$.
G_i, G_j	Steel grades of charges i and j
W_i, W_j	Cast widths of charges i and j
D_i, D_j	Rolling due dates of charges i and j
TL	Tundish lifespan
p_{ij}^{G}	Penalty caused by steel grade jumps under steel grade compatibility conditions of charges i and j ($\lvert G_i - G_j \rvert \le 3$). When the steel grade is not compatible ($\lvert G_i - G_j \rvert > 3$), $p_{ij}^{G} = \infty$
p_{ij}^{W}	Penalty caused by casting width jumps of charges i and j
p_{ij}^{D}	Penalty caused by jumps in rolling due dates of charges i and j
p_j^{TL}	Penalty caused by the difference between the tundish lifespan of the jth tundish and the total number of charges in the cast
p_i^{US}	Penalty caused by the ith charge is not selected for the cast batch planning
φ_k	Weighting coefficients. $\sum_k \varphi_k = 1, k = \{1,2,\ldots,5\}$
$[L_{chr}, H_{chr}]$	Lower and upper limits of the number of charges required in the production protocols
$[L_{RH}, H_{RH}]$	Lower and upper limits on the number of refining charges required in the production protocols
$[L_{pre}, H_{pre}]$	Lower and upper limits for the weight of the hot roll materials required by the production protocols
$[L_f, H_f]$	Lower and upper limits for the total mass of downstream slabs required in production protocols
Q_i^{rh}	Refining mark of charge i
Q_i^{pre}	The weight of hot roll materials in charge i
Q_i^{f}	The slab weight required by the downstream production process f in charge i
F	Total number of downstream processes
f	Downstream process index, $f = \{1,2,\ldots,F\}$.
Decision variables	
x_{ij}	A binary variable, which is equal to 1 if charge i is assigned to produce with j, otherwise 0
x_{jj}	A binary variable, which is equal to 1 if the charge j selected as a charge centre of the cast batch planning, otherwise 0

1 Minimising penalties caused by steel grade jumps in adjacent charges within the same cast.

$$Z^G = min\sum_{i=1}^{N}\sum_{j=1}^{N}\left|G_i - G_j\right| \cdot p_{ij}^G \cdot x_{ij} \tag{1}$$

2 Minimising penalties caused by casting width jumps in adjacent charges.

$$Z^W = min\sum_{i=1}^{N}\sum_{j=1}^{N}\left|W_i - W_j\right| \cdot p_{ij}^W \cdot x_{ij} \tag{2}$$

3 Minimising penalties caused by jumps in rolling due dates in adjacent charges.

$$Z^D = min\sum_{i=1}^{N}\sum_{j=1}^{N}\left|D_i - D_j\right| \cdot p_{ij}^D \cdot x_{ij} \tag{3}$$

4 Minimise the difference between the tundish lifespan and the number of charges (the charges included in the cast).

$$Z^{TL} = min\sum_{j=1}^{N} p_j^{TL} \cdot \left(TL \cdot x_{jj} - \sum_{i=1}^{N} x_{ij}\right) \tag{4}$$

5 Minimising penalties caused by unchecked charges.

$$Z^{US} = min\sum_{i=1}^{N} p_i^{US} \cdot \left(1 - \sum_{j=1}^{N} x_{ij}\right) \tag{5}$$

The objective function is weighted and constraints are added to get the following form:

$$\min Z$$

$$\text{with } Z = \varphi_1 \cdot Z^G + \varphi_2 \cdot Z^W + \varphi_3 \cdot Z^D + \varphi_4 \cdot Z^{TL} + \varphi_5 \cdot Z^{US} \tag{6}$$

Subject to (s.t.):

$$\sum_{j=1}^{N} x_{ij} \le 1,\ i = \{1, 2, \ldots, N\}. \tag{7}$$

$$2 \le \sum_{i=1}^{N} x_{ij} \le TL,\ j = \{1, 2, \ldots, N\}. \tag{8}$$

$$\sum_{j=1}^{N} x_{jj} = M \tag{9}$$

$$L_{chr} \le \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij} \le H_{chr} \tag{10}$$

$$L_{RH} \le \sum_{i=1}^{N}\sum_{j=1}^{N} Q_i^{rh} \cdot x_{ij} \le H_{RH} \tag{11}$$

$$L_{pre} \le \sum_{i=1}^{N}\sum_{j=1}^{N} Q_i^{pre} \cdot x_{ij} \le H_{pre} \tag{12}$$

$$L_f \le \sum_{i=1}^{N}\sum_{j=1}^{N} Q_i^{f} \cdot x_{ij} \le H_f, f \in F \tag{13}$$

$$x_{ij} \in \{0,1\}, x_{jj} \in \{0,1\},\ i = \{1,2,\ldots,N\}. j = \{1,2,\ldots,N\}. \tag{14}$$

Constraint (7) guarantees that each charge is assigned to at most one cast plan. Constraint (8) tundish capacity constraint which means that the number of charges processing on tundish cannot exceed the tundish lifespan. Constraints (9) ensure that the total number of casts should be equal to the number of available casts in the given planning. Constraints (10)–(13) indicate that the number of selected charges, the number of refining charges, the weight of the hot roll materials, and the total weight of slabs processed in the downstream production line are within the limits specified for production. Constraint (14) indicates the range of values of the decision variable.

3 Solution methodology

3.1 Improved surrogate absolute-value Lagrangian relaxation model

The study of improved surrogate absolute-value Lagrangian relaxation (ISAVLR) function model framework has been proposed by Liu et al. (2021), in this method, a piecewise linear function $F(x)$ is introduced by the convergence factor r to replace the absolute value penalty function by Bragin et al. (2018). Because the quadratic approximation difference of the (SAVLR) function cannot capture the quadratic growth characteristics of the quadratic function well when the multiplier is far from the optimal value and the degree of constraint violation is large, the absolute value function cannot impose a large enough penalty in the early stage of optimisation. Where the convergence factor r is used to construct an ISAVLR function, which introduces an absolute-value penalty term through the convergence factor r. The absolute-value term penalises the violation and thus improves the speed of convergence and the lower bound of the conventional Lagrangian relaxation (LR). The $F(x)$ piecewise function is as follows:

$$F(x) \equiv max\left[0,(4x-3),(-4x-3)\right] \tag{15}$$

The CBP problem is optimised based on the ISAVLR framework by a set of Lagrangian multipliers $\{u_i, i = 1,2,\ldots,N.\}$ relax constraint (7), and by introducing convergence factor r and piecewise function $F(x)$, the above problem model can be transformed into the following form:

$$(LR) Z_{ISAVLR}(u_i) = min(Z + B) \tag{16}$$

$$B = B_1 + B_2 \tag{17}$$

$$B_1 = \sum_{i=1}^{N} u_i \cdot \left(\sum_{j=1}^{N} x_{ij} - 1\right) \tag{18}$$

$$B_2 = r \cdot \sum_{i=1}^{N} F\left(\sum_{j=1}^{N} x_{ij} - 1\right) \tag{19}$$

s.t.(8)–(14). $i = \{1, 2, \ldots, N\}.j = \{1, 2, \ldots, N\}$.

$$Z_{ISAVLR}(u_i) = min = \left\{ \begin{array}{l} \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} \cdot x_{ij} + \varphi_4 \cdot \sum_{j=1}^{N} p_j^{TL} \cdot \left(TL \cdot x_{jj} - \sum_{i=1}^{N} x_{ij}\right) \\ +\varphi_5 \cdot \sum_{i=1}^{N} p_i^{US} \cdot \left(1 - \sum_{j=1}^{N} x_{ij}\right) \\ +\sum_{i=1}^{N} u_i \cdot \left(\sum_{j=1}^{M} x_{ij} - 1\right) + r \cdot \sum_{i=1}^{N} F\left(\sum_{j=1}^{M} x_{ij} - 1\right) \end{array} \right\} \tag{20}$$

s.t. (8)–(14). $i = \{1, 2, \ldots, N\}.j = \{1, 2, \ldots, N\}$.

Let $P_{ij} = \varphi_1 \cdot \left|G_i - G_j\right| \cdot p_{ij}^G + \varphi_2 \cdot \left|W_i - W_j\right| \cdot p_{ij}^W + \varphi_3 \cdot \left|D_i - D_j\right| \cdot p_{ij}^D$, formula (18) can be expressed as (20).

3.2 Dual problem

Since constraint (7) is relaxed, the optimal solution of the ISAVLR problem is not necessarily the optimal solution of the original problem. To get closer to the optimal solution of the original problem, the solution of the relaxation problem is replaced by the dual solution of the dual problem. It can be expressed as (21):

$$(LD)\, Z_{ISAVLR}^{D}(u_i) = \max \min Z_{ISAVLR}(u_i) \tag{21}$$

s.t. (8)–(14). $i = \{1, 2, \ldots, N\}.j = \{1, 2, \ldots, N\}$.

Let $P_{ij}^{'} = P_{ij} - p_j^{TL} - \varphi_5 \cdot p_i^{US}$, formula (21) can be expressed as (22):

$$Z_{ISAVLR}^{D}(u_i) = max\, min \left\{ \begin{array}{l} \varphi_4 \cdot TL \cdot \sum_{j=1}^{N} p_j^{TL} \cdot x_{jj} \\ +\sum_{i=1}^{N}\sum_{j=1}^{N} \left(P_{ij}^{'} + u_i\right) \cdot x_{ij} \\ +r \cdot \sum_{i=1}^{N} F\left(\sum_{j=1}^{N} X_{ij} - 1\right) \\ +\varphi_5 \cdot \sum_{i=1}^{N} p_i^{US} - \sum_{i=1}^{N} u_i \end{array} \right\} \tag{22}$$

s.t. (8)–(14). $i = \{1, 2, \ldots, N\}.j = \{1, 2, \ldots, N\}$.

To solve the optimisation model of CBP more efficiently, formula (22) was transformed into a form containing a subproblem. Each subproblem represents an optimal value of CBP with charge *j* as the clustering centre for the optimal solution. The conversion form is as formula (23).

$$Z_{ISAVLR}^{D}\left(u_i\right)=\max\min\left\{\sum_{j=1}^{N}v_j\left(x_{ij}\right)\cdot x_{jj}+\varphi_5\cdot\sum_{i=1}^{N}p_i^{US}-\sum_{i=1}^{N}u_i\right\} \tag{23}$$

s.t. (8)–(14). $i=\{1,2,\ldots,N\}.j=\{1,2,\ldots,N\}$.

3.3 Dynamic programming to solve subproblems

Based on constraints (9) and $x_{jj}\in\{0,1\}, j=\{1,2,\ldots,N\}$. Formula (23) can be decomposed into M subproblems with a single cast as the optimal value for solving, and each cast is based on the charge j as the clustering centre and the form of subproblems is as follows:

$$\left(LD_j\right)v_j\left(x_{ij},y\right)=min\left\{\begin{array}{c}\varphi_4\cdot TL\cdot p_j^{TL}+\sum_{i=1}^{N}\left(P_{ij}^{'}+u_i\right)\cdot x_{ij}\\+r\cdot\sum_{i=1}^{N}\max\left[0,\left(4x_{ij}-7\right),\left(-4x_{ij}+1\right)\right]\end{array}\right\},$$

$$j=\{1,2,\ldots,N\}. \tag{24}$$

s.t. (8), (10)–(14). $i=\{1,2,\ldots,N\}$.

According to the linearisation method on page 150 of Boyd and Vandenberghe (2004), the auxiliary target variable y is introduced to linearise the function $\sum_{i=1}^{N}max\left[0,\left(4x_{ij}-7\right),\left(-4x_{ij}+1\right)\right]$, $j=1,2,\ldots N$. (y is the least upper bound of piecework linear function), and the subproblem is transformed into:

$$\left(LD_j\right)v_j\left(x_{ij},y\right)=min\left\{\varphi_4\cdot TL\cdot p_j^{TL}+\sum_{i=1}^{N}\left(P_{ij}^{'}+u_i\right)\cdot x_{ij}+r\cdot y\right\},$$

$$j=\{1,2,\ldots,N\}. \tag{25}$$

s.t. (8), (10)–(14) $i=\{1,2,\ldots,N\}$. and:

$$0\le y \tag{26}$$

$$4x_{ij}-7\le y, i=\{1,2,\ldots,N\}. \tag{27}$$

$$-4x_{ij}+1\le y, i=\{1,2,\ldots,N\}. \tag{28}$$

p_j^{TL}, TL and φ_4 are constants in formula (25), and the decision variable x_{ij}, the variable y and the objective function v_j are the unknowns to be solved. The model of LD_j is to satisfy the objective function $min\left\{v_j\left(x_{ij},y\right)\right\}$ and all of its constraints: (8), (10)–(14), (26)–(28), where constraint (8) is the machine capability constraint. Therefore the LD_j can be abstracted to be solved as a 0-1 knapsack problem. And solve the problem based on a dynamic programming algorithm.

N charges are combined into M casts, and the tundish lifespan is TL. The charges combined into each cast will result in penalties due to differences in physical properties (steel grade, width, delivery time, etc.). Therefore, the objective optimisation decision can be expressed as, under the premise of not exceeding the lifespan of tundish, the decision to assign which charges to cast will minimise the penalty brought by the different physical properties of these charges. However, different from the traditional backward dynamic programming algorithm, there is an additional penalty term $r \cdot y$ in formula (25), where y is a function based on the value of the variable x_{ij}, so y can be used as a coefficient and r as a penalty term in the objective function. According to constraints (8), (26)–(28) it follows that $\left\{y = 0 \mid \text{when } x_{ij} = 1\right\}$, and $\left\{y = 1 \mid \text{when } x_{ij} = 0\right\}$.

Based on backward dynamic programming (Ibaraki, 1987) the service lifespan of a tundish is divided into $\{TL_{sum}^1, TL_{sum}^2, \ldots, TL_{sum}^d, \ldots, TL_{sum}^D \mid TL_{sum}^2 - TL_{sum}^1 = \ldots = TL_{sum}^d - TL_{sum}^{d-1} = \ldots = TL_{sum}^D - TL_{sum}^{D-1} = \delta\}$. ($TL_{sum}^D = TL$) capacity stages. $\overline{t}\left(i, TL_{sum}^d\right)$ is the ideal target penalty generated in the optimisation process when the capacity stage number is TL_{sum}^d for the first i charges. $g[i]$ is the service lifespan of tundish consumed by charge i. $P_{ij}^{'}$ is the penalty caused by assigning the i charge for production in the cast with j as the clustering center. $P_{ij}^{''} = -P_{ij}^{'}$. T denotes the set of charge attribute difference penalties for clustering into a cast. T' denotes the set of charge attribute difference penalties that are not clustered into a cast. R denotes the set of additional penalties associated with charges that are not clustered into a cast. a and a' denote the element position indexes, for example, in the set T, t_{aj} denotes the optimal objective value of the ath position with j as the clustering center. β denotes the state parameter, which is used during iteration to update convergence factor r. The recursive equations based on backward dynamic programming and the optimisation order are given in Table 2.

Table 2 Backward dynamic programming optimisation rules (see online version for colours)

No.	Weight	Quantization parameter	Tundish lifespan				
			TL_{sum}^1	…	TL_{sum}^d	…	TL_{sum}^D
1	$g[1]$	$P_{1j}^{''}$	↑	→	→	→	Objective value
…	…	…	↑		Recursion relation:		↑
i	$g[i]$	$P_{ij}^{''}$	↑		$\bar{t}(i, TL_{sum}^d) = max\begin{cases} \bar{t}(i-1, TL_{sum}^d), \\ \bar{t}(i-1, TL_{sum}^d - g[i]) + P_{ij}^{''} \end{cases}$		↑
…	…	…	↑				↑
N	$g[N]$	$P_{Nj}^{''}$	↑	→	→	→	↑

Based on the above description, this paper designs a subproblem optimisation method based on the inner heuristic method.

In the inner heuristic algorithm, the subproblem feasible solution is solved and optimised. Based on equation (25) it is known that the subproblem objective function is composed of three parts. The first part is the constant term $\left\{\varphi_4 \cdot TL \cdot p_j^{TL}\right\}$. The second part is the penalty due to the attribute differences during the charge clustering process in each cast, to facilitate the expression in the algorithm, let the part be expressed as

$\left\{t_{ij} = \sum_{i=1}^{N}\left(P_{ij}^{'} + u_i\right) \cdot x_{ij}\right\}$. The third part is the additional penalty $\{r \cdot y\}$ for unselected charges. The inner heuristic first abstracts the second part of the subproblem as a knapsack problem, for which the dynamic programming method is used to solve, and based on the results obtained it can compute the sets T, T' and R. And arrange the elements of the three sets in increasing order. $T = \left\{t_{aj}, t_{(a+1)j}, \ldots, t_{(n-1)j}, t_{nj}\right\}$, it denotes the set of charge attribute difference penalties for clustering into a cast. $T' = \left\{t_{a'j}, t_{(a+1)^{'}j}, \ldots, t_{(n-1)^{'}j}, t_{n'j}\right\}$, it denotes the set of charge attribute difference penalties that are not clustered into a cast. $R = \left\{r \cdot y\left(x_{a'j}\right), r \cdot y\left(x_{(a+1)^{'}j}\right), \ldots, r \cdot y\left(x_{(n-1)^{'}j}\right), r \cdot y\left(x_{n'j}\right)\right\}$, it denotes the set of additional penalties associated with charges that are not clustered into a cast. Finally, if the sum of the charge attribute difference penalty clustered into the cast and the additional penalty not clustered into the cast is less than the charge attribute difference penalty not clustered into the cast, the current solution is optimal, otherwise, it is adjusted. The pseudo-code of the inner heuristic algorithm is shown in Figure 2.

Figure 2 Pseudo-code of the inner heuristic algorithm

Inner heuristic

Initialization:

1.Set i=1, j=1, a=1, $a^{'} = 1$, r=10, $\beta = 1.1$, $TL_{sum}^{d} = 1$, $T = null$, $T^{'} = null$, $R = null$

2. Total number of charges N

3. Service lifespan of a tundish TL_{sum}^{D}

For j=1 **to** N

/* *Solve the solution of subproblem* */

For $TL_{sum}^{1} = 1$ **to** TL_{sum}^{D}

For i=1 **to** N

Obtain the value of the objective value function t_{ij}

Obtain the decision variable x_{ij}

End For

End For

/* *Optimization subproblem feasible solution* */

Update the following set according to $\{t_{ij}\}$ and $\{x_{ij}\}$

$T = \{t_{aj}, t_{(a+1)j}, \ldots, t_{(n-1)j}, t_{nj}\}$ /* *charge clustering penalty increasing order set for grouping into cast* */

$T^{'} = \{t_{a'j}, t_{(a+1)'j}, \ldots, t_{(n-1)'j}, t_{n'j}\}$ /* *charge clustering penalty incremental order set for ungrouped cast* */

$R = \{r \cdot y(x_{a'j}), r \cdot y(x_{(a+1)'j}) \ldots, r \cdot y(x_{(n-1)'j}), r \cdot y(x_{n'j})\}$ /* *Additional penalty increment set for charges not grouped into cast* */

While $\sum_{a \in n} t_{aj} + r \cdot \sum_{a' \in n'} y(x_{a'j}) \geq \sum_{a' \in n'} t_{a'j}$ **Do**

$t_{nj} \leftarrow t_{a'j}$

$r \leftarrow \frac{r}{\beta}$

Update T, $T^{'}$, R, $\{t_{ij}\}$, $\{x_{ij}\}$

End While

$v_j(x_{ij}, y) \leftarrow \varphi_4 \cdot TL \cdot p_j^{TL} + \sum_{a \in n} t_{aj} + r \cdot \sum_{a' \in n'} y(x_{a'j})$

End For

Output $\{v_j\}$, $\{x_{ij}\}$

In the process of solving the subproblem of CBP based on backward dynamic programming, the tundish life of the cast is divided into TL_{sum}^{D} stages. In the optimisation process of each stage, it is necessary to judge the number of charges from 1 to N_j (N_j is the total number of charges when the clustering center charge j is not included. Whether $N_j \in N$) is combined into CBP with charge j as the clustering center to meet the optimal value under the constraint of not exceeding tundish life. So, it is worth noting that the complexity of the optimisation process is $O\left(\left(N_j\right)^2\right)$.

3.4 Construction of feasible solutions to the original problem

The Lagrangian dual function $Z_{ISAVLR}^{D}\left(u_i\right)$ is a non-differentiable, piecewise linear concave function. At the same time, this function has a large scale in the process of CBP optimisation, so it is difficult to optimise $Z_{ISAVLR}^{D}\left(u_i\right)$ function efficiently (Sun et al., 2022a). Therefore, this paper develops a heuristic optimisation method and proposes a controlled direction of the gradient iterative strategy based on the ISAVLR framework, which is combined with the local search (LS) (Lorena and Senne, 2003) algorithm in the optimisation process. The controlled subgradient iteration direction (Bragin et al., 2014; Sun et al., 2022b) can effectively avoid the problem of zigzagging caused by the obtuse angle of the adjacent two iteration directions during the iteration process of the Lagrangian relaxation subproblem, which greatly improves the solution speed. The LS algorithm can further select the appropriate charge as the clustering center to ensure the optimal objective function of the subproblem. Figure 3 shows the optimisation sequence diagram.

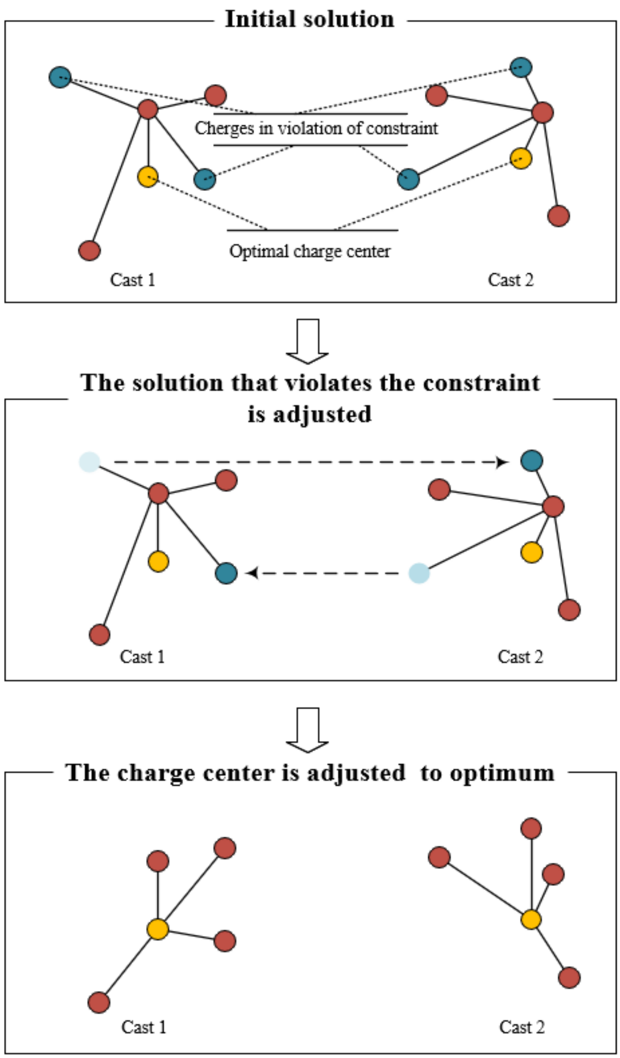
The outer heuristic algorithm is described in three stages. In the outer heuristic-I, the approximate optimised solution and the best dual value are obtained by iterating the Lagrangian multipliers. The relationship between the objective function values of the original problem, the dual problem, and the relaxation problem obtained based on the ISAVLR framework is $Z > Z_{ISAVLR}^{D}\left(u_i\right) > Z_{ISAVLR}\left(u_i\right)$. The core idea of the proposed method is to keep converging to the original problem objective by iterating the best dual values, and finally, the original problem objective function value is approximated by the best dual value when the stopping condition is satisfied. An iterative strategy with a controlled gradient direction is used for Lagrangian multiplier updating. The Lagrangian multipliers obtained from each iteration are fed into the inner heuristic, which solves the objective function and decision variables of the subproblem based on the known multipliers and feeds them into the outer heuristic to compute the best dual values. The stopping conditions are as follows.

$$\| u_i^{(m+1)} - u_i^{(m)} \| < \varepsilon_1 \tag{29}$$

$$\| x_{ij}^{(m+1)} - x_{ij}^{(m)} \| < \varepsilon_2 \tag{30}$$

The iteration can be stopped when one of the conditions of equation (29) or (30) is satisfied. The pseudo-code of the outer heuristic-I is as Figure 4.

Figure 3 Optimisation sequence diagram (see online version for colours)



In outer heuristic-II, the feasible solutions that violate the constraints being relaxed are adjusted. To reduce the difficulty of solving the model, the charge allocation constraint (7) is relaxed during the optimisation process of the ISAVLR framework and the original problem model is transformed into a CBP model with a separable structure. Therefore, the feasible solution output by outer heuristic-I may not satisfy the constraint (7). The outer heuristic-II is mainly used to adjust the feasible solutions that violate the constraints (7); the feasible solutions of the outer heuristic-I output are further optimised. The pseudo-code of the outer heuristic-II algorithm is shown in Figure 5.

Figure 4 Pseudo-code of the outer heuristic-I algorithm

```
Outer heuristic-I
Initialization:
   1. Set m=1, x^(0)=0, u^(0)=0, β = 1.6, γ = 1.2 , δ = 0.6
   2. Total number of charges N
While the stop criterion is not satisfied Do
   /* Whether the surrogate optimality condition is satisfied */
   If Z^D_ISAVLR(u^(m), x^(m)) < Z^D_ISAVLR(u^(m), x^(m-1)) Do
      Save x^(m)
   Else
      x^(m) ← x^(m-1)
   End If
   For i=1 to N
      h^(m)(u^(m)) ← Σ_{j=1}^{N} x_ij^(m) − 1  /* Calculate the subgradient */
      ξ^(m) ← max{0, −β( d̂^(m)(u_i^(m-1))^T · d̂^(m)(u_i^(m-1)) / d̂^(m)(u_i^(m))^T · d̂^(m)(u_i^(m)) )}
      d̂^(m)(u_i^(m)) ← h^(m)(u_i^(m)) + ξ^(m) · d̂^(m-1)(u_i^(m-1)) /* Modified subgradient */
      ρ ← 1 − 1/m^δ
      α^(m) ← 1 − 1/(γ m^ρ)
      s^(m) ← α^(m) · s^(m-1) · d̂^(m)(u_i^(m-1))^T · d̂^(m)(u_i^(m-1)) / d̂^(m)(u_i^(m))^T · d̂^(m)(u_i^(m)) /* Calculate step size */
      u_i^(m+1) ← max{0, u_i^(m) + s^(m) · d̂^(m)(u_i^(m))} /* Updating the multipliers */
   End For
   v_j^(m+1) ← {v_j(u^(m+1))}, x^(m+1) ← {x_ij^(m+1)} /* Inner heuristic for solving subproblems */
   u^(m+1) ← {u_i^(m+1)}
   Calculate Z^D_ISAVLR(u^(m), x^(m)) according to Eq. (23)
   m ← m + 1
End While
Output x^(m), u^(m)
```

Figure 5 Pseudo-code of the outer heuristic-II algorithm

```
Outer heuristic-II
Initialization:
   1. Total number of charges N
For i=1 to N
   While Σ_{j=1}^{N} x_ij^(m) ≥ 1 Do
      t̂_ij̃ ← min_{j=1,2,...,N} t_ij
      x_ij̃^(m) ← 1
      {x_ij^(m) | j ≠ j̃} ← {0}
      Update x^(m) ← {x_ij^(m)}
   End While
End For
Output x^(m)
```

In outer heuristic-III, a local search (LS) algorithm is used to adjust the clustering centers for each CBP to further optimise the objective function value. The feasible solution for the outer heuristic-II output presents the charge numbers in the CBP, and the selection of which charge to use as the clustering center still needs to be further optimised, as the value of the objective function is directly affected by the selection of the clustering center. The proposed method is implemented by enumerating the charges in each CBP, the function value of each charge as a clustering center is calculated, and finally, the charge that can achieve the smallest function value is selected as the most clustering center in the cast. The symbols of outer heuristic-III are defined as well as the flow as follows. The number of casts is M. The total number of charges in cast j is X_j, $\{X_j \mid j = 1, 2, \ldots, M\}$. The index of the clustering center in cast j is $c_j, \{c_j \in X_j\}$; The other charges in cast j are indexed by i, $\{i \in X_j, i \neq c_j\}$. The value of the objective function solved with c_j as the clustering center in cast j is $z_j = \sum_{k \in X_j} \hat{t}_{kc_j}$. The pseudo-code of the outer heuristic-III algorithm is shown in Figure 6. The flowchart of the overall algorithm is shown in Figure 7.

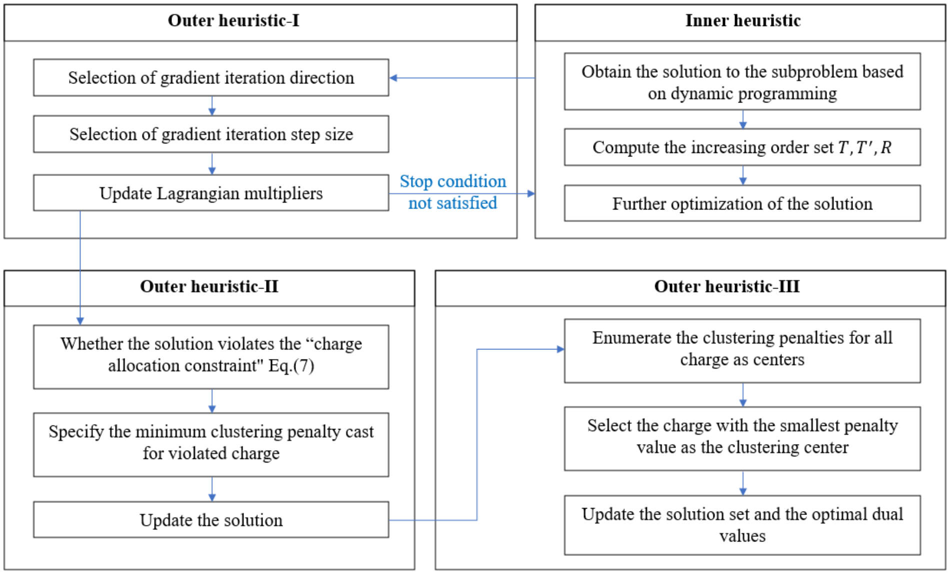
Figure 6 Pseudo-code of the outer heuristic-III algorithm

```
Outer heuristic-III
Initialization:
  1. Total number of casts M
  2. The set of charge quantities contained in all casts {X_j | j = 1,2, ..., M}
For j = 1 to M
  For i = 1 to X_j
    z_i = Σ_{k∈X_j} t̂_ki
    If z_i < z_j Do
      z_j ← z_i
      c_j ← i
    Else
      z_j ← z_j
      c_j ← c_j
    End If
  End For
End For
Update x ← {x_ij}
Solving for Z^D_ISAVLR (u, x) according to Eq. (23)
Output Z^D_ISAVLR, x
```

4 Computational results

To test the optimisation performance of CBP based on the ISAVLR, and to test the characteristics of its solutions, a simulation experiment is carried out on a random data example of a large steel mill.

Figure 7 Flowchart of the overall algorithm (see online version for colours)



4.1 Parameter setting

To analyse the computational performance of the algorithm proposed in this paper in solving the optimisation process of CBP, this paper is based on the LR, SAVLR, and ISAVLR. The proposed algorithms are evaluated for the following performance indexes: duality gaps (%), running times, and iteration numbers. Where the duality gaps calculation method is $G = \frac{(UB-LB)}{LB} \times 100\%$, UB is the upper bound calculated based on the feasible solution in the original problem $Z_{ISAVLR}(u_i)$, LB is the lower bound calculated based on the approximate feasible solution in the dual problem $Z_{ISAVLR}^{D}(u_i)$. The parameters are as follows: $\varepsilon_1 = 1e-3$, $\varepsilon_2 = 1e-5, \beta = 1.02, \gamma = 1.05, \delta = 0.25$. The specific values of penalty parameters are given in Table 3.

Table 3 The value of parameters

TL	p_{ij}^{G}	p_{ij}^{W}	p_{ij}^{D}	p_{j}^{TL}	p_{i}^{US}	φ_k	r
20	15	12	20	15	100	0.2	10

The above algorithm implementation process is in the same simulation environment optimisation effect, several algorithms are implemented in Matlab R2022b and installed Intel Core i5-1135 2.4GHz CPU, Windows 11 operating system (64 bit) of the PC running.

4.2 Problem instances

In this paper, 20 groups of case data are used for simulation, and each 4 groups of data is a batch. In each batch, the number of charges of each group of data is the same, which is {100,150,200,250,300}. To test under different conditions of LR, SAVLR, and ISAVLR algorithm in duality gaps, running times, and iteration numbers performance indicators, each group of charge steel grade in data quantisation parameters by randomly selected from the set {5,6,7,8,9,10,11,12,13,14,15}. Table 4 shows the optimisation results and average values of duality gaps, running times, and iteration numbers based on three different algorithms when the number of different charges and the number of the same charges correspond to different randomly selected steel grades.

Table 4 Computational results

No.	*Number of charges*	*Steel grade*	*Duality gaps* (%)			*Running times* (s)			*Iteration numbers*		
			LR	*SAVLR*	*ISAVLR*	*LR*	*SAVLR*	*ISAVLR*	*LR*	*SAVLR*	*ISAVLR*
1	100	5	1.71	1.46	1.17	13.95	10.64	8.35	86.00	73.00	68.00
2	100	6	2.13	1.65	1.42	16.47	11.93	10.17	67.00	61.00	52.00
3	100	8	2.60	2.12	1.73	21.73	15.98	12.04	75.00	58.00	49.00
4	100	10	2.84	2.30	1.96	40.69	24.73	14.21	63.00	44.00	24.00
Average			**2.32**	**1.88**	**1.57**	**23.21**	**15.83**	**11.19**	**72.75**	**59.00**	**48.25**
5	150	6	2.24	2.03	1.82	56.35	40.86	26.73	118.00	106.00	86.00
6	150	5	1.94	1.76	1.73	43.81	27.81	24.59	142.00	115.00	83.00
7	150	4	1.68	1.52	1.23	49.53	28.53	22.08	174.00	133.00	113.00
8	150	9	2.68	1.94	1.67	66.79	49.48	30.32	169.00	127.00	107.00
Average			**2.14**	**1.81**	**1.61**	**54.12**	**36.67**	**25.93**	**150.75**	**120.25**	**97.25**
9	200	10	3.24	2.53	2.14	82.53	54.7	40.39	157.00	114.00	94.00
10	200	12	3.41	3.06	2.56	71.42	51.92	41.27	143.00	111.00	91.00
11	200	13	3.47	3.09	2.14	78.13	58.47	43.04	176.00	132.00	109.00
12	200	4	2.08	1.76	1.56	65.52	45.36	32.93	229.00	183.00	153.00
Average			**3.05**	**2.61**	**2.10**	**74.40**	**52.61**	**39.41**	**176.25**	**135.00**	**111.75**
13	250	6	2.50	1.94	1.72	84.76	64.29	55.2	208.00	167.00	137.00
14	250	7	2.64	2.23	2.01	94.89	69.75	59.38	201.00	159.00	129.00
15	250	8	3.05	2.45	2.27	105.87	75.34	63.42	195.00	161.00	131.00
16	250	10	3.78	3.3	2.96	119.46	76.86	65.74	194.00	159.00	129.00
Average			**2.99**	**2.48**	**2.24**	**101.25**	**71.56**	**60.94**	**199.50**	**161.50**	**131.52**
17	300	12	3.86	3.32	2.67	144.63	114.58	83.03	267.00	206.00	174.00
18	300	15	4.36	3.51	3.24	176.47	126.29	93.46	202.00	148.00	116.00
19	300	14	4.32	3.66	2.86	128.52	99.31	87.25	238.00	182.00	152.00
20	300	13	3.93	2.83	2.50	126.21	96.74	84.86	245.00	208.00	159.00
Average			**4.12**	**3.33**	**2.82**	**143.96**	**109.23**	**87.15**	**238.00**	**186.00**	**150.25**

As shown in Figures 8–10, it can be seen that with the increase in the number of charges, the duality gaps, running times, and iteration numbers all show an overall upward trend under the action of the three algorithms. Meanwhile, ISAVLR is superior to SAVLR and LR in duality gaps, running times, and iteration numbers.

1 The average value of duality gaps in {100,150,200,250,300} group is ISAVLR: {1.57%, 1.61%, 2.1%, 2.24%, 2.82%}, SAVLR: {1.88%, 1.81%, 2.61%, 2.48%, 3.33%}, LR: {2.32%, 2.14%, 3.05%, 2.99%, 4.12%}, based on the average data, it can be concluded that the duality gaps generated by ISAVLR in the optimisation process is the smallest, indicating that the optimisation quality of this algorithm is the best among the three algorithms.

2 The average running time in {100,150,200,250,300} group is ISAVLR: {11.19, 25.93, 39.41, 60.94, 87.15}, SAVLR: {15.83, 36.67, 52.61, 71.56, 109.23}, LR: {23.21, 54.12, 74.40, 101.25, 143.96}.

3 The average number of iterations in {100,150,200,250,300} group is ISAVLR: {48.25, 97.25, 111.75, 131.52, 150.25}, SAVLR: {59.00, 120.25, 135.00, 161.50, 186.00}, LR: {72.75, 150.75, 176.25, 199.50, 238.00}.

Figure 8 Comparison of duality gaps (%) (see online version for colours)

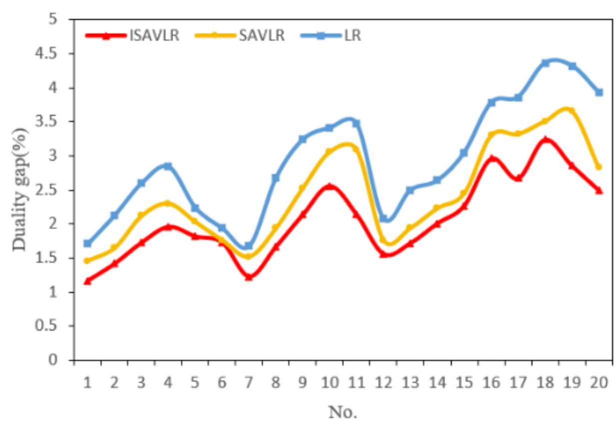


Figure 9 Comparison of running times (see online version for colours)

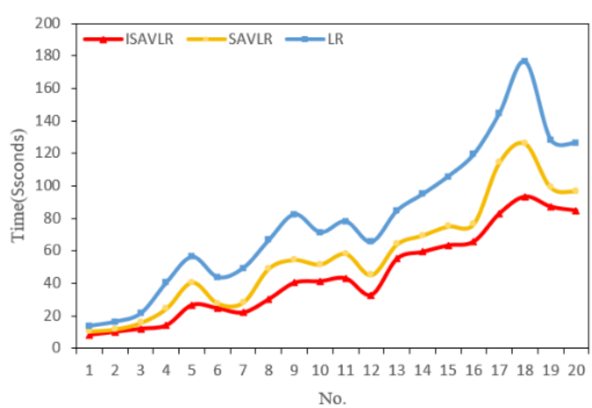
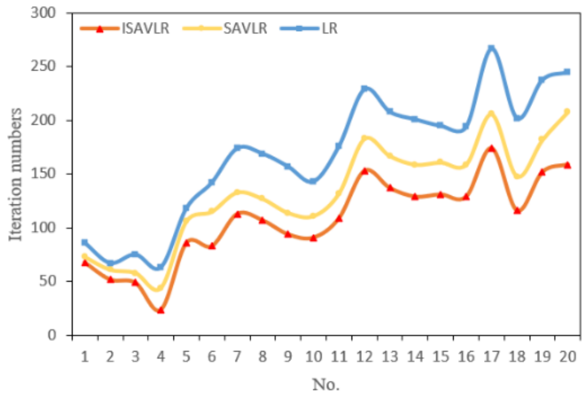


Figure 10 Comparison of iteration numbers (see online version for colours)



Based on the results in (2) and (3), it can be seen that among the three algorithms, ISAVLR is the smallest in terms of the running time and iteration numbers, indicating that the optimisation efficiency of ISAVLR is the best among the three algorithms.

5 Conclusion

In this paper, a heuristic optimisation method proposes a controlled direction of the gradient iterative strategy based on the ISAVLR framework for the optimisation problem of CBP. On this basis, the constraint condition "each charge can only be combined into one cast" is relaxed, and a piecewise linear function is introduced to accelerate the convergence rate. The objective function is decomposed into subproblems with a single cast as the optimisation unit, and the backward dynamic programming algorithm is used to solve the subproblems. Finally, the experimental results show that the proposed optimisation method can guarantee optimisation efficiency as well as optimisation quality.

Acknowledgements

This work was partly supported by the Natural Science Foundation of Hebei Province of China (F2023501006), National Natural Science Foundation of China (92367106).

References

Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*, Cambridge University Press, Cambridge, UK.

Bragin, M.A., Luh, P.B., Yan, B. and Sun, X.R. (2018) 'A scalable solution methodology for mixed-integer linear programming problems arising in automation', *IEEE Transaction on Automation Science and Engineering*, Vol. 16, No. 2, pp.1–11.

Bragin, M.A., Luh, P.B., Yan, J.H., Yu, N. and Stern, G.A. (2014) 'Convergence of the surrogate lagrangian relaxation method', *Journal of Optimization Theory and Applications*, Vol. 164, No. 1, pp.173–201.

Chang, S.Y., Chang, M-R. and Hong, Y. (2000) 'A lot grouping algorithm for a continuous slab caster in an integrated steel mill', *Production Planning and Control*, Vol. 11, No. 4, pp.363–368.

Ibaraki, T. (1987) 'Enumerative approaches to combinatorial optimization-part II', *Annals of Operations Research*, Vol. 11, Nos. 1–4, pp.345–602.

Liu, A.B., Luh, P.B., Yan, B. and Bragin, M.A. (2021) 'A novel integer linear programming formulation for job-shop scheduling problems', *IEEE Robotics and Automation Letters*, Vol. 6, No. 3, pp.5937–5944.

Liu, X.J., Ni, Z.H. and Qiu, X.L. (2016) 'Application of ant colony optimization algorithm in integrated process planning and scheduling', *International Journal of Advanced Manufacturing Technology*, Vol. 84, No. 4, pp.393–404.

Long, J.Y., Sun, Z.Z., Chen, H.B. and Hong, Y. (2018) 'Variable neighbourhood search for integrated determination of charge batching and casting start time in steel plants', *Journal of Intelligent and Fuzzy Systems*, Vol. 34, No. 2018, pp.3821–3832.

Lorena, L.A.N. and Senne, E.L.F. (2003) 'Local search heuristics for capacitated p-median problems', *Networks and Spatial Economics*, Vol. 3, No. 4, pp.407–419.

Ma, T.M., Luo, X.C. and Chai, T.Y. (2015) 'Multi-objective tundish planning model and hybrid optimization algorithm', *Journal of Systems Engineering*, Vol. 30, No. 4, pp.451–456.

Sun, L.L. and Mao, K. (2022a) 'An efficient and effective approach for the scheduling of steelmaking-continuous casting process with multi different refining routes', *IEEE Robotics and Automation Letters*, Vol. 7, No. 4, pp.10454–10461.

Sun, L.L., Lu, T.Y., Li, Z., Li, Y., Yu, Y.Q. and Liu, J.Y. (2022b) 'Research on steelmaking-continuous casting production scheduling problem with uncertain processing time based on lagrangian relaxation framework', *International Journal of Automation and Control*, Vol. 16, No. 1, pp.87–107.

Sun, L.L., Sha, S.Y., Qu, Q.X., Ying, Y., Yu, M. and Wang, J. (2021) 'A survey of integrated optimization method of batch planning and production scheduling for steelmaking-continuous casting process under uncertainties based on Lagrangian relaxation framework', *Control and Decision*, Vol. 36, No. 8, pp.1973–1803.

Tang, L.X. and Luo, J.X. (2007) 'A new ILS algorithm for cast planning problem in steel industry', *ISIJ International*, Vol. 47, No. 3, pp.443–452.

Tang, L.X. and Wang, G.S. (2008) 'Decision support system for the batching problems of steelmaking and continuous-casting production', *Omega*, Vol. 36, No. 6, pp.976–991.

Tang, L.X., Wang, G.S., Liu, J.Y. and Liu, J.Y. (2011) 'A combination of lagrangian relaxation and column generation for order batching in steelmaking and continuous-casting production', *Naval Research Logistics*, Vol. 58, No. 4, pp.370–388.

Wang, Y.Z., Zheng, Z., Zhu, M.M., Zhang, K.T. and Gao, X.Q. (2022) 'An integrated production batch planning approach for steelmaking-continuous casting with cast batching plan as the core', *Computers and Industrial Engineering*, Vol. 173, pp.976–991.

Xu, W.J. and Wang, G.S. (2015) 'An improved differential evolution for batching problem in steelmaking production', *IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, pp.377–384.

Xu, W.J., Zou, F. and Tang, L.X. (2016) 'A subpopulation-based differential evolution algorithm for scheduling with batching decisions in steelmaking-continuous casting production', *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, pp.2784–2790.

Xu, Z.J., Zheng, Z. and Gao, X.Q. (2021) 'Operation optimization of the steel manufacturing process: a brief review', *International Journal of Minerals, Metallurgy and Materials*, Vol. 28, No. 8, pp.1274–1287.

Yang, F., Li, Q.Q. and Wang, G.R. (2015) 'Hybrid improved algorithm for cast planning problem with flexible width', *Control and Decision*, Vol. 30, No. 2, pp.348–352.

Yang, F., Li, Q.Q., Liu, S. and Wang, G.R. (2014) 'Hybrid heuristic-cross entropy algorithm for cast planning problem', *Computer Integrated Manufacturing Systems*, Vol. 20, No. 9, pp.2241–2247.

Yi, J., Jia, S.J., Du, B. and Liu, Q. (2019) 'Multi-objective model and optimization algorithm based on column generation for continuous casting production planning', *Journal of Iron and Steel Research International*, Vol. 26, No. 3, pp.242–250.

Yi, J., Tan, S.B., Li, W.G. and Du, B. (2012) 'Hybrid optimization algorithm for solving combining tundish MTSP model on continuous casting plan', *Journal of Northeastern University*, Vol. 33, No. 9, pp.1235–1239.

Zhu, W.Y., Li, Y.P., Sun, L.L., Hou, T.B., Bai, S., Yuan, B.L. and Zhao, L.G. (2021) 'Research on batch planning method of steelmaking-continuous casting tundish', *3rd International Conference on Industrial Artificial Intelligence*, Shenyang, China.