Advanced Engineering Informatics 66 (2025) 103402

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Dynamic multiobjective operation optimization of blast furnace ironmaking process☆

Yumeng Zhao [a,b], Xianpeng Wang [c,d,*], Xiangman Song [a]

[a] National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang, 110819, Liaoning, China
[b] Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education, Shenyang, 110819, Liaoning, China
[c] Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Northeastern University, Shenyang, 110819, Liaoning, China
[d] Liaoning Key Laboratory of Manufacturing System and Logistics Optimization, Northeastern University, Shenyang, 110819, Liaoning, China

ARTICLE INFO

Keywords:
Blast furnace ironmaking
Dynamic multiobjective optimization
Evolutionary algorithm
Process optimization

ABSTRACT

As the largest energy-consuming and carbon-emitting facility in the iron and steel industry, blast furnace needs to dynamically adjust its operating parameters according to the production conditions to minimize energy consumption and carbon emissions. Most of the previous studies focus only on the static operation optimization of blast furnace, which does not allow for the timely updating of operating parameters and the effective reduction of carbon emissions and production costs. Therefore, in this paper, a dynamic multiobjective operation optimization model for blast furnace, which integrates the identification approach of production condition variations, is developed and a tensor-based dynamic multiobjective evolutionary algorithm (T-DCMOEA/D) is devised to achieve cleaner production and lower cost of ironmaking process. The computational results based on real-world blast furnace operation optimization problems demonstrate that the proposed model and algorithm can effectively address the dynamic variations in the actual blast furnace production, which ensures that the blast furnace always operates in an optimized state. Furthermore, it is shown that our T-DCMOEA/D significantly outperforms some of the state-of-the-art dynamic multiobjective evolutionary algorithms in the literature, with regard to faster convergence, higher adaptivity, and better performance.

1. Introduction

Ironmaking process in a blast furnace plays a vital role in industrial production, accounting for the highest energy consumption and carbon emissions in the iron and steel industry [1]. The operational process of a blast furnace is illustrated in Fig. 1. During production, iron ore and coke are firstly loaded into the furnace from the top, forming layers of materials. Simultaneously, hot air, oxygen-enriched air, and pulverized coal are injected into the furnace through tuyeres located around the lower region. After continuous heating, reduction, melting and decarburization to form pig iron, the impurities in the iron ore are combined with the added flux to form slag. Finally, the pig iron is discharged and loaded into the molten iron tank, slag is discharged from the slag outlet, and blast furnace gas is exported from the top of the furnace [2]. This process is a very complex nonlinear dynamic process with complex physico-chemical coupling [3].

With the global emphasis on energy saving and emission reduction, blast furnace ironmaking needs to not only maintain smooth operation but also needs to achieve energy conservation and emission reduction [4,5]. Numerous control parameters in the blast furnace have a significant impact on these objectives. In actual production, the control parameters in the blast furnace are usually set according to manual experience, which has great operational risks and is difficult to realize the energy conservation and emission reduction. Therefore, the operation optimization of the blast furnace ironmaking process has been a hot and challenging topic in industrial and academic circles [6].

Due to the complexity of the blast furnace ironmaking process, the existing research on blast furnace operation optimization is usually categorized into three categories: mechanism-based optimization, data-driven optimization, and mechanism-data fusion-based optimization. In the field of mechanism-based optimization, Zhou et al. [7] constructed a mathematical model for blast furnace dosing and operation based on industrial principles, with cost and carbon emissions as objectives, and obtained the resulting solutions by using the NSGA-II algorithm. Zou

☆ This research was supported by the Major Program of National Natural Science Foundation of China (72192830, 72192831), the Fund for the National Natural Science Foundation of China (62473086), the 111 Project (B16009), and the Science and Technology Plan Joint Program (Key R&D Program Project) of Liaoning Province (2023JH2/101800047).

* Corresponding author.
E-mail addresses: zhaoym@stumail.neu.edu.cn (Y. Zhao), wangxianpeng@ise.neu.edu.cn (X. Wang), songxiangman@ise.neu.edu.cn (X. Song).

https://doi.org/10.1016/j.aei.2025.103402
Received 23 December 2024; Received in revised form 28 March 2025; Accepted 24 April 2025
Available online 13 May 2025

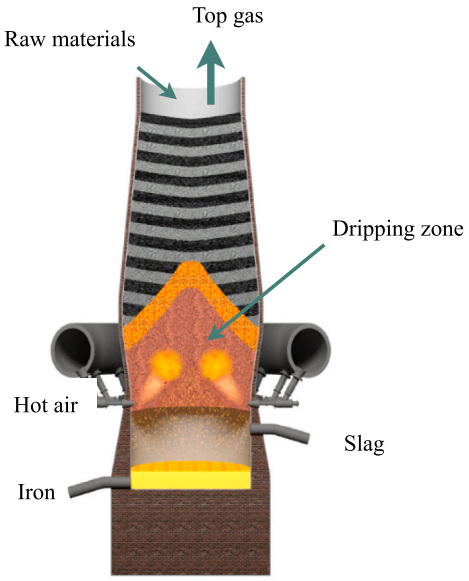


Fig. 1. Blast furnace operation process.

et al. [8] considered the various dynamics of the actual allocation process in the steel industry and proposed the KPLSEA algorithm to deal with this realistic multiobjective optimization problem. Liu et al. [9] developed a single-objective blast furnace dosing model and solved it with the DHHO-HITS metaheuristic algorithm. In terms of data-driven operation optimization, Li et al. [10] constructed a data-driven model for the burden surface of the blast furnace using an adaptive particle swarm-based extreme learning machine and solved the model using the differential evolution algorithm. Zhang et al. [11] used deep learning techniques to predict the hot metal temperature in blast furnace ironmaking. Mahanta and Chakraborti [12] implemented a data-driven approach to construct a blast furnace ironmaking model and solved it using the reference vector-guided evolutionary algorithm RVEA. Zhu et al. [13] utilized a modified Swin Transformer method to detect steel surface defects. Parihar et al. [14] used a large amount of data related to blast furnace variables to develop a data-driven machine learning model. They employed a genetic algorithm to maximize furnace productivity and minimize fuel consumption. In terms of mechanism-data fusion-based optimization, Li et al. [4] integrated measured data with production principles to construct a burden surface model for the blast furnace. A hybrid kinetic model was developed by Azadi et al. [15] for the prediction of molten iron silica content and slag alkalinity during the blast furnace process. Huang et al. [16] used a sequence partitioning method combined with a variable structure Bayesian network (VSBN) to model the blast furnace resilience control process. Wu et al. [17] proposed a quantitative causal analysis and optimization framework to solve the problem of inclusions in steel products.

Based on the above analysis, it can be seen that most of the previous studies focus only on the static operation optimization of the blast furnace ironmaking process. However, in practical production, fluctuations in raw material quality often result in the need for adjusting operational parameters 5 to 10 times per hour. Obviously, the actual ironmaking production is a typical dynamic process with black-box characteristics. When adopting traditional static models, the operator has to keep an eye on whether the production environment changes or not, and once the production environment changes, the static operation optimization method has to be implemented, which is very difficult for the operator. Therefore, there are relatively few studies on the dynamic operation optimization of blast furnace ironmaking process. Although existing dynamic multiobjective optimization algorithms can be adopted for this problem, they are too complex to adapt to the intricate industrial environment (the analysis of mainstream dynamic multiobjective algorithms is provided in Section 3.1). Therefore, this paper presents a solution to the dynamic multiobjective operation optimization for the blast furnace ironmaking to optimize the production cost and carbon emissions. Its superiorities are threefold.

- A dynamic operation optimization model for the blast furnace based on data and mechanisms is developed, which differs from traditional static models. The model incorporates time-related parameters and specific conditional variables, enabling real-time adaptation to changes in the production environment.
- A tensor-based dynamic multiobjective optimization algorithm (T-DCMOEA/D) is designed to solve this dynamic model, which effectively considers key parameter relationships in blast furnace operation optimization, improving operational efficiency and stability.
- A dynamic operation optimization framework is constructed, allowing for the dynamic updating of operational variables as production condition changes, ensuring the real-time and flexible operation optimization of the production process.

The rest of the paper is organized as follows. Section 2 describes the methodology for constructing the operation optimization model for blast furnace ironmaking process. Section 3 introduces the proposed T-DCMOEA/D algorithm in detail. The computational analysis of the proposed model and algorithm is performed through a large number of experiments in Section 4. Finally, a summary is given in Section 5.

2. Dynamic multiobjective operation optimization model for blast furnace ironmaking

In practical production of blast furnace, there are three kinds of process parameters: ore conditions (ore grade and type), fuel conditions (quality of coke and pulverized coal injection), and operation parameters. Ore conditions and fuel conditions, which can be measured directly, generally change dynamically over time and are therefore considered as conditional variables. Operation parameters can be dynamically adjusted and optimized according to requirements and are therefore set as decision variables. In the following subsections, we will present the description of variables and then construct the dynamic bi-objective operation optimization model of blast furnace to simultaneously minimize production cost and carbon emissions.

2.1. Conditional variables and identification of environment changes

Iron and steel enterprises usually use multiple types of ore sources in order to reduce raw material costs and control the composition of the iron. The type and grade of ore delivered each time cannot be fixed, which leads to high fluctuations in the raw material of blast furnace ores. Similarly, the quality of the fuel is always fluctuating. Therefore, in the blast furnace ironmaking process, condition variables (ore and fuel conditions) usually change over time and have important implications for the optimization objectives. The conditional variables employed in our model are detailed in Table 1.

In practical production, the identification of changes in the production environment is very important for dynamic operation optimization of blast furnace. If every small change of a conditional variable is regarded as an environmental change, it means that the operation optimization will be too frequent, which will increase computational complexity and lead to blast furnace instability. Therefore, we propose a method for identifying environmental changes in the blast furnace ironmaking process, whose main idea is to significantly reduce the number of environmental changes by merging adjacent similar environments (i.e., with small fluctuations in the conditional variables) in the ironmaking process into a single environment.

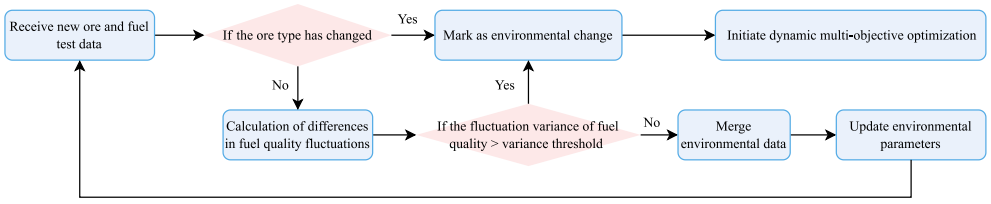


Fig. 2. Environmental identification approach.

Table 1
Conditional variables.

Conditional variables	Symbol	Unit
Hematite ore grade	e_1	%
Lump ore grade	e_2	%
Sinter ore grade	e_3	%
Type A pellet ore grade	e_4	%
Acid sinter ore grade	e_5	%
Type B pellet ore grade	e_6	%
Metallurgical coke grade	e_7	%
Pulverized coal quality	e_8	%
CaO content of hematite ore	e_9	%
CaO content of lump ore	e_{10}	%
CaO content of sinter ore	e_{11}	%
CaO content of type A pellet ore	e_{12}	%
CaO content of acid sinter ore	e_{13}	%
CaO content of type B pellet ore	e_{14}	%
SiO_2 content of hematite ore	e_{15}	%
SiO_2 content of lump ore	e_{16}	%
SiO_2 content of sinter ore	e_{17}	%
SiO_2 content of type A pellet ore	e_{18}	%
SiO_2 content of acid sinter ore	e_{19}	%
SiO_2 content of type B pellet ore	e_{20}	%
Metallurgical coke ash content	e_{21}	%
Pulverized coal ash content	e_{22}	%

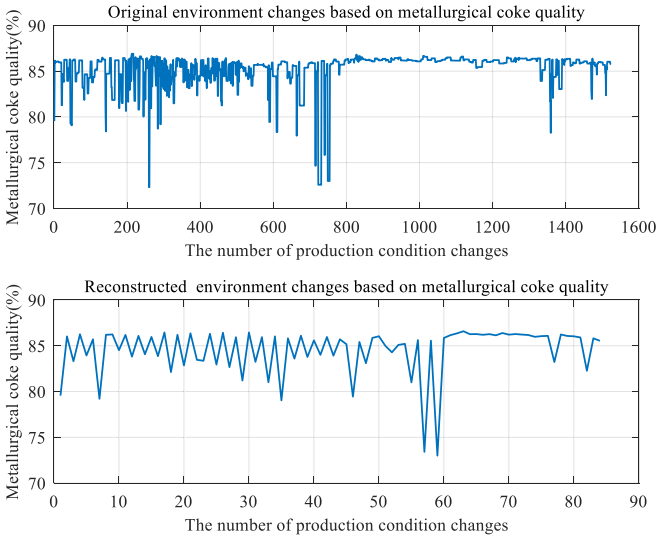


Fig. 3. Comparison of original and reconstructed environment changes based on metallurgical coke quality.

The identification approach is shown in Fig. 2. Whenever new ore and fuel test data are received, the system first checks whether the ore type has changed. If the ore type changes, it is classified as an environmental change. If the ore type remains the same, the system further analyzes the fluctuation in fuel quality. Specifically, the fluctuation difference in fuel quality parameters is calculated according to Eq. (1) (In the formula, e represents the received fuel quality, and $\bar{e}$ represents the average fuel quality under the current environment). If it exceeds the threshold of the average fuel quality under the current environment, it is also classified as an environmental change. Conversely, if there is no significant change, the new data is merged into the current environment, and statistical parameters (such as the average fuel quality under the merged environment) are recalculated based on the merged dataset. Once an environmental change is confirmed, the system immediately initiates the dynamic multiobjective optimization process to generate the optimal control variable combination under the new production environment. Taking metallurgical coke quality as an example, Fig. 3 compares the original environmental change (1525 occurrences) with the reconstructed environmental change (84 occurrences) when the fuel quality fluctuation threshold is set to 0.1. This significantly reduces the optimization frequency while ensuring continuity in the blast furnace production process. The adjusted environmental division strategy allows the system to respond more efficiently to dynamic production conditions. The effectiveness of this strategy will be analyzed in Section 4.3.1.

$$variance = \frac{|e - \bar{e}|}{\bar{e}} \tag{1}$$

Table 2
Operation variables.

Operation variables	Symbol	Unit
Batch weight of hematite ore	x_1	t/h
Batch weight of lump ore	x_2	t/h
Batch weight of sinter ore	x_3	t/h
Batch weight of type A pellet ore	x_4	t/h
Batch weight of acid sinter ore	x_5	t/h
Batch weight of type B pellet ore	x_6	t/h
Batch weight of coke	x_7	t/h
Injection rate	x_8	t/h
Proportion of ore fines with size < 5mm	x_9	%
Proportion of coke with particle size < 10mm	x_{10}	%
Proportion of small coke	x_{11}	%
Phase I compaction and dry quenching coke ratio	x_{12}	%
Phase I compaction and water quenching coke ratio	x_{13}	%
Phase II compaction and dry quenching coke ratio	x_{14}	%
Phase II compaction and water quenching coke ratio	x_{15}	%
Oxygen enrichment content	x_{16}	m^3/h
Hot blast temperature	x_{17}	°C
Air supply volume	x_{18}	m^3/h
Total tuyere area	x_{19}	m^2

2.2. Operation variables

In actual production, there are many operational variables that affect the production cost and carbon emissions of the blast furnace. Some of the important variables that operators need to regulate are detailed in Table 2, and they are the decision variables used in our model to optimize the blast furnace ironmaking process.

Table 3
Constants.

Constants	Symbol	Unit
Hematite ore price	c_1	¥/t
Lump ore price	c_2	¥/t
Sinter ore price	c_3	¥/t
Type A pellet ore price	c_4	¥/t
Acid sinter ore price	c_5	¥/t
Type B pellet ore price	c_6	¥/t
Metallurgical coke price	c_7	¥/t
Pulverized coal price	c_8	¥/t

Table 4
Intermediate observed variables.

Intermediate observed variables	Symbol	Unit
Fe element recovery rate	$p_1(x,e)$	%
Fe element content in molten iron	$p_2(x,e)$	%
Si element content in molten iron	$p_3(x,e)$	%
CaO Recovery Rate	$p_4(x,e)$	%
SiO_2 Recovery Rate	$p_5(x,e)$	%
C element content in molten iron	$p_6(x,e)$	%

2.3. Constants and intermediate observed variables

The constants used in our model are mainly ore and fuel prices, which fluctuate very little over the course of a few days and can therefore be considered as constants over the time horizon of the dynamic run optimization. The detailed descriptions of these constants are given in Table 3. In this article, prices are expressed in Chinese Yuan (¥).

To calculate the objective functions and construct process constraints, we also need some important intermediate observed variables, whose descriptions are shown in Table 4. Since these variables cannot be precisely determined by the model, they are obtained from the random forest [18] prediction models based on all the decision variables (x_1, x_2, …, x_{19}) and conditional variables (e_1, e_2, …, e_{22}).

2.4. Objective functions

As described in the Introduction section, there are two objectives in our model, i.e., production cost and carbon emissions per ton of molten iron in each hour. In theory, a reduction in ironmaking costs should make production more economical, leading to an increase in iron output. However, the reality is more complex. Production capacity and market demand can limit iron output. A decrease in iron production implies a lower output per unit time, which may indirectly result in reduced energy efficiency and higher allocation of fixed costs, among other issues. These factors can make production less economical, potentially leading to an increase in carbon emissions per ton of iron. According to the statistics in Ref. Liu et al. [19] and the data we have collected, a reduction in ironmaking costs is more likely to result in higher carbon emissions per ton of iron. Consequently, these two objectives are often in conflict. The two objectives are generally conflicting with each other, as a reduction in the cost of ironmaking leads to a significant reduction in iron production, which in turn leads to an increase in carbon emissions per ton of iron.

Since the predominant costs in blast furnace ironmaking are associated with the purchase of fuel and ore, the first objective function, i.e., production cost per hour, can be calculated as the costs for ore and fuel consumption per hour, as shown in Eq. (2).

$$f_1 = iron_{cost} = \sum_{i=1}^{8} c_i x_i \tag{2}$$

With respect to carbon emissions, there are numerous influence factors such as process energy consumption, fuel composition, and resource efficiency. Without considering the recycling of carbon within the process, the amount of carbon dioxide emissions required to produce one ton of pig iron can be calculated using Eqs. (3) and (4).

$$f_2 = CO_{2_{emission}} = \frac{\left(\sum_{i=7}^{8} e_i x_i - iron_{output} \times p_6(x,e)\right) \times \frac{44}{12}}{iron_{output}} \tag{3}$$

$$iron_{output} = \sum_{i=1}^{6} e_i x_i \times \frac{p_1(x,e)}{p_2(x,e)} \tag{4}$$

where $iron_{output}$ is the molten iron output per hour.

The production cost objective f_1 focuses on fuel and ore expenses because these constitute over 80% of the total operational costs in blast furnace ironmaking, as reported in industry benchmarks [19]. Maintenance and labor costs are excluded due to their relatively stable nature over short-term optimization horizons and their minor contribution to dynamic decision-making. Similarly, indirect emissions (e.g., from energy production) are omitted because they are governed by external energy supply systems and are not directly controllable through blast furnace parameter adjustments. This simplification aligns with prior studies [7] that prioritize dominant and actionable factors for real-time optimization.

2.5. Constraints

In the blast furnace ironmaking process, the content of certain crucial elements in hot metal, as well as the composition of slag, are important influence factors for the quality of hot metal and the stable operation of blast furnace. Therefore, this paper primarily focuses on two process constraints: the silicon content in hot metal and the slag basicity.

The silicon content in the molten iron is directly proportional to the temperature of the molten iron, which can reflect the operating condition of the blast furnace. In addition, silicon element can enhance the melting point of pig iron, and improve the fluidity of molten iron and its casting capabilities. Insufficient silicon content usually results in brittle pig iron, which is unfavorable for subsequent processing. Timely adjustments of silicon content can be made to avoid production fluctuations in the blast furnace and guarantee the quality of molten iron [20]. Therefore, it is essential to maintain the silicon content (denoted as p_3) within a specific range in molten iron, which can be expressed as follows.

$$S_{i-min} < p_3(x,e) < S_{i-max} \tag{5}$$

where the silicon content $p_3(x,e)$ is obtained from the random forest prediction model based on all decision variables and conditional variables, and S_{i-min} and S_{i-max} are the lower and upper bounds for silicon content.

Slag basicity is also a very important indicator in the ironmaking process, and is expressed as the ratio of basic oxide content to acidic oxide content in the blast furnace slag [21]. It directly affects the flow properties of the slag and the desulfurization process of pig iron. In this study, we calculate the binary basicity R, whose value exceeding 1 indicates the basic furnace slag, and it is crucial to maintain it within a specific range to ensure the stable operation of the blast furnace. The calculation for R is defined as Eq. (6):

$$R = \frac{\omega_{CaO}}{\omega_{SiO_2}} \tag{6}$$

where ω_{CaO} and ω_{SiO_2} denote the content of CaO and the SiO_2 content in the slag, respectively. CaO in the slag mainly comes from the ash in the ore and blown coal, while SiO_2 in the slag is mainly derived from ash in ore, metallurgical coke and pulverized coal. Their calculation formulas can be defined in Eqs. (7) and (8), respectively.

$$\omega_{CaO} = p_4(x,e) \cdot \left(e_9 x_1 + e_{10} x_2 + e_{11} x_3 + e_{12} x_4 + e_{13} x_5 + e_{14} x_6\right) \tag{7}$$

$$\omega_{SiO_2} = p_5(x,e) \cdot \left(e_{15} x_1 + e_{16} x_2 + e_{17} x_3 + e_{18} x_4 + e_{19} x_5 + e_{20} x_6\right) \tag{8}$$

Based on the above definitions, the constraint for the Slag basicity can be defined as follows.

$$R_{min} < R(x,e) < R_{max} \tag{9}$$

where R_{min} and R_{max} represent the lower and upper bounds for the binary basicity R.

3. Strategies for solving optimization models for blast furnace operation

3.1. Related work

In recent years, owing to their exceptional performance, evolutionary algorithms have been extensively applied to real-world optimization problems [5,22]. The development of a blast furnace operation optimization model in this paper addresses a dynamic multiobjective optimization problem, typically solvable using dynamic multiobjective evolutionary algorithms. While existing dynamic algorithms have demonstrated strong performance on benchmark problems, they encounter certain limitations when applied to the model presented in this paper. Classic dynamic multiobjective optimization evolutionary algorithms, such as DNSGA-II [23], directly extend static algorithms into dynamic scenarios, making it difficult to effectively handle irregular environmental changes, particularly in industrial processes. In recent years, dynamic multiobjective optimization evolutionary algorithms based on prediction strategies have made significant progress, such as the DM-DMOEA algorithm proposed by Wang et al. [24], which is based on Diffusion Model prediction, and the DMOA-MHKT algorithm proposed by Song et al. [25], which relies on Multisource and Hidden Source-Based Knowledge Transfer. While these prediction strategies improve the performance of the algorithms, they also increase computational complexity, posing a challenge for complex industrial scenario models. Firstly, the environmental changes in the model have no clear regularity, which increases the complexity of solving the model. Secondly, some dynamic algorithms based on transfer learning or machine learning exhibit excessively long runtimes [26], posing a hindrance to their practical applicability in problem-solving. Thirdly, this paper introduces simple dynamic constraint conditions, requiring algorithmic adjustments to be made promptly. Fourthly, real-world problems typically lack the scale standardization observed in benchmark test problems, which necessitates timely adjustments of scales by dynamic algorithms to enhance their evolutionary and evaluative performance.

In response to the challenges prevalent in traditional dynamic multiobjective evolutionary algorithms, this study proposes a Tensor-based dynamic multiobjective evolutionary algorithm (T-DCMOEA/D) based on the unique characteristics of the model constructed. This algorithm operates with the decomposition framework of MOEA/D [27] and employs tensors to represent the dynamic variations of operation variables under different environmental conditions. Using tensor decomposition and ARIMA, it achieves predictive generation of initial operation variables in new environments, thus enhancing the quality of these initial operation variables and accelerating the algorithm's convergence rate in new environments.

3.2. Main framework of solving for dynamic blast furnace operation optimization

The strategic framework for the T-DCMOEA/D algorithm in solving the blast furnace operation optimization problem is illustrated in Fig. 4. This framework encompasses three primary modules: the CMOEA/D static evolutionary module, the Decomposition-based spatio-temporal tensor prediction module, and the Modeling Blast Furnace Ironmaking module. The Modeling Blast Furnace Ironmaking module serves as the model requiring optimization. Initial environmental parameters are fed into this framework, where the CMOEA/D module processes and outputs the optimal operating variables for the current environment. When an environmental change is detected, the prediction module forecasts the initial operating variables for the current environment and inputs them into the CMOEA/D module to repeat the previous step. If there is no change in the environment, the current operating variables are maintained, awaiting any environmental changes.

The pseudocode for the T-DCMOEA/D algorithm is presented in Algorithm 1. Initially, random initial operating variables POP_0, are generated based on the initialization parameters (including the number of sets of operating variables N, the number of operating variables per set V, the blast furnace operation optimization problem $F(x,t)$, constraints C, weight vectors W, and the environmental parameter $T = 0$). The CMOEA/D algorithm with normalization and constraints is used to find the set of operational variables POP_1 that is closest to the optimal operational variables in the first-generation environment (Algorithm 1, line 2). The specific methods for normalization and constraints will be elaborated on in Sections 3.3 and 3.4. When a change in the environment is detected, the environmental parameter T is updated (Algorithm 1, line 5). A tensor-based approach is employed to generate initial operating parameters in the new environment (Algorithm 1, line 6). This method incorporates a decomposition-based prediction concept, as outlined in Fig. 5. Operating variables for each sub-problem are predicted along their respective weight vectors, and then the predicted for each sub-problem are merged to form the initial operating variables in the new environment. Based on the predicted operational variables, the CMOEA/D algorithm processes and outputs the near-optimal set of operational variables for the current environment. If the environment remains unchanged, the current population of operational variables is maintained.

Algorithm 1: Solving for dynamic blast furnace operation optimization

1 Randomly initialize the operation variables POP_0 and the parameters.;
2 $POS_1 = \text{CMOEA/D}\left(POP_0, F(x,0), N, V, C\right)$;
3 **while** *termination criterion is not reached* **do**
4 **if** *environment change* **then**
5 $T = T + 1$;
6 POP_T =T-Prediction(POS_{all});
7 $POS_T = \text{CMOEA/D}\left(POP_T, F(x,T), N, V, C\right)$;
8 $POS_{all} = POS_{all} \cup POS_T$;
  Output: POS_T
9 **else**
10 Keep the current population POS_T;

3.3. The normalization of dynamic optimization algorithms

In the same context, substantial disparities in the scales of solving target carbon emissions and costs can lead to an imbalance in target weights. Target normalization proves to be a valuable technique for enhancing the consistency of solutions, particularly when the targets lack precise scaling [27]. When selecting based on weight vectors, the replacement method for normalizing the target variable f_i is:

$$\bar{f}_i = \frac{f_i - z_i^{min}}{z_i^{max} - z_i^{min}} \tag{10}$$

Where, z_i^{min} functions as the reference point and signifies the minimum value among all current operation variables, i.e., $z_i^* = min\left\{f_i(x) | x \in PS\right\}$. z_i^{nad} represents the nadir point in the objective space, corresponding to the maximum value among all current operation variables, i.e., $z_i^{nad} = max\left\{f_i(x) | x \in PS\right\}$. This method serves to normalize the range of each objective within PFs to the $[0,1]$ range. Consequently, in the algorithm, the scalar optimization problem of the Chebyshev method can be replaced by:

$$minimize\ g^{te}\left(x|W, z_i^{min}, z_i^{max}\right) = \max_{1<i<m}\left\{W_i|\bar{f}_i|\right\} \tag{11}$$

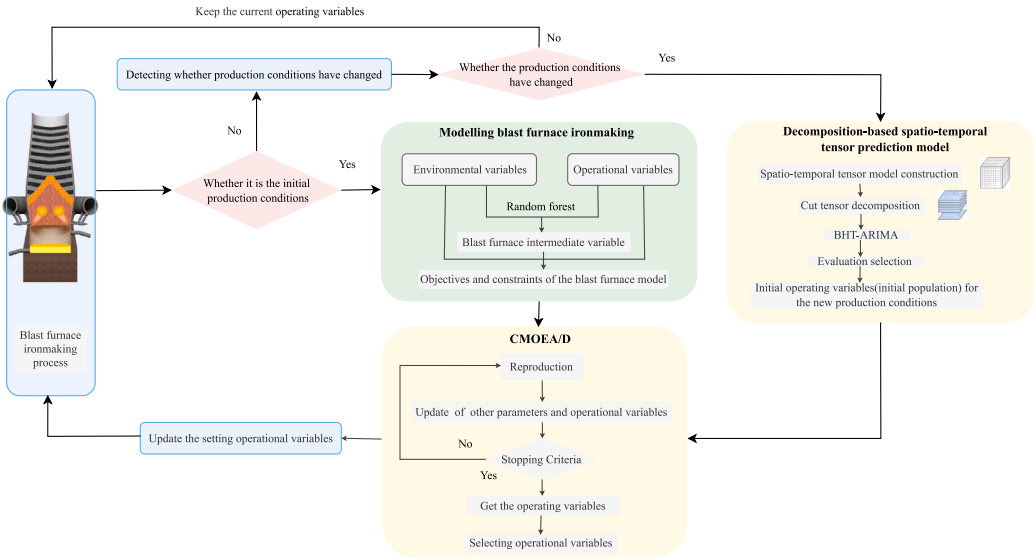


Fig. 4. Flowchart of solving for dynamic blast furnace operation optimization.

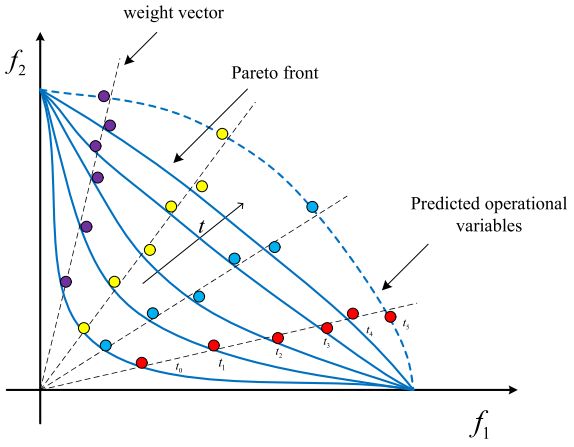


Fig. 5. Decomposition-based prediction schematic.

Where W_i is the weight vector in the MOEA/D algorithm. The selection strategy of the initial operational variables generator is the same as the substitution in MOEA/D.

3.4. Operational variables selection with constraints

The blast furnace operational optimization model formulated in this paper is a dynamic multiobjective optimization problem featuring dynamic constraints. The inclusion of these supplementary constraints not only decelerates the convergence rate for solving the model but also amplifies the intricacy of the solution process. To ensure compliance with these constraints, the selection of operational variables is guided by the method described by Himanshu Jain and his colleagues in Ref. Jain and Deb [28].

In blast furnace operation optimization problems, the feasibility of the operating variables is crucial. Therefore, the constraint violation degree of the objective needs to be calculated to judge whether the operating variables are feasible or not. The calculation of constraint violation degree for the model proposed in this paper is as follows:

$$C = C_{Si} + C_R \tag{12}$$

$$C_{Si} = max\left\{p_3(x,e) - Si_{max,0}\right\} + max\left\{Si_{min} - p_3(x,e), 0\right\} \tag{13}$$

$$C_R = max\left\{R(x,e) - R_{max}, 0\right\} + max\left\{R_{min} - R(x,e), 0\right\} \tag{14}$$

Where, C_{Si} denotes the constraint violation degree for silicon content in molten iron, while C_R signifies the constraint violation degree related to slag basicity. An operation variable is considered feasible when the total constraint violation, designated as C, equals zero. In the event of a comparison between a feasible and an infeasible operation variable, the feasible variable is favored. When two feasible operation variables are in competition, the one with the lower objective value prevails. When comparing two infeasible operation variables, the one that approaches feasibility more closely (i.e., exhibits a lower constraint violation degree) takes precedence.

3.5. Tensor-based initial operating variable generation method

Firstly, we employ a three-dimensional spatiotemporal tensor $\chi \in R^{N\times V\times T}$ to depict the distribution of operational variables across all environments (Algorithm 2, Line 1). Here, N represents the number of sets of operational variables generated in each environment, V denotes the count of variables within each set, and T represents the number of changes in environmental conditions, increasing with environmental shifts. For this third-order spacetime tensor, two directions of slicing are used in this paper. As illustrated in Fig. 6, the slice obtained by cutting along the sets of operational variables dimension is denoted as $\chi_{n::}$, representing the information of the nth set of operational variables across all environments. Cutting along the environmental change number dimension results in $\chi_{::t}$, representing the distribution of all operational variables under the tth environmental condition. Similarly, each slice, when cut along different dimensions, can yield two directional fibers. For example, $\chi_{n:t}$ represents the values of the nth set of blast furnace operational variables under the tth environmental condition, while $\chi_{nv:}$ represents the time series of the vth operational variable within the nth set, changing with varying environmental conditions. These slices can be partitioned as needed based on algorithmic requirements.

Secondly, perform the completion operation on the operational variable tensor $\mathcal{X}$ (Algorithm 2, line 2). Ensure that the time series specifications within the tensor $\mathcal{X}$ meet the minimum training specification, Trainlow. Specifically, accomplish this by repetitively rearranging

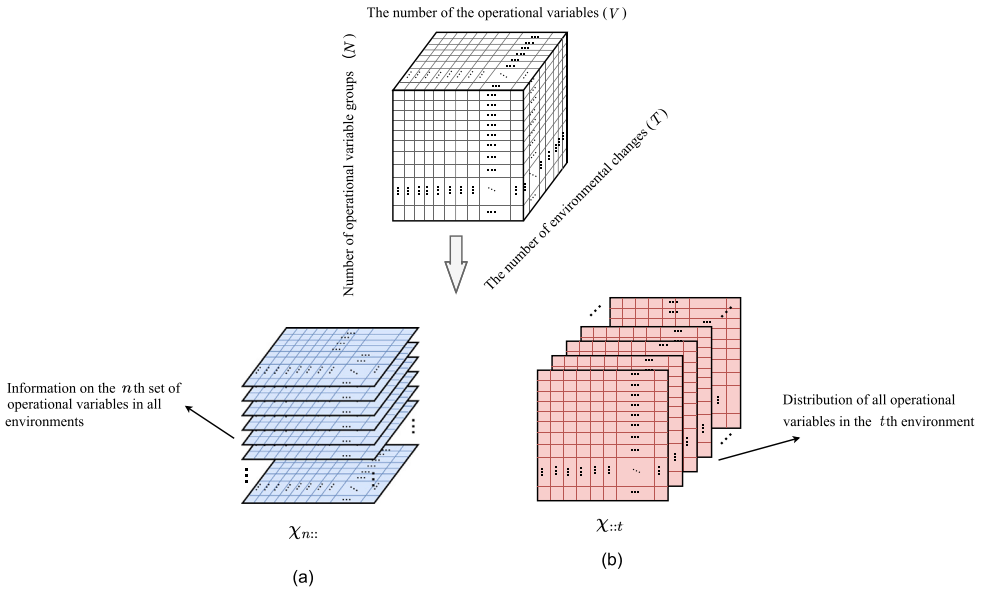


Fig. 6. Schematic for operational variables tensor and its slices.

the existing slices $\mathcal{X}_{::t}$ in the order of environmental changes, such that the resulting newly completed operational variable tensor has a dimension T, representing the number of environmental change occurrences, greater than or equal to Trainlow.

Thirdly, the prediction of operational variables is approached through a decomposition-based methodology. This methodology, grounded in the principles of decomposition, serves to eliminate disparities among different sets of operational variables, facilitating a more insightful examination of how each set of operational variables responds to environmental variations. Comparative experimentation has empirically demonstrated the superior predictive performance of this decomposition-inspired methodology in contrast to holistic predictions. Building on the decomposition concept from MOEA/D, in the third step of Algorithm 2, we break down the problem into operational variables several subproblems, following the approach illustrated in Fig. 6(a). By segmenting the tensor along the number of operational variable groups, we obtain several time series slices, each containing individual sets of operational variables denoted as $\mathcal{X}_{n::}$. Subsequently, each slice $\mathcal{X}_{n::}$ is subjected to sequential predictions using the BHT-ARIMA model [29], resulting in the generation of slices representing the predicted operational variables for the new environmental context, denoted as $\mathcal{X}_{n:T+1}$ (as indicated in Algorithm 2, line 4). Furthermore, boundary checks are conducted for the predicted operational variables in each new environmental context (as outlined in Algorithm 2, line 5). Given the partial similarity of solutions in adjacent environments in dynamic problems, we employ a selection strategy to choose dominating operational variables from the preceding environment to replace the predicted operational variables (as detailed in Algorithm 2, line 6).

Finally, the predicted operational variables for each subproblem are combined to become the initial operational variables in the new environment. The pseudo-code of the tensor decomposition-based algorithm for predicting initial operating variables is shown in Algorithm 2.

Algorithm 2: T-Prediction

Input: Operating variables in all environments: POS_{all};
minimum training size: $Train_{low}$;
Output: Initial operating variables for new environments: POS_T;

1 $\mathcal{X} \in R^{N\times V\times T}$ =Tensor(POS_{all}) ;
2 $\mathcal{X}^* = \text{Complementation}\left(\mathcal{X}, \text{Train}_{\text{low}}\right)$;
3 **for** $\mathcal{X}_{n::}$ *in* $\mathcal{X}^*$ **do**
4 $\quad \mathcal{X}_{n:T+1}$ =BHT-ARIMA($\mathcal{X}_{n::}$);
5 $\quad \mathcal{X}_{n:T+1}$ =BoundaryCheck($\mathcal{X}_{n:T+1}$);
6 $\quad \mathcal{X}_{n:T+1}$ =WeightSelection($\mathcal{X}_{n:T+1}, \mathcal{X}_{n:T}$);
7 $POP_T = \{\mathcal{X}_{1:T+1}, \mathcal{X}_{2:T+1}, ..., \mathcal{X}_{N:T+1}\}$;
8 **return** POP_T;

3.6. CMOEA/D algorithm

The T-DCMOEA/D algorithm primarily relies on the evolution driven by CMOEA/D. CMOEA/D is a variant of MOEA/D that incorporates constraint handling. Its core concept revolves around a decomposition strategy, which transforms multiobjective optimization problems into a series of single-objective sub-problems for resolution. The decomposition approach of the MOEA/D algorithm is particularly advantageous in addressing high-dimensional multiobjective optimization problems, and it yields a balanced solution set that covers the PF.

The pseudocode for CMOEA/D, presented as Algorithm 3, follows a specific sequence. In the first step, described in the line 2 of Algorithm 3, the reproductive operation is executed on the operation variables within the current environment. This operation involves the random selection of two operation variables from the neighborhood of a single operation variable and the use of genetic operators to generate new operation variables from this pair. The second step, represented by the line 3 of Algorithm 3, comprises a repair process. This paper applies boundary constraints to the operation variables generated in the first step to ensure that all operation variables remain within their specified ranges. The third step involves the continuous updating of the ideal point to adapt to changes in operation variables. Finally, in the fourth step, as specified in the line 6 of Algorithm 3, the operation variables are updated based on neighborhood information. Each subproblem's operation variable is assessed against its neighbors, and if a neighbor outperforms the subproblem's operation variable, it replaces the current operation variable, following the domination relationship described in Section 3.4. The result is the updated operation variables within the current environmental context. For a more detailed explanation of these strategies, please refer to Ref. Zhang and Li [27].

3.7. Computational complexity

The tensor prediction model primarily consists of MDT, Tucker decomposition, and ARIMA prediction. Their computational complexities

Algorithm 3: CMOEA/D

Input: Operational variables in the current environment: POP_T; blast furnace operation optimization model: $F(x,t)$; number of subproblems considered in the algorithm: N; constraints on optimization models for blast furnace operations: C; number of weight vectors in the neighborhood of each weight vector: T; ideal point in the current environment: z;

Output: Updated operational variables in the current environment: POP_T;

while *termination criterion is not reached* **do**
- POP_T^* =Reproduction(POP_T, T) ;
- POP_T' =Improvement($POP_T^*, F(x,T)$) ;
- z =Update(z);
- POP_T =Update(POP_T', T, C);

return POP_T;

are O(VT^3), O(NVn^2), and $O((p+q)T)$, respectively. Here, V is the number of operational variables, T is the number of environmental changes, N is the number of subproblems, n=2 is the core tensor size, and the ARIMA prediction has parameters (p=1, d=2, q=3). Boundary checking and selection add an additional complexity of O(NV). Since tensor prediction is applied to all N subproblems, the total complexity of T-Prediction is O($NV(T^3+n^2+T)$). The complexity of the CMOEA/D framework is O(mN^2B), where N is the number of subproblems, B is the neighborhood size, and m is the number of objectives.

The dominant term in the computational complexity of the T-DCMOEA/D algorithm comes from T-Prediction, especially the MDT operation. In comparison to the T-Prediction module, the computational complexity of the CMOEA/D module can be considered negligible. Therefore, the overall computational complexity of T-DCMOEA/D is O(NVT^3).

4. Experimental results and discussion

In order to ascertain the superior problem-solving capabilities of T-DCMOEA/D in the context of blast furnace operation optimization, this section conducts an extensive array of numerical experiments. Section 4.2 focuses on evaluating the algorithm's performance using benchmark test problems that closely mimic real-world issues. In Section 4.3, these algorithms are applied to tackle practical problems. Finally, a comparison is made between the actual problem values and the results achieved through algorithmic optimization.

All experiments presented in this paper were executed on a personal computer featuring an Intel Core i7-12700 CPU. Uniform performance metrics and comparative algorithms were utilized. Optimal values within the experimental outcomes are denoted in deep gray. The statistical comparison between the algorithm introduced in this paper and other competing algorithms was conducted through the Wilcoxon rank-sum test [30] at a significance level of 0.05. The test results are symbolized as "+", "=", and "-", indicating statistical superiority, no significant difference, and inferiority of the proposed algorithm compared to competing algorithms, respectively.

4.1. Performance metrics and compared algorithms

4.1.1. Performance indicators

Performance metrics chosen for this study include MIGD [31] and MHV [32]. MIGD is an enhancement of IGD, designed to comprehensively reflect the convergence and diversity of solutions in dynamic environments. Smaller MIGD values indicate better algorithm performance. The calculation of MIGD is as follows:

$$MIGD = \frac{\sum_{t\in T} IGD\left(PF_t^*, PF_t\right)}{|T|} \tag{15}$$

$$IGD_t\left(PF_t^*, POF_t\right) = \frac{\sum_{p\in PF_t^*} d\left(p, PF_t\right)}{|PF_t^*|} \tag{16}$$

Where T represents a set of discrete time points, and $|T|$ signifies the number of environmental changes occurring during runtime. PF_t^* constitutes a series of sampled points from the actual PFs at time window t, while PF_t represents the algorithm's estimated values at the same time window. The term $d(p, PF_t)$ denotes the minimum Euclidean distance from point p in the actual PFs to the algorithm's estimated value at PF_t^*. $|PF_t^*|$ serves as the cardinality of the actual PF.

MHV, improved through Hypervolume (HV), quantifies the diversity of solutions by calculating the hypervolume of the solution dominance region. A higher MHV value indicates a broader distribution of solutions. The calculation of MHV is as follows:

$$\text{MHV} = \frac{\sum_{t\in T} HV_t}{|T|} \tag{17}$$

$$HV_t = \text{HV}(PF_t, z) \tag{18}$$

Where z is a reference point dominated by all PS.

4.1.2. Comparison algorithm

In this paper, the improved dynamic algorithms IGP-DCMOEA/D [33], ISVM-DCMOEA/D [34], KGB-DCMOEA/D [35], and HRS-DCMOEA/D [36] are used as comparison algorithms. These improved algorithms are better suited to dynamic constrained multiobjective optimization problems.

IGP-DCMOEA/D employs an inverse Gaussian process (IGP) to construct predictors, mapping solutions from the objective space to the decision space. Its advantages lie in rapid convergence and shorter model computation time. ISVM-DCMOEA/D constructs an incremental support vector machine model to mine and record solution features for prediction. This method is more effective in enhancing solution diversity but requires more time to identify feature solutions. KGB-DCMOEA/D uses knowledge-guided Bayesian classification to predict the initial population in new environments, effectively utilizing information from all historical environments to achieve more robust predictions. HRS-DCMOEA/D designs a hierarchical response system to predict the initial population in new environments. This system responds to environmental changes by integrating mainstream ideas for dynamic behavior handling and combining the advantages of various approaches.

4.2. Performance analysis on benchmark test

4.2.1. Experiment setting

The benchmarking dataset for this paper comes from the latest dynamic-constrained multiobjective optimization test set [37]. The test set was developed with 10 benchmark test functions covering a wide range of features. They are classified into three categories based on the dynamically changing features:

I. No change in objectives, change in constraints;
II. Objectives change, constraints remain the same;
III. Both objectives and constraints change

Due to the fact that the optimization problem for blast furnace ironmaking operations constructed in this paper represents a continuous dynamic constrained multiobjective optimization problem with both changing objectives and constraints, we chose to conduct tests using the continuous benchmark problems (DCF3 and DCF8) from category 3.

The benchmark problems selected for the examination are dual-objective problems, with a fixed population size of 100 individuals and 10 decision variables. The environmental variable is expressed as $t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$, where τ signifies the generation number. We have configured four diverse combinations of τ_t (alteration frequency) and

Table 5
Mean and variance of MIGD for benchmarking problems.

Prob.	(τ_t, n_t)	IGP-DCMOEA/D	ISVM-DCMOEA/D	KGB-DCMOEA/D	HRS-DCMOEA/D	T-DCMOEA/D
DCF3	(10,5)	3.55e−02(4.32e−03)+	1.14e−01(3.56e−02)+	3.59e−02(5.88e−03)+	3.86e−02(3.37e−03)+	1.86e−02(5.56e−03)
	(5,10)	5.68e−02(6.09e093)+	2.81e−01(5.54e−02)+	1.06e−01(8.51e−03)+	1.24e−01(2.27e−02)+	3.34e−02(7.48e−03)
	(10,10)	2.22e−02(1.58e−03)+	1.18e−01(3.96e−02)+	2.81e−02(5.02e−03)+	3.18e−02(3.89e−03)+	1.47e−02(1.78e−03)
	(20,10)	1.02e−02(9.71e−03)+	3.64e−02(8.52e−03)+	1.07e−02(1.28e−03)+	1.06e−02(1.37e−03)+	8.24e−03(6.22e−04)
DCF8	(10,5)	3.14e−02(4.30e−03)+	3.39e−02(4.98e−03)+	1.27e−01(4.28e−02)+	3.04e−02(4.36e−03)+	1.93e−02(2.24e−03)
	(5,10)	4.95e−02(4.16e−02)+	9.56e−02(2.38e−02)+	1.43e−01(7.93e−02)+	8.47e−02(1.70e−02)+	3.22e−02(4.51e−03)
	(10,10)	2.56e−02(2.63e−03)+	3.51e−02(5.57e−03)+	5.75e−02(6.01e−02)+	2.28e−02(3.00e−03)+	1.49e−02(2.55e−03)
	(20,10)	1.51e−02(1.57e−03)+	1.44e−02(2.46e−03)+	2.06e−02(1.95e−02)+	1.48e−02(1.72e−03)+	1.13e−02(1.69e−03)

Table 6
Mean and variance of MHV for benchmarking problems.

Prob.	(τ_t, n_t)	IGP-DCMOEA/D	ISVM-DCMOEA/D	KGB-DCMOEA/D	HRS-DCMOEA/D	T-DCMOEA/D
DCF3	(10,5)	3.71e−01(4.30e−03)+	3.76e−01(6.86e−03)+	3.71e−01(4.63e−03)+	3.73e−01(2.75e−03)+	3.88e−01(8.20e−04)
	(5,10)	3.35e−01(6.66e−03)+	3.49e−01(1.42e−02)=	2.85e−01(7.14e−03)+	2.65e−01(1.05e−02)+	3.53e−01(5.10e−03)
	(10,10)	3.90e−01(9.87e−04)=	3.75e−01(7.66e−03)+	3.81e−01(4.28e−03)+	3.80e−01(4.08e−03)+	3.99e−01(1.23e−03)
	(20,10)	4.09e−01(4.76e−04)+	4.06e−01(6.62e−03)+	4.10e−01(3.20e−04)=	4.09e−01(3.99e−04)=	4.12e−01(6.48e−04)
DCF8	(10,5)	4.57e−01(1.46e−03)+	4.58e−01(5.95e−03)+	4.15e−01(9.70e−03)+	4.62e−01(1.68e−03)=	4.69e−01(4.64e−03)
	(5,10)	4.14e−01(4.18e−03)+	3.32e−01(8.91e−03)+	3.59e−01(1.96e−02)=	4.00e−01(9.83e−03)+	4.49e−01(2.54e−03)
	(10,10)	4.59e−01(6.52e−04)+	4.48e−01(4.45e−03)+	4.46e−01(1.95e−02)+	4.64e−01(1.92e−03)=	4.71e−01(6.72e−04)
	(20,10)	4.75e−01(5.89e−04)+	4.78e−01(3.96e−04)+	4.76e−01(6.53e−03)+	4.79e−01(2.25e−04)=	4.83e−01(3.33e−04)

n_t (alteration amplitude). Termination of the algorithm occurs after 30 environmental changes, and 100 generations transpire prior to the inaugural environmental shift. Each algorithm was executed independently 20 times for every problem, with calculations of mean and variance conducted on their evaluation metrics. In order to accentuate the clarity of experimental comparisons, the frequency of individual replacement in the selection strategy has been deliberately reduced, thus retarding the convergence pace. This replacement frequency was stipulated at 5.

In the section related to tensor-based population initialization, the critical threshold for the tensor training model denoted as $Train_{low}$, is set to 8. Within the context of BHT-ARIMA, the parameter for multi-way delay embedding transform(MDT) [38] decomposition is configured as 2, while in the Tucker decomposition method, the size of the core tensor is specified as 3×2. The parameters for ARIMA, namely q, d, and p, are individually established as 3, 2, and 1, respectively.

The reference point z in MHV is set at (1.1, 1.1). A set of 300 uniformly sampled points is drawn from the true PFs in each scenario to calculate the values of MIGD and MHV.

4.2.2. Experimental results and analysis

Table 5 and Table 6 display the computed performance indicators for the five algorithms. Fig. 6 portrays the evolutionary progress of MIGD, whereas Fig. 7 visualizes the spatial distribution of solutions for DCF3 and DCF8. The outcomes demonstrate the consistent superiority of the T-DCMOEA/D algorithm in all four environmental configurations for two test problems. In most instances, its performance stands out significantly. A closer examination suggests that this superiority can be attributed to the tensor-based method for initializing the population, which enhances its convergence, as well as the uniform distribution of the population based on the decomposition concept.

Both DCF3 and DCF8 present continuous problems with dynamic changes in both objective functions and constraints. DCF3 exhibits a relatively straightforward dynamic behavior, whereas DCF8's PF consistently aligns with the boundary of the feasible domain and oscillates over time. The constraints in both benchmark problems exert considerable pressure on algorithm convergence. Experimental findings reveal that increasing the neighborhood size leads to a faster convergence rate. However, to maintain a wide distribution of solutions and ensure clear comparisons, this paper intentionally reduces the neighborhood size, deliberately slowing down the convergence. Under this parameter configuration, the superiority of T-DCMOEA/D becomes more conspicuously apparent.

4.3. Performance analysis on practical industrial data test

4.3.1. Data preprocessing

The experimental data in this section were obtained from the actual production process of a blast furnace in a large steel enterprise in China. Data collection began in 2020 and continued through 2022, with data being collected hourly. The data collection system relies on a multi-sensor network, infrared and thermal imaging systems, with data being aggregated into the central control system via PLC and Ethernet. The samples are arranged hourly, and those with missing values or collected during periods of unstable furnace operation are discarded. A total of 1525 valid samples were collected. Each sample contains 48 data items, including 2 target variables, 19 operating variables, 21 condition variables, and 6 intermediate variables.

The range of permissible values for operational variables and conditional variables is specified as $(\mu - 3\sigma, \mu + 3\sigma)$. In this context, μ denotes the mean value of the project's parameters, while σ represents the standard deviation of these parameters. In cases where $\mu - 3\sigma$ is less than 0, it is reset to 0. The range necessary for constraint violation calculations also extends to the silicon content in the molten iron and the alkalinity of the slag, within the same confines of $(\mu - 3\sigma, \mu + 3\sigma)$.

We applied the environmental recognition method introduced in Section 2.1 to reduce the original 1525 samples to 84. To evaluate this process, we selected three assessment metrics: Variance Retention Rate (VRR) [39], Critical Point Coverage (CPC) [40], and the Kolmogorov–Smirnov test (KS) [41], to analyze four fuel quality indicators. Specifically, if the VRR is greater than 90%, it indicates that the reduced data sufficiently retains the original fluctuations; if the CPC exceeds 90%, it shows that the critical points are adequately covered; and if the p-value from the KS test is greater than 0.05, it suggests that the data distributions are consistent. The results, as shown in Table 7, demonstrate that all VRR values are close to or exceed 90%, indicating that the data reduction process effectively preserves the fluctuation characteristics of the original environment. All CPC values exceed 90%, indicating that the reduced data covers the extreme conditions of the original environment. Moreover, all p-values from the KS tests are greater than 0.05, further confirming the consistency of the data distributions between the original and reduced environments. These results suggest that the environmental recognition method we proposed exhibits excellent performance.

For the purpose of constructing three distinct practical scenarios, the dataset is partitioned based on temporal divisions into Instance1, Instance2, and Instance3, corresponding to the operational states of

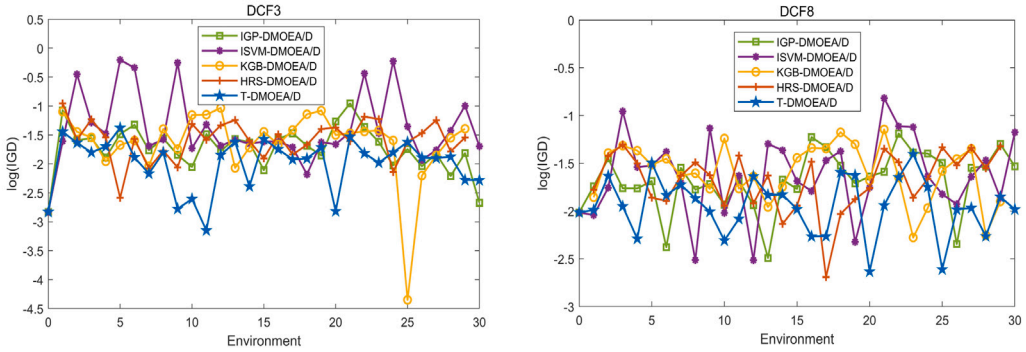


Fig. 7. Evolution of MIGD values for benchmark test problems with $\tau_t = 10$ and $n_t 10$.

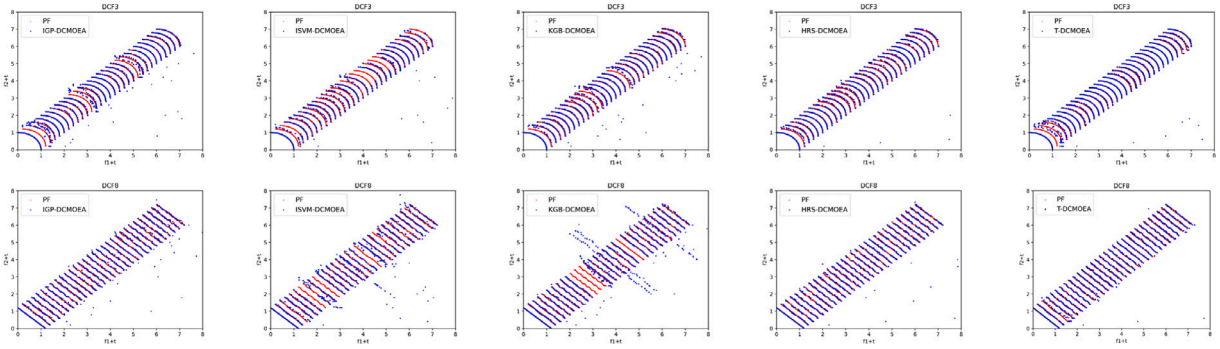


Fig. 8. Distribution of solutions to the benchmark test problems for $\tau_t = 10$ and $n_t 10$.

Table 7
Environmental data evaluation.

Conditional variables	Symbol	VRR	CPC	p-value
Metallurgical coke grade	e_7	88.0%	90.7%	0.214
Pulverized coal quality	e_8	92.5%	91.7%	0.301
Metallurgical coke ash content	e_{21}	89.3%	90.3%	0.118
Pulverized coal ash content	e_{22}	90.4%	90.2%	0.185

the blast furnace in the years 2020, 2021, and 2022. Due to practical considerations, 2020 witnessed frequent fluctuations in fuel quality. In 2021, both fuel and ore displayed variations. In 2022, the types of ore underwent frequent changes. These ore fluctuations are visually presented in Fig. 8, where concavities represent the absence of specific ore types, and convexities signify their presence. When certain ore types are absent, the corresponding operational variables are fixed at 0, leading to a shift in the number of operational variables and an added layer of complexity in problem-solving. The primary fuel variations predominantly revolve around fluctuations in the quality of metallurgical coke, as outlined in the fluctuation pattern in Fig. 9 of Section 2.1.

In this study, the true PFs within our model remains unknown, necessitating the determination of an approximate PFs to assess algorithm performance. When dealing with static problems, it is common practice to combine non-dominated solutions from multiple algorithms to approximate the true PFs [42]. However, dynamic problems often prove more intricate than their static counterparts, rendering the solutions obtained through this approach less accurate. Consequently, this paper introduces a method designed specifically for finding approximate PFs in dynamic scenarios. Each dynamic problem is viewed as an amalgamation of several static problems, which are tackled individually. The condition parameters of each environmental setting are treated as constants. We employ the multiobjective evolutionary algorithm based on decomposition (MOEA/D) to independently evolve the problems in each environmental setting for 10 calculations, each extending to 1000 generations. In the selection strategy, the neighborhood replacement threshold is set at 2 to enhance the distribution of solutions. The set of non-dominated solutions from these 10 calculations constitutes the approximate PFs for the current environmental context.

Due to the differing scales of objectives in various environments, enabling a comprehensive evaluation and comparison of multiple algorithms across multiple scenarios is essential. In our assessment, we normalize the approximate PFs and PFs obtained by all algorithms within each environment simultaneously, scaling them to the range of 0 to 1. The normalization formulas for the approximate PFs and objectives are as follows:

$$\overline{PF_i} = \frac{PF_i - PF_i^{min}}{PF_i^{max} - PF_i^{min}} \tag{19}$$

$$\overline{F_i} = \frac{F_i - PF_i^{min}}{PF_i^{max} - PF_i^{min}} \tag{20}$$

Where, PF_i signifies the approximate PFs value for the ith environment, where PF_i^{max} represents the maximum value within this approximate PF, and PF_i^{min} denotes its minimum value. F_i, on the other hand, represents the achieved objective values for the ith environment.

4.3.2. Experimental settings

All models are dual-objective problems, with a population size of 300 and 19 operational variables for T-DCMOEA/D. The evolution was halted when the environmental changes for Instance1, Instance2, and Instance3 reached 34, 26, and 25 times respectively. Three different change frequencies, denoted by τ_t, were employed for testing. The first environmental change occurred after 200 generations of evolution. All algorithms were independently run 20 times for each problem, and the mean and variance of their evaluation metrics were calculated. The replacement frequency of individuals in the selection strategy was set to 2.

Within the section pertaining to the creation of the initial population using tensors, the low sensitivity of the parameters can be seen based

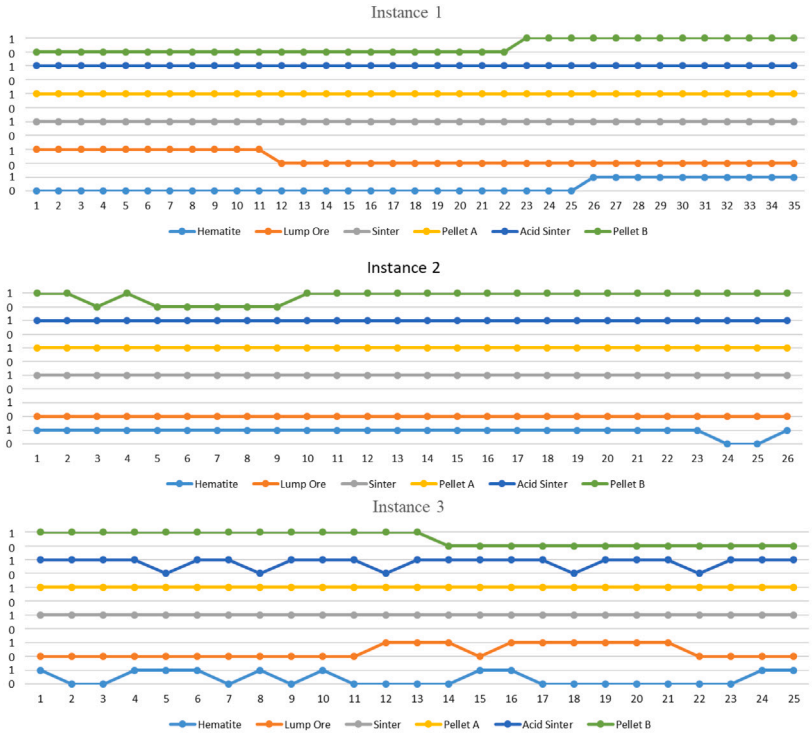


Fig. 9. Change in ore type.

Table 8
Mean and variance of MIGD for blast furnace operation optimization problems.

Prob.	τ_t	IGP-DCMOEA/D	ISVM-DCMOEA/D	KGB-DCMOEA/D	HRS-DCMOEA/D	T-DCMOEA/D
Instance1	5	4.68e−02(3.46e−03)+	7.78e−02(3.54e−03)+	1.50e−01(6.13e−03)+	2.76e−02(1.88e−03)+	2.08e−02(8.90e−02)
	10	3.81e−02(2.55e−03)+	3.98e−02(1.62e−03)+	8.49e−02(5.89e−03)+	2.22e−02(1.59e−03)+	1.61e−02(9.26e−04)
	20	3.03e−02(8.01e−04)+	2.51e−02(9.65e−04)+	4.13e−02(4.27e−03)+	1.35e−02(1.25e−03)+	1.25e−02(8.70e−04)
Instance2	5	4.28e−02(1.15e−03)+	6.09e−02(4.55e−03)+	1.49e−01(3.58e−02)+	3.64e−02(9.65e−04)+	1.27e−02(3.12e−03)
	10	3.77e−02(1.33e−03)+	4.10e−02(4.21e−03)+	9.93e−02(2.89e−02)+	3.17e−02(1.46e−02)+	1.10e−02(3.44e−03)
	20	2.96e−02(9.05e−04)+	3.12e−02(3.85e−03)+	4.86e−02(2.89e−02)+	3.04e−02(1.29e−03)+	8.28e−03(1.28e−03)
Instance3	5	6.26e−02(2.62e−03)+	6.38e−02(2.54e−03)+	7.56e−02(6.85e−03)+	3.12e−02(9.31e−04)+	2.80e−02(5.21e−04)
	10	4.77e−02(1.02e−03)+	4.67e−02(1.89e−03)+	7.21e−02(5.32e−03)+	2.91e−02(9.12e−04)+	2.18e−02(9.70e−04)
	20	3.41e−02(1.23e−03)+	3.24e−02(1.06e−03)+	7.02e−02(4.63e−03)+	2.87e−02(9.31e−04)+	1.65e−02(9.85e−04)

on the experimental analysis in Literature [29]. Therefore, the crucial threshold $Train_{low}$ for the tensor training model is set to 8. In the BHT-ARIMA approach, the parameter for MDT decomposition is configured as 2, and in Tucker decomposition, the size of the core tensor is fixed at 10×2. As for ARIMA, the parameters q, d, and p are set at 3, 2, and 1, respectively.

Graphs are generated and computing MIGD and MHV are performed using normalized values. The reference point z for MHV is set at (1.0, 1.0). The comparative algorithms employed in the experiments remain IGP-DCMOEA/D, ISVM-DCMOEA/D, KGB-DCMOEA/D and HRS-DCMOEA/D. The fundamental parameters of these comparative algorithms are configured identically to T-DCMOEA/D.

4.4. Experimental results and analysis

The results of solving the optimization model for blast furnace operation, as formulated in this manuscript, are presented below. Tables 8 and 9 depict the mean and variance values of MIGD and MHV, respectively. It is noteworthy that T-DCMOEA/D demonstrates a clear advantage in solving the practical problem model under all conditions. The effectiveness of the tensor-based initial population generation method endures.

When $\tau_t = 10$, the evolution process of MIGD is depicted in Fig. 10. T-DCMOEA/D demonstrates a quicker adaptability to environmental changes in most scenarios. In contrast to benchmark test problems, the irregular variations in real-world environments impose additional convergence pressure on the solutions. These pressures further underscore the advantages of the initial population generation strategy within T-DCMOEA/D.

Fig. 11 illustrates the dynamic PFs obtained by the three algorithms for each problem under all environments when $\tau_t = 10$. A more intuitive observation reveals that IGP-DCMOEA/D and HRS-DCMOEA/D demonstrate superior convergence, while ISVM-DCMOEA/D and KGB-DCMOEA/D exhibit a broader distribution. Due to the decomposition-based prediction approach in T-DCMOEA/D, it effectively compensates for the shortcomings of the four comparative algorithms. The distribution plots of solutions indicate that rapidly changing environments exert pressure on the distribution of all five algorithms.

4.4.1. Actual data comparison experiments

In demonstrating the algorithm's efficacy in blast furnace operations optimization, this section contrasts the target values of a subset of operational variables obtained from the algorithm with the actual values in the sample. The two-objective optimization scenario presents a situation where the enhancement of one objective occurs concurrently with the deterioration of another. Therefore, in order to facilitate the comparison of the superiority of the algorithms, under the condition

Table 9
Mean and variance of MHV for blast furnace operation optimization problems.

Prob.	τ_t	IGP-DCMOEA/D	ISVM-DCMOEA/D	KGB-DCMOEA/D	HRS-DCMOEA/D	T-DCMOEA/D
Instance1	5	6.15e−01(6.16e−03)+	5.82e−01(5.63e−03)+	2.67e−01(3.98e−03)+	6.56e−01(2.62e−03)=	6.65e−01(1.62e−03)
	10	6.32e−01(3.89e−03)+	6.33e−01(1.13e−03)+	2.88e−01(4.39e−03)+	6.66e−01(2.67e−03)+	6.72e−01(1.71e−03)
	20	6.47e−01(1.51e−03)+	6.57e−01(1.85e−03)+	2.90e−01(4.82e−03)+	6.78e−01(2.57e−03)+	6.89e−01(1.16e−03)
Instance2	5	6.37e−01(2.75e−03)+	6.01e−01(5.19e−03)+	2.84e−01(3.54e−03)+	6.63e−01(3.38e−02)+	6.89e−01(7.06e−03)
	10	6.46e−01(3.17e−03)+	6.41e−01(4.62e−03)+	2.92e−01(3.92e−03)+	6.72e−01(1.83e−02)+	6.92e−01(5.70e−03)
	20	6.62e−01(1.43e−03)+	6.65e−01(4.84e−03)+	3.03e−01(3.94e−03)+	6.78e−01(2.31e−02)+	6.98e−01(2.33e−03)
Instance3	5	5.93e−01(2.08e−03)+	6.01e−01(1.85e−03)+	2.82e−01(3.21e−03)+	6.77e−01(1.41e−03)-	6.58e−01(1.03e−03)
	10	6.24e−01(2.21e−03)+	6.23e−01(2.26e−03)+	2.96e−01(3.45e−03)+	6.78e−01(9.01e−04)-	6.67e−01(1.05e−03)
	20	6.46e−01(2.89e−03)+	6.58e−01(2.56e−03)+	2.99e−01(3.76e−03)+	6.79e−01(1.04e−03)=	6.76e−01(1.06e−03)

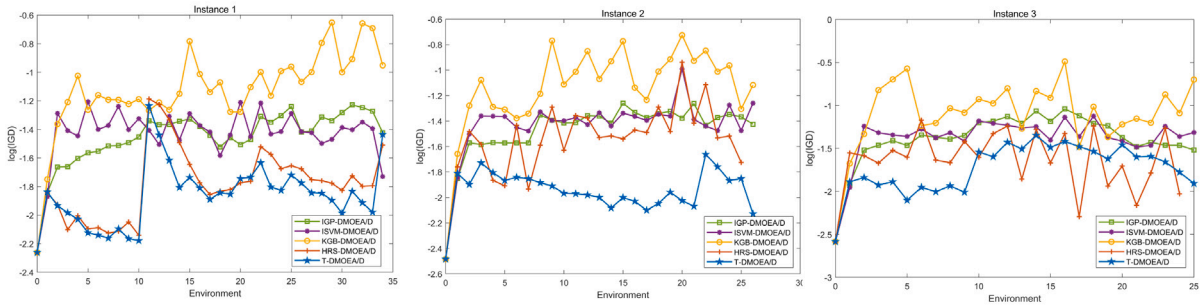


Fig. 10. Evolution of MIGD values for blast furnace operation optimization problems with $\tau_t = 10$ and $n_t 10$.

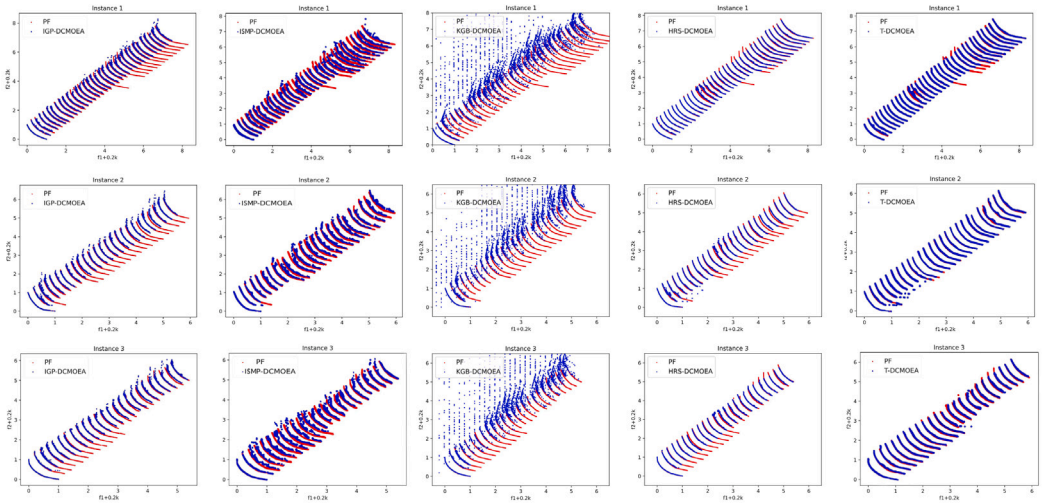


Fig. 11. Distribution of solutions to blast furnace operation optimization problems for $\tau_t = 10$.

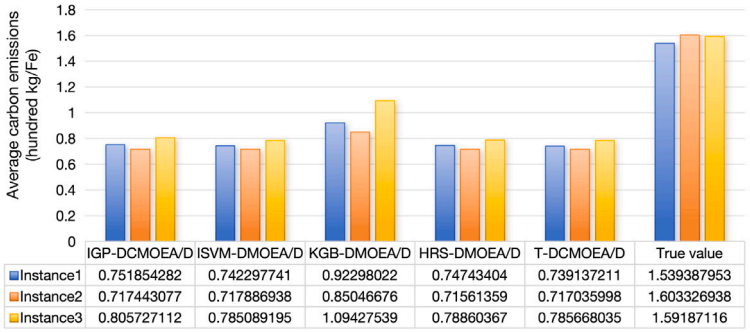
that one of the objectives in the operational variables is consistent, we select another objective value for comparison. We identify, based on actual values, a set of operational variables with target values identical to or closest to the given values for the purpose of comparison.

Fig. 12 shows the comparison of the average carbon emissions at the same cost and the percentage reduction in carbon emissions for each algorithm compared to the actual situation. The graph clearly indicates that the carbon emissions calculated by the five algorithms are significantly lower than the actual values. Despite the minor differences among the algorithms, the algorithm proposed in this paper shows a slight advantage. Fig. 13 presents the comparison of the average cost at the same level of carbon emissions and the percentage of cost savings for each algorithm compared to the actual situation. It is easily observed that the costs calculated by the five algorithms are much lower than the actual values, highlighting a distinct advantage of the algorithm proposed in this paper. In summary, the proposed algorithm proves to be highly effective and applicable to the practical model.

4.4.2. Comparative study of different strategies

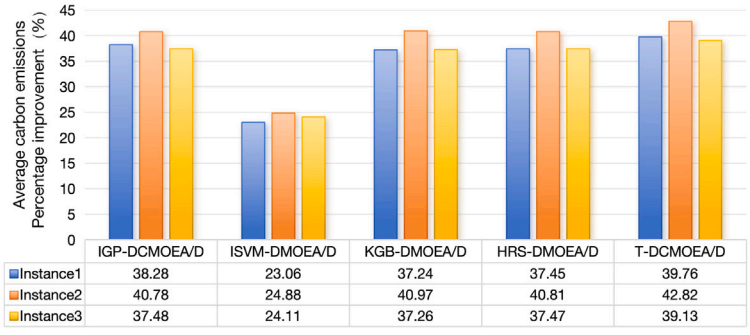
This section analyzes the influence of the tensor prediction model on the algorithm and concurrently assesses the effectiveness of a tensor model based on decomposition. The DCMOEA/D algorithm is an approach that does not employ a tensor prediction model to predict initial population changes in the environment. T-DCMOEA/D-NonDec is an algorithm that tackles subproblems without utilizing the decomposition concept. T-DCMOEA/D-NonDec considers the three-dimensional tensor formed by operational variables as a single problem for prediction. In this context, the core tensor size in the Tucker decomposition is set to 200 × 10× 2. The remaining parameters and strategies of DCMOEA/D and T-DCMOEA/D-NonDec are consistent with those of the T-DCMOEA/D algorithm.

When τ_t is set to 10, Table 10 displays the MIGD values for the three algorithms. The table distinctly indicates that the incorporation of the



	IGP-DCMOEA/D	ISVM-DMOEA/D	KGB-DMOEA/D	HRS-DMOEA/D	T-DCMOEA/D	True value
Instance1	0.751854282	0.742297741	0.92298022	0.74743404	0.739137211	1.539387953
Instance2	0.717443077	0.717886938	0.85046676	0.71561359	0.717035998	1.603326938
Instance3	0.805727112	0.785089195	1.09427539	0.78860367	0.785668035	1.59187116

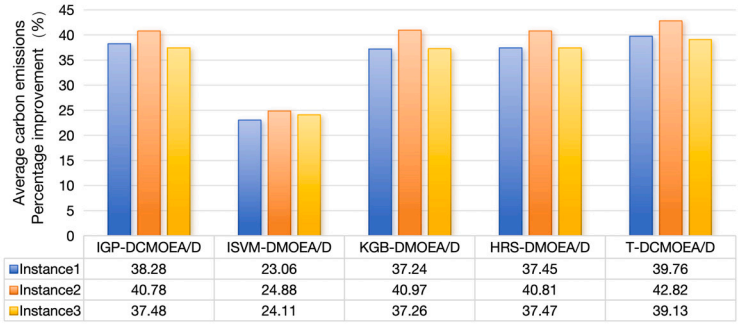
(a) Comparison with actual values



	IGP-DCMOEA/D	ISVM-DMOEA/D	KGB-DMOEA/D	HRS-DMOEA/D	T-DCMOEA/D
Instance1	38.28	23.06	37.24	37.45	39.76
Instance2	40.78	24.88	40.97	40.81	42.82
Instance3	37.48	24.11	37.26	37.47	39.13

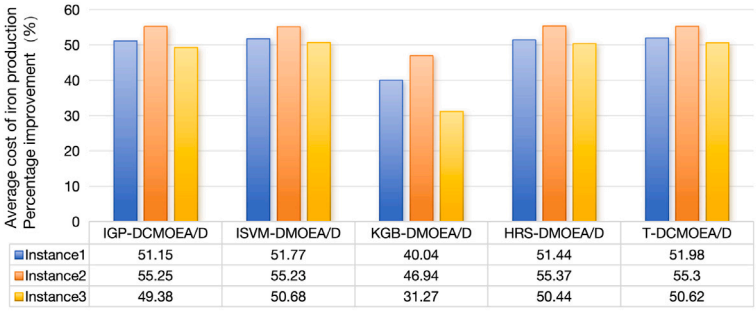
(b) Percentage improvement

Fig. 12. Comparison of carbon emissions for the same iron production.



	IGP-DCMOEA/D	ISVM-DMOEA/D	KGB-DMOEA/D	HRS-DMOEA/D	T-DCMOEA/D
Instance1	38.28	23.06	37.24	37.45	39.76
Instance2	40.78	24.88	40.97	40.81	42.82
Instance3	37.48	24.11	37.26	37.47	39.13

(a) Comparison with actual values



	IGP-DCMOEA/D	ISVM-DMOEA/D	KGB-DMOEA/D	HRS-DMOEA/D	T-DCMOEA/D
Instance1	51.15	51.77	40.04	51.44	51.98
Instance2	55.25	55.23	46.94	55.37	55.3
Instance3	49.38	50.68	31.27	50.44	50.62

(b) Percentage improvement

Fig. 13. Comparison of iron production for the same carbon emissions.

Table 10
Comparison of algorithmics strategy MIGD values.

Prob.	DCMOEA/D	T-DCMOEA/D-NonDec	T-DCMOEA/D
Instance1	5.92e−02(2.42e−02)+	1.81e−02(2.50e−03)+	1.61e−02(9.26e−04)
Instance2	7.06e−02(4.60e−02)+	1.44e−02(2.97e−03)+	1.10e−02(3.44e−03)
Instance3	7.27e−02(2.35e−02)+	2.22e−02(7.43e−04)+	2.18e−02(9.70e−04)

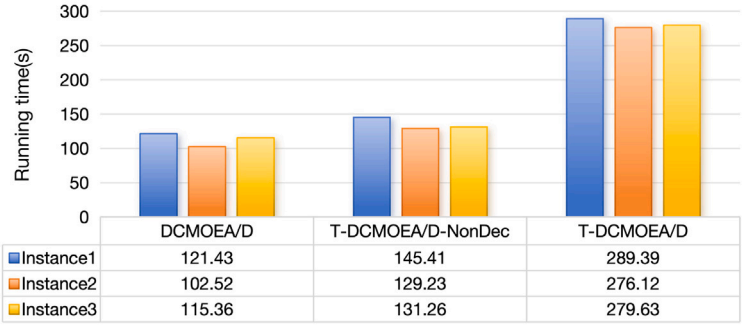


Fig. 14. Algorithm runtime comparison.

tensor prediction strategy significantly enhances the algorithm's performance. The T-DCMOEA/D algorithm, grounded on the decomposition approach, exhibits notably exceptional performance.

The runtime performance of the three algorithms for each problem is illustrated in Fig. 14. Upon comparison, it becomes apparent that the incorporation of the prediction strategy and an increased number of subproblems both leads to an extension of the algorithm's runtime. While the MIGD value of the T-DCMOEA/D-NonDec algorithm is slightly larger than that of T-DCMOEA/D, the algorithm's runtime is significantly reduced. In scenarios where algorithmic performance requirements are not strict but a shorter runtime is desired, the T-DCMOEA/D-NonDec algorithm can be given priority.

5. Conclusions

In this paper, we have developed a dynamic operation optimization framework for a blast furnace, based on actual data from a steel plant, to optimize production costs and carbon emissions. A dynamic multiobjective operation optimization model is built based on data and mechanisms, which involves time-related parameters and specific conditional variables. A T-DCMOEA/D algorithm is designed with a more essential relationship among the model parameters of blast furnace operation optimization. The setting of operational variables is updated as long as the production condition changes.

The experimental tests across practical instances demonstrate that our proposed algorithm significantly reduces the production cost by approximately 50% and carbon emissions by 40% compared to traditional manual parameter-setting methods. Comparative experiments with existing algorithms (IGP-DCMOEA/D, ISVM-DCMOEA/D, KGB-DCMOEA/D, and HRS-DCMOEA/D) further validate the superiority of our approach, which achieves the best performance in most experimental scenarios. In the future research, efforts will be made to apply the proposed method to the blast furnace production environments with frequent fluctuations in ore or fuel quality, to further improve the cost-effectiveness and environmental protection of the blast furnace production, and to provide technical support for the decarbonization transformation of the iron and steel industry.

CRediT authorship contribution statement

Yumeng Zhao: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation. **Xianpeng Wang:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Xiangman Song:** Writing – review & editing, Resources, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Luo, X. Zhang, M. Kan, L. Deng, C. Yang, Z. Song, Data-driven soft sensors in blast furnace ironmaking: a survey, Front. Inf. Technol. Electron. Eng. 24 (3) (2023) 327–354, http://dx.doi.org/10.1631/FITEE.2200366.
[2] M. Naito, K. Takeda, Y. Matsui, Ironmaking technology for the last 100 years: deployment to advanced technologies from introduction of technological know-how, and evolution to next-generation process, ISIJ Int. 55 (1) (2015) 7–35, http://dx.doi.org/10.2355/isijinternational.55.7.
[3] P. Zhou, D. Guo, H. Wang, T. Chai, Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2017) 4007–4021, http://dx.doi.org/10.1109/TNNLS.2017.2749412.
[4] Y. Li, H. Li, J. Zhang, S. Zhang, Data and knowledge driven approach for burden surface optimization in blast furnace, Comput. Electr. Eng. 92 (2021) 107191, http://dx.doi.org/10.1016/j.compeleceng.2021.107191.
[5] L. Tang, Y. Meng, Data analytics and optimization for smart industry, Front. Eng. Manag. 8 (2) (2021) 157–171, http://dx.doi.org/10.1007/s42524-020-0126-0.
[6] D. Gao, C. Yang, X.-Y. Tang, X. Zhu, X. Huang, Fault diagnosis of blast furnace based on incomplete multi-source domain adaptation with feature fusion, Adv. Eng. Inform. 62 (2024) 102946, http://dx.doi.org/10.1016/j.aei.2024.102946.
[7] Q. Zhou, Y. Yin, D. Peng, H. Zhao, L. Xing, X. Jiang, Z. Xu, C. Xu, Multi-objective optimization of blast furnace dosing and operation based on NSGA-II, in: 2022 4th International Conference on Electrical Engineering and Control Technologies, CEECT, IEEE, 2022, pp. 165–169, http://dx.doi.org/10.1109/CEECT55960.2022.10030395.
[8] F. Zou, Q. Guo, G.G. Yen, Dynamic evolutionary multiobjective optimization for open-order coil allocation in the steel industry, Appl. Soft Comput. 146 (2023) 110661, http://dx.doi.org/10.1016/j.asoc.2023.110661.
[9] Z. Liu, Y. Fang, L. Liu, S. Ma, Dynamic harris hawks optimizer based on historical information and tournament strategy and its application in numerical optimization of blast furnace ingredients, Appl. Soft Comput. 164 (2024) 111976, http://dx.doi.org/10.1016/j.asoc.2024.111976.

[10] Y. Li, H. Li, J. Zhang, S. Zhang, Y. Yin, Burden surface decision using MODE with TOPSIS in blast furnace ironmkaing, IEEE Access 8 (2020) 35712–35725, http://dx.doi.org/10.1109/ACCESS.2020.2974882.
[11] X. Zhang, M. Kano, S. Matsuzaki, A comparative study of deep and shallow predictive techniques for hot metal temperature prediction in blast furnace ironmaking, Comput. Chem. Eng. 130 (2019) 106575, http://dx.doi.org/10.1016/j.compchemeng.2019.106575.
[12] B.K. Mahanta, N. Chakraborti, Tri-objective optimization of noisy dataset in blast furnace iron-making process using evolutionary algorithms, Mater. Manuf. Process. 35 (6) (2020) 677–686, http://dx.doi.org/10.1080/10426914.2019.1643472.
[13] W. Zhu, H. Zhang, C. Zhang, X. Zhu, Z. Guan, J. Jia, Surface defect detection and classification of steel using an efficient swin transformer, Adv. Eng. Inform. 57 (2023) 102061, http://dx.doi.org/10.1016/j.aei.2023.102061.
[14] M.S. Parihar, S.H. Nistala, R. Kumar, S. Raj, A. Ganguly, V. Runkana, Optimization of blast furnace ironmaking using machine learning and genetic algorithms, Steel Res. Int. (2024) 2300788, http://dx.doi.org/10.1002/srin.202300788.
[15] P. Azadi, J. Winz, E. Leo, R. Klock, S. Engell, A hybrid dynamic model for the prediction of molten iron and slag quality indices of a large-scale blast furnace, Comput. Chem. Eng. 156 (2022) 107573, http://dx.doi.org/10.1016/j.compchemeng.2021.107573.
[16] X. Huang, C. Yang, H. Zhang, S. Lou, D. Gao, L. Kong, Data and knowledge collaborative-driven fault identification and self-healing control action inference framework for blast furnace, Expert Syst. Appl. 245 (2024) 123040, http://dx.doi.org/10.1016/j.eswa.2023.123040.
[17] Y. Wu, H. Zhang, L. Jian, Z. Lv, A quantitative causal analysis and optimization framework for inclusions of steel products, Adv. Eng. Inform. 62 (2024) 102629, http://dx.doi.org/10.1016/j.aei.2024.102629.
[18] Z.-H. Zhou, Machine Learning, Springer Nature, 2021.
[19] X. Liu, R. Peng, C. Bai, Y. Chi, H. Li, P. Guo, Technological roadmap towards optimal decarbonization development of China's iron and steel industry, Sci. Total Environ. 850 (2022) 157701, http://dx.doi.org/10.1016/j.scitotenv.2022.157701.
[20] X. Wang, T. Hu, L. Tang, A multiobjective evolutionary nonlinear ensemble learning with evolutionary feature selection for silicon prediction in blast furnace, IEEE Trans. Neural Netw. Learn. Syst. 33 (5) (2022) 2080–2093, http://dx.doi.org/10.1109/TNNLS.2021.3059784.
[21] L. Zhang, Y. Jia, H. Shu, X. Lu, F. Bai, Q. Zhao, D. Tian, The effect of basicity of modified ground granulated blast furnace slag on its denitration performance, J. Clean. Prod. 305 (2021) 126800, http://dx.doi.org/10.1016/j.jclepro.2021.126800.
[22] M.-D. Shieh, Y. Li, C.-C. Yang, Comparison of multi-objective evolutionary algorithms in hybrid Kansei engineering system for product form design, Adv. Eng. Inform. 36 (2018) 31–42, http://dx.doi.org/10.1016/j.aei.2018.02.002.
[23] K. Deb, U.B. Rao N, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 803–817.
[24] F. Wang, J. Xie, A. Zhou, K. Tang, A new prediction strategy for dynamic multi-objective optimization using diffusion model, IEEE Trans. Evol. Comput. (2025) http://dx.doi.org/10.1109/TEVC.2025.3551323, 1–1.
[25] W. Song, Z. Liu, J. Yu, X. Sun, Y. Jin, K.W. Lai, Multisource and hidden source-based knowledge transfer for solving dynamic multiobjective optimization problems, IEEE Trans. Evol. Comput. (2025) http://dx.doi.org/10.1109/TEVC.2025.3550557, 1–1.
[26] L. Yan, W. Qi, J. Liang, B. Qu, K. Yu, C. Yue, X. Chai, Inter-individual correlation and dimension based dual learning for dynamic multi-objective optimization, IEEE Trans. Evol. Comput. (2023) http://dx.doi.org/10.1109/TEVC.2023.3235196.
[27] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.
[28] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2013) 602–622, http://dx.doi.org/10.1109/TEVC.2013.2281534.
[29] Q. Shi, J. Yin, J. Cai, A. Cichocki, T. Yokota, L. Chen, M. Yuan, J. Zeng, Block Hankel tensor ARIMA for multiple short time series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, (04) 2020, pp. 5758–5766, http://dx.doi.org/10.1609/aaai.v34i04.6032.
[30] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.
[31] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via Kalman filter prediction, IEEE Trans. Cybern. 46 (12) (2015) 2862–2873, http://dx.doi.org/10.1109/TCYB.2015.2490738.
[32] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2017) 609–622, http://dx.doi.org/10.1109/TEVC.2017.2749619.
[33] H. Zhang, J. Ding, M. Jiang, K.C. Tan, T. Chai, Inverse gaussian process modeling for evolutionary dynamic multiobjective optimization, IEEE Trans. Cybern. 52 (10) (2021) 11240–11253, http://dx.doi.org/10.1109/TCYB.2021.3070434.
[34] D. Xu, M. Jiang, W. Hu, S. Li, R. Pan, G.G. Yen, An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 26 (4) (2021) 690–703, http://dx.doi.org/10.1109/TEVC.2021.3115036.
[35] Y. Ye, L. Li, Q. Lin, K.-C. Wong, J. Li, Z. Ming, Knowledge guided Bayesian classification for dynamic multi-objective optimization, Knowl.-Based Syst. 250 (2022) 109173, http://dx.doi.org/10.1016/j.knosys.2022.109173.
[36] H. Li, Z. Wang, C. Lan, P. Wu, N. Zeng, A novel dynamic multiobjective optimization algorithm with hierarchical response system, IEEE Trans. Comput. Soc. Syst. 11 (2) (2024) 2494–2512, http://dx.doi.org/10.1109/TCSS.2023.3293331.
[37] Y. Guo, G. Chen, C. Yue, J. Liang, Y. Wang, S. Yang, Benchmark problems for CEC2023 competition on dynamic constrained multiobjective optimization, 2023, https://www.scholat.com/vpost.html?pid=205654.
[38] T. Yokota, B. Erem, S. Guler, S.K. Warfield, H. Hontani, Missing slice recovery for tensors using a low-rank model in embedded space, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8251–8259, http://dx.doi.org/10.48550/arXiv.1804.01736.
[39] S. Joe Qin, Statistical process monitoring: basics and beyond, J. Chemom.: A J. Chemom. Soc. 17 (8–9) (2003) 480–502.
[40] M. Greenacre, P.J. Groenen, T. Hastie, A.I. d'Enza, A. Markos, E. Tuzhilina, Principal component analysis, Nat. Rev. Methods Prim. 2 (1) (2022) 100.
[41] V.W. Berger, Y. Zhou, Kolmogorov–smirnov test: Overview, 2014, Wiley statsref: Statistics reference online.
[42] X. Wang, Y. Wang, L. Tang, Q. Zhang, Multiobjective ensemble learning with multiscale data for product quality prediction in iron and steel industry, IEEE Trans. Evol. Comput. 28 (4) (2024) 1099–1113, http://dx.doi.org/10.1109/TEVC.2023.3290172.