Journal of Control, Automation and Electrical Systems
https://doi.org/10.1007/s40313-021-00837-3

Product Sequencing and Blending of Raw Materials to Feed Arc Furnaces: A Decision Support System for a Mining-Metallurgical Industry

Rafael de Freitas Bacharel[1] · Marcone Jamilson Freitas Souza[2] · Luciano Perdigão Cota[3]

Received: 29 April 2021 / Revised: 5 August 2021 / Accepted: 10 September 2021

Abstract

A large amount of data available today and the complex situations present in the industry make decision support systems increasingly necessary. This work deals with a problem of a mining-metallurgical industry in which the production of products used to feed arc furnaces must be sequenced in work shifts. There is a due date and a quality specification for each product. These products are generated from raw materials available in a set of silos and must satisfy the required quality specifications. The aim is to minimize the total production time and the total tardiness. To solve it, we developed a decision support system that applies a matheuristic algorithm to do the product schedule and determine the amount of raw material to produce each product. In the proposed algorithm, the products generated in each work shift are chosen through a dispatch heuristic rule based on the shortest production time. In turn, the amount of raw material to be used is calculated by solving a goal linear programming formulation of a blending problem. We generate instances that simulate real cases to evaluate the developed algorithm. The results show a good performance of the proposed algorithm, validating its use as a tool to support decision-making.

Keywords Arc furnace · Decision support system · Blending · Heuristic · Operations research in industry

1 Introduction

Arc furnaces are one of the most important pieces of equipment in the metallurgical industry due to their versatility in producing different types of steel. According to Worldsteel (2020), of all global crude steel production in 2019, 27.9% result from arc furnaces. Usually, an arc furnace is fed by scrap and elements from different alloys, allowing the adjustment of the steel's chemical composition to be generated.

Equipment such as arc furnaces still has a low level of automation in its operation, leaving the production decisions to be carried out by its operators according to their experiences (Shyamal and Swartz 2017). A survey carried out by Olivier and Craig (2017) on the degree of automation in the global mineral processing industry shows that operators' actions on processes are still frequent. Given the massive amount of data generated by industrial processes and the different situations for decision making, decision support systems are increasingly important, as they allow a complete analysis of the entire production chain (Liu and Zaraté 2014).

The present work deals with a problem of a mining-metallurgical industry in which the production that feeds arc

The authors are grateful for the support provided by the Universidade Federal de Ouro Preto, Instituto Tecnológico Vale, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001, CNPq (Grant 303266/2019-8), and FAPEMIG (Grant PPM CEX 676/17). An early version of the paper was presented at XXIII Congresso Brasileiro de Automática (CBA 2020).

✉ Luciano Perdigão Cota
luciano.p.cota@itv.org

Rafael de Freitas Bacharel
rafael.bacharel@aluno.ufop.edu.br

Marcone Jamilson Freitas Souza
marcone@ufop.edu.br

1 Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração, Instituto Tecnológico Vale, Universidade Federal de Ouro Preto, Ouro Preto, Brazil

2 Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto, Brazil

3 Instituto Tecnológico Vale, Ouro Preto, Brazil

furnaces must be sequenced in work shifts. In the company under study, this task is done manually.

Sequencing the production and performing the correct blending of materials is a problem found in different industry types. For example, in the oil industry, Bayu et al. (2020) deals with the scheduling of gasoline blending and distribution. Franzoi et al. (2019) address the scheduling of processing units and blending for gasoline production. Pereira et al. (2018, 2020) deals with a crude oil scheduling problem in a refinery using a multi-objective evolutionary algorithm. For the mineral industry, Rezakhah et al. (2020) consider ore blending in stockpiles from the open-pit mine scheduling. However, to our knowledge, the problem of product sequencing and blending of raw materials to feed arc furnaces has not yet been addressed in the literature. Feeding an arc furnace with the correct blend of raw materials requires important decisions, as this can affect steel specifications. The blending process allows generating a product with the right proportions of raw material, increasing the quality of production and its economic benefits (Lingshuang et al. 2013). Besides, scheduling efficiently the products that feed the furnace reduces the total production time and generates more products.

In this work, we propose a decision support system based on a matheuristic algorithm to perform the production scheduling and determine the blends that form each product in different planning horizons. A heuristic dispatch rule chooses the products to be produced in each work shift. The optimal quantities of raw materials to generate each product are determined through a goal linear programming formulation. As the proposed algorithm combines heuristic and exact procedures, it is considered a matheuristic, according to Ribeiro et al. (2020). We generate instances that simulate real cases to evaluate the developed algorithm. The results generated show a good performance of the proposed algorithm, validating its use as a tool to support decision-making.

The remainder of this paper is organized as follows. A literature review is done in Sect. 2. In Sect. 3, the problem is characterized. Section 4 shows the formulation developed for the blending problem. Section 5 presents the proposed algorithm for the scheduling production. Computational experiments are reported and discussed in Sect. 6. Finally, the conclusions and proposals for future work are presented in Sect. 7.

2 Literature Review

A literature review involving arc furnaces, scheduling problems, and blending problems in mining and metallurgical industries is performed in this section.

Several works related to arc furnaces deal with reducing energy consumption since this consumption represents a high cost. Hernández et al. (2020) seek an optimal melting profile to minimize energy losses during production batches using mathematical models of an arc furnace. The models are solved by differential equations, while a control vector parametrization technique obtains the optimum values for the setpoints. Saboohi et al. (2019) propose a framework to define various settings for the control of an arc furnace, such as the employed power, carbon injection, among others. Based on models of an arc furnace and their restrictions, the authors seek to increase their efficiency by maximizing the energy transferred to the scrap during refining and, consequently, reduce operating costs. The authors combine optimization algorithms such as genetic algorithms (GA) and sequential quadratic programming (SQP) to solve the proposed model. The results show a reduction in energy consumption, in contrast to an increase in additives, such as oxygen and carbon. Lin et al. (2012) propose a Multi-objective Particle Swarm Optimization (MOPSO) algorithm to reduce electricity consumption, refining time, electrode consumption, and extend the life of the interior furnace. As an output of the algorithm, the developed power supply model achieves the objectives established in the simulations performed.

There are several approaches in the literature to solve scheduling problems. The most common is applying heuristic methods, given the complexity of the problem (Pinedo 2012). Gomes et al. (2021) develop a matheuristic to schedule the heat treatment line of a multinational steel company, seeking to minimize the total tardiness and energy costs. Thus, the initial solutions for each objective are generated by a Mixed-Integer Linear Programming (MILP) formulation. The Multi-objective General Variable Neighborhood Search (MOVNS) metaheuristic is used to explore the solution space. Tests carried out with industry data showed reductions in energy consumption and tardiness. Baykasoğlu and Ozsoydan (2018) present a study of dynamic scheduling of production in heat treatment furnaces. Events such as machine breakdown, change in due dates, change in task priorities, and the arrival of new tasks are considered in the scheduling. The authors used the Greedy Randomized Adaptive Search Procedure (GRASP) algorithm to generate the solution. In the pilot study carried out, the proposed solution reduced the total downtime of the furnaces and the consequent increase in production. Araujo et al. (2008) treat the problem of batch sizing and scheduling production in small foundries in Brazil. The developed model considers characteristics such as a planning horizon, variation of the furnace capacity, stock generation, the penalty for delay, and changing the furnace setup. The authors developed two local search methods and used the Simulated Annealing (SA) metaheuristic to find better solutions. In the literature, there are several works that address scheduling problems (Saberi-Aliabad et al. 2020; Cota et al. 2014, 2019; Haddad et al. 2015).

Goal linear programming is widely used to solve blending problems (Chanda and Dagdelen 1995; Costa et al. 2005; Moraes et al. 2006). This optimization technique is used in these works to minimize the quality deviations of the generated products.

Other approaches to the problem are also used, such as fuzzy logic in Xu et al. (2008) and stochastic optimization in Lingshuang et al. (2013). Yuan et al. (2020) propose an optimization model for blending coke to be used in blast furnaces. The goal is to determine the best blend for coke at the lowest cost. Initially, the authors seek to estimate the quality of the coke using Gaussian functions and the Extreme Gradient Boosting Algorithm (Xgboost) to select the most relevant characteristics. Then, they apply the Support Vector Regression (SVR) algorithm to forecast the products to be generated. Next, the blending optimization model is solved using a modified version of the Particle Swarm Optimization (MPSO) algorithm. A similar approach to the preparation of the sinter blend is covered in Zhang et al. (2019). Seeking to maximize profit in the blending of sinter, the authors use the SVR algorithm to estimate the sinter classification based on the properties of the raw materials and, thus, price the generated blend. Finally, they use the Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm to solve the blending problem. The authors report a reduction in the cost of the sinter and an increase in production profit. Huang et al. (2019) seek to reduce energy consumption and production costs with an optimization model for the sinter blend. Initially, the quantities of raw materials that make up the blending are defined considering, in addition to the different qualities, the granulation, and mineralization properties. An initial solution for the blend is generated through linear programming. This solution is used as input to a GA, which generates a group of blending schemes. Then, the chemical properties of the raw materials of these blending schemes are used as inputs of the Least Squares Support Vector Machine (LS-SVM) algorithm for the prediction of energy consumption and productivity in the sintering process. Finally, the blends and their respective blending cost, energy consumption, and productivity rates are evaluated. The blending scheme with the highest economic and technical value is selected. The authors report a significant reduction in carbon emissions.

This work differs from the others in the literature for dealing with the problem of product sequencing and blending of raw materials to feed arc furnaces, considering simultaneously: (1) the blending of raw materials, stored in silos, to generate products obeying lower and upper bounds for their control parameters; (2) the blending aiming to minimize deviations from the grade targets for the control parameters of the products generated; (3) the continuous feeding of silos; (4) the sequencing of these products into work shifts within a planning horizon.

3 Problem Statement

The problem of sequencing products to feed the arc furnace under study has the following characteristics:

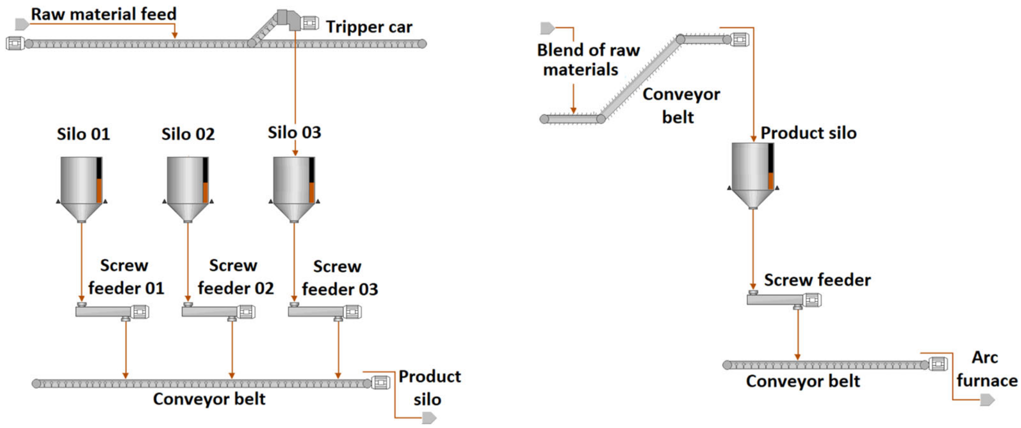
1. There is a set M of ores of different qualities to be used as raw material, $M = \{1, \ldots, |M|\}$;
2. There is a set SM of silos to store the raw material, $SM = \{1, \ldots, |SM|\}$;
3. There is a set PR of products to be generated from the blend of raw materials, $PR = \{1, \ldots, |PR|\}$;
4. There is a set S of quality parameters for raw materials and products, $S = \{1, \ldots, |S|\}$;
5. There is a production planning horizon H, in minutes;
6. There is a set AL of screw feeders, $AL = \{1, \ldots, |AL|\}$;
7. Each screw feeder $ar \in AL$ belongs to a raw material silo $a \in SM$;
8. There is a set CT of conveyor belts, $CT = \{1, \ldots, |CT|\}$;
9. Each product $j \in PR$ is associated with a processing time p_j, a due date d_j;
10. Each product $j \in PR$ is associated with a type of raw material;
11. Every silo $a \in SM$ stores a maximum of Qu tonnes of raw material;
12. Every silo $a \in SM$ must store at least Ql tonnes of raw material;
13. The material contained in each raw material silo $a \in SM$ is associated with a type of raw material;
14. The grade of the parameter $b \in S$ in the raw material silo $a \in SM$ is given by t_{ab}, in %;
15. The recommended grade of the parameter $b \in S$ in the product $j \in PR$ is given by tr_{jb} in %;
16. For each product $j \in PR$, there is a lower and upper bound for the control parameter $b \in S$, given by tl_{jb} and tu_{jb}, in %, respectively, to be met;
17. When a raw material silo is being fed, it cannot be used in the productive process.

The goal is to generate a production sequence that meets the required quality specifications and minimizes the weighted sum of the *makespan* and the total tardiness, given by:

$$\min \alpha C_{\max} + (1 - \alpha) \sum_{j \in PR} T_j \tag{1}$$

where:

1. $\alpha \in [0.1]$ is a parameter that reflects the importance of each parcel of the objective function;
2. C_j is the completion time of the product j, in minutes;
3. $C_{\max} = \max\{C_j, j \in PR\}$, in minutes;



(a) Example of setting up silos for storing blends and tripper car. **(b)** Example of setting up silos for storing blends.

Fig. 1 Stages of the analyzed production process

Table 1 Production planning to be executed

Products	Mass (tonnes)	Material	Due date (min)	Parameters (%)				
				A	B	C	D	E
1	18	0	480	21.65	0.87	0.66	68.59	4.32
2	18	3	960	24.11	2.26	1.23	69.17	4.16
3	18	1	960	24.39	1.32	0.59	68.53	5.38
4	21	2	480	22.75	1.92	1.17	67.49	4.29

4. $T_j = \max\{C_j - d_j, 0\}$ is the tardiness of the product j, in minutes.

The following images describe the production process. Figure 1a illustrates raw material silos and the tripper car to distribute the material into the silos. As Wills and Finch (2015), a tripper car is a machine capable of moving on rails and positioning itself above the silos, allowing its load to be delivered to one or more silos. In the analyzed process, a silo is fed at a time. Each raw material silo receives a type of ore with a given chemical composition, determined by laboratory analysis. The material that feeds each of the raw material silos comes from a previous process step, not covered in this work.

Below each silo of raw material, there is a screw feeder, equipment responsible for removing the desired amount of material to compose the product to be generated. This removed material is then deposited on conveyor belts, which send it to the product storage silo, represented in Fig. 1b. Once the product is generated, it is then consumed in the next stage of the production process, that is, in the feeding of the arc furnace.

To illustrate a solution to this problem, let Table 1 be an example of a production planning to be executed. Table 1 reports the product characteristics in this sequence: product ID, mass, type of material, the due date, and its respective desired chemical composition.

Table 2 Characteristics of raw material silos

Silo	Mass (tonnes)	Material	Parameters (%) A	B	C	D	E
1	67	0	24.23	2.48	0.69	68.58	4.08
2	67.51	2	20.66	2.48	1.12	67.69	5.38
3	70.18	3	24.02	0.85	1.00	68.22	5.33
4	159.65	3	24.93	2.24	1.15	69.47	4.28
5	61.23	1	21.37	1.59	0.89	68.26	5.44
6	171.16	0	26.57	0.74	1.23	69.79	5.89
7	253.44	2	24.35	0.50	1.04	68.11	5.12

Table 2 presents the values of mass, type of material, and the grade of each control parameter of the ores contained in the silos of raw materials in the initial instant.

From Tables 1 and 2, it is possible to calculate the composition of the blending required to generate each product. The raw materials used in this blend must be those available in silos that have the same type of raw material required by the product.

When defining the amount of material to be removed from each raw material silo, the screw feeders located just below these are activated, except for those whose amount of material is null. Then, the material is transferred by conveyor belts and sent to the product silo.

4 The Mathematical Formulation of the Blending Problem

The blending problem can be approached through a goal linear programming formulation, associating a deviation variable for each constraint that cannot be obeyed. These deviation variables measure distances from target values to be achieved and represent goals to be reached. So, the objective function of this formulation is formed by the weighted sum of these goals and should be minimized. We adapt the model by Moraes et al. (2006) to solve the blending problem. As the model is applied to generate a single product, for simplicity, we have eliminated the index j in the input data defined in items 15, and 16 of Sect. 3.
Input data:

1. SM: Set of raw material storage silos;
2. S: Set of quality parameters for raw materials and products;
3. PR: Set of products to be generated from the blend of raw materials;
4. t_{ab}: Grade of the parameter $b \in S$ of the raw material stored in the silo $a \in SM$;
5. tr_b: Target for the parameter $b \in S$ in the blending;
6. tl_b: Lower bound for the parameter $b \in S$ in the blending;
7. tu_b: Upper bound for the parameter $b \in S$ in the blending;
8. Q_a: Mass available in the silo $a \in SM$;
9. ws_a: Weight for the use of the silo $a \in SM$;
10. $Qdes$: Desired mass for the product;
11. wm_b: Weight for the deviations from the grade target of the parameter $b \in S$;
12. $type_a$: Binary parameter that assumes value 1 if the raw material silo $a \in SM$ has the same type of material desired for the product j to be blended and 0, otherwise;

Decision variables:

1. x_a: Quantity of mass to be taken from the raw material of the silo $a \in SM$;
2. $dnSiloSM_a$: Negative deviation concerning the amount of raw material remaining in the silo $a \in SM$, used for blending, in tonnes;
3. dnm_b: Negative deviation concerning the grade target for the parameter $b \in S$, in tonnes;
4. dpm_b: Positive deviation concerning the grade target for the parameter $b \in S$, in tonnes;

The blending problem can be formulated through Eqs. (2)–(12).

$$\min \sum_{b \in S} wm_b \times (dpm_b + dnm_b) + \sum_{a \in SM} ws_a \times dnSiloSM_a \tag{2}$$

$$\sum_{a \in SM} x_a = Qdes \tag{3}$$

$$x_a \leq Q_a \quad \forall a \in SM \tag{4}$$

$$\sum_{a \in SM} (t_{ab} - tl_b) \times x_a \geq 0 \quad \forall b \in S \tag{5}$$

$$\sum_{a \in SM} (t_{ab} - tu_b) \times x_a \leq 0 \quad \forall b \in S \tag{6}$$

$$\sum_{a \in SM} (t_{ab} - tr_b) \times x_a + dnm_b - dpm_b = 0 \quad \forall b \in S \tag{7}$$

$$x_a + dnSiloSM_a = Q_a \times type_a \quad \forall a \in SM \tag{8}$$

$$x_a \geq 0 \quad \forall a \in SM \tag{9}$$

$$dnSiloSM_a \geq 0 \quad \forall a \in SM \tag{10}$$

$$dnm_b \geq 0 \quad \forall b \in S \tag{11}$$

$$dpm_b \geq 0 \quad \forall b \in S \tag{12}$$

The objective function described by Eq. (2) seeks to minimize the deviations concerning the grade target for all parameters and the amount of raw material remaining in the silo used for blending. The constraints applied to the model are presented by Eqs. (3)–(8). Constraints (3) determine that $Qdes$ tonnes of product will be produced. Constraints (4) ensure that the mass to be removed from each raw material silo cannot be greater than its available mass. Constraints (5) and (6) ensure compliance with the lower and upper specification limits, while constraints (7) seek to meet the grade target. Constraints (8) indicate that all material from a raw material silo should be removed whenever possible. Equations (9–12) establish the domain of the decision variables.

When solving the blending problem, the mathematical model returns that there is no feasible solution or, if there is, it returns a solution that satisfies the lower and upper bounds stated for the product, minimizing the weighted objective function given by Eq. (2).

5 Proposed Algorithm

A matheuristic algorithm for decision support, named DSS_ASPT, is proposed to schedule the desired products. It combines the Adaptive Shortest Processing Time (ASPT) (Baker 1974) heuristic procedure to generate the production sequence and a goal linear programming formulation to solve the blending problem. Its pseudocode is presented by Algorithm 1.

As input parameters, Algorithm 1 receives the set SM of raw material silos, the set PR of products to be generated, and the production planning horizon H. Besides, the parameter α, the minimum (Ql) and maximum (Qu) mass values for a raw material silo, and a list containing the minimum (tl_b) and maximum (tu_b) values of grades for each parameter b of a given type of material are algorithm inputs. Finally, the algorithm's outputs are the solution s containing the production sequence and its value *fo* according to Eq. (1).

First, some variables, sets, and counters used in the algorithm are initialized, such as the makespan $C_{\max}$, the total tardiness T, the instant h of the horizon planning, and others. In line 9, the list with the limit values of the grades is obtained. In line 10, it is determined how many days make up the production planning horizon H. So a loop is initialized, and in line 12, it is determined which products $p \in PR$ have a due date for the day analyzed, generating the set PR_{Day}. These products are then removed from the set PR so that they are not analyzed in the next iteration.

In line 14, the rejected products from the set $PR_{Rejected}$ are handled by the rejected products treatment (RPT) function. Nothing will be processed in the first iteration of the loop, as there are still no rejected products. Algorithm 2 describes the pseudocode of the RPT function (see Sect. 5.1.1).

Once processed the rejected products, line 15 checks which products belonging to the set PR_{Day} are candidates to be produced. Then, it is solved the blending problem for each product in the set PR_{Day}. The optimizer returns the mass values to be extracted from the raw material silos, indicating whether the product is a candidate to be generated or not. If the optimizer returns null values for the mass to be removed, the product is considered rejected. If the product is rejected, its priority is increased in one unit. The rejected products are then added to the set $PR_{Rejected}$ to be treated in the next iteration.

Then, we start a loop that operates as long as there is a product $p \in PR_{Candidates}$. First, in line 21, we select a product with the shortest completion time to be produced during the current work shift. This product is associated with all the information necessary for its production, such as mass, recommended grades, and acceptable quality limits, i.e., lower and upper bounds for its control parameters. Once the product is selected, the raw material feed (RMF) function analyzes in line 22 if there is any raw material silo that must be fed. The RMF function is described through the pseudocode of Algorithm 3 (see Sect. 5.1.2).

Next, we check whether the raw material silos at that time analyzed contain sufficient mass to generate the product p. The product is rejected in the lack of mass, and its priority is increased by one unit for each rejection. If the product is rejected, it is included in the set $PR_{Rejected}$ and is removed from the set $PR_{Candidates}$ in lines 32 to 34.

If there are sufficient raw materials in silos, the time analyzed h is updated by increasing it with the processing time of the product generated. The processing time p_j is determined by the longest time spent by the screw feeders for extracting the required amount of mass from the raw material silos. As the feeders are activated simultaneously, the one that consumes the most time to finish its activity will be the one that will determine the total duration of the product processing. The makespan $C_{\max}$, the total tardiness T, and the mass present in each silo of raw material are updated using the *UpdateMass*(.) function. Then, in line 28, the partial solution s is updated with the product p and then excluded from the set $PR_{Candidates}$. The loop ends when the set $PR_{Candidates}$ is empty.

Finally, we increase the variable Day in line 37, and the loop continues until all the days of the planning horizon are analyzed.

After evaluating the products for each day of the planning horizon, there may still be rejected products. Therefore, they are treated in the loop initialized in line 40 until all products have been generated, ending the production schedule.

5.1 Complementary Functions

This section presents the functions that contribute to the execution of the DSS_ASPT algorithm.

5.1.1 RPT Function

This function does the treatment of the rejected products in the DSS_ASPT algorithm. It receives as input parameters the set SM of raw material silos, the set $PR_{Rejected}$ of rejected products, the time horizon H, the mass limit values for a silo of raw material (Qu, Ql), and the lower and upper grade bounds (tl_{jb}, tu_{jb}) for each parameter b in the product j. The output is the solution s updated and the set of rejected products $PR_{Rejected}$.

Algorithm 2 describes its pseudocode. First, we check which product can be generated among the rejected products. Then, we initialize a loop to investigate the candidate products. In line 3, the product of the set $PR'_{Candidates}$ with the highest priority and the shortest duration to generated it is selected. Then, we check by the RMF function if there is an empty raw material silo that must be fed.

Suppose there is a sufficient mass of raw material. In that case, we generate the selected product, update the makespan $C_{\max}$, the total tardiness T of the solution, and the mass present in each raw material silo. We update the partial solution s with the product p in line 10 and exclude it from the set $PR'_{Candidates}$. If there is not enough mass, we reject the product and increase its priority in line 14. Products that were not considered candidates have their priority increased in line 19.

5.1.2 RMF Function

This function executes the feed of a raw material silo identified as empty. Initially, we initialize the variables. In line 5, we check if there is an empty raw material silo. A silo is considered empty if its mass value is less than Ql. If it is

Table 3 Characteristics of the instances

# Instance	# Products		# Raw material	Planning horizon	
	Per work shift	Total	silos	Minutes	Days
1	9	30	14	1440	1
2	9	81	14	4320	3
3	10	90	14	4320	3
4	9	189	14	10,080	7
5	10	210	14	10,080	7
6	9	270	14	14,400	10
7	10	300	14	14,400	10

empty, the variable *EmptySilo* assumes the ID of the empty silo. Otherwise, it receives the value 0.

Lines 6–8 identify the most demanded raw material for both the products of the following day of the planning horizon and rejected products. Knowing the raw material to be fed, the empty silo is fed until the mass value reaches the value Q_u (line 9).

When defining the raw material silo to be fed, the RMF function also determines the position of the tripper car along the production planning horizon.

6 Computational Experiments

The DSS_ASPT algorithm was implemented using the C# programming language on the .NET Framework 4.7.2 platform in the Microsoft Visual Studio Community 2017 development environment. We execute the tests on a computer with an Intel Core i7-8550U @ 1.80 GHz × 8 processor, with 16 GB of RAM and Windows 10 64-bit operating system.

To test the algorithm, we generate seven instances based on actual production data from a mining-metallurgical industry. Each instance has the number of raw material silos, the number of products to be generated, the planning horizon, and the due date of each product per work shift. We generate these instances according to the schema below:

1. The flow rate of the screw feeders was set at 30 tonnes/h, and the feed flow of the tripper car at 28 tonnes/h;
2. The mass of the product $j \in PR$ is chosen randomly in the range $[0.7 \times 24, 24]$, where the value 24 represents the maximum mass of the product to be generated in tonnes;
3. The maximum capacity of each raw material silo is Qu = 285 tonnes;
4. The raw material type for generate the product j is chosen randomly in the range [0, 3] to represent each of the four products generated by the company under study;
5. The grades t_{ab} of the mass stored in the raw material silo $a \in SM$ are chosen randomly in the range $[tl_{ab}, tu_{ab}]$, according to the type of raw material stored in silo a;
6. The recommended grades tr_{jb} are chosen randomly in the range $[tl_{jb}, tu_{jb}]$ according to the type of the product j;
7. The due date d_j of the product $j \in PR$ is chosen according to the number of products per work shift, that is, 9 or 10;
8. The mass of material stored in the raw material silo $a \in SM$ is chosen randomly in the interval $[0.7 \times 24, Qu]$;
9. The raw material type stored in silo a is chosen randomly in the range [0, 3].

Table 4 Weights for the quality parameters

	Quality parameters				
	A	B	C	D	E
Weight	5	5	10	100	5

Table 3 shows the characteristics of the instances generated. It has the following organization: (i) the first column: index of the instance; (ii) the second column: the total of products to be generated; (iii) the third column: the number of products with due date per shift work; (iv) the fourth column: number of raw material silos available; and (v) the fifth and sixth: the horizon production planning. The number of silos represents exactly the structure present in the studied industry. We will not disclose the quality parameters and raw materials due to industrial secrecy.

We modeled the blending problem through the goal linear programming formulation described in Sect. 4 and used the LINGO modeler and optimizer software from Lindo Systems Inc., version 10, to solve it. In the developed code, the blending input data are passed to the optimizer through a DLL. The optimizer, in turn, returns the mass to be extracted from each raw material silo. The weights of quality parameters in the model were classified as *Very Important*, *Critical*, and *Very Critical*, with weights set to 5, 10, and 100, respectively, according to the classification of Moraes et al. (2006). Table 4 shows the weights used for each quality parameter

Algorithm 1 DSS_ASPT

Input: Set *SM* of raw material silos, set *PR* of products to be generated, time horizon H, parameter α, minimum mass value Ql, maximum mass value Qu, lower and upper bounds for the parameters

. **Output**: Solution s and objective function value *fo*

$C_{\max} = 0$;
$T = 0$;
$h = 0$;
$Day = 0$;
$fo = 0$;
$PR_{Rejected} \leftarrow \emptyset$;
$PR_{Day} \leftarrow \emptyset$;
$s \leftarrow \emptyset$;
$GradeList = GradeBounds()$;
$NumDays = GetNumberofDays(H)$;
while $Day < NumDays$ **do**
    $PR_{Day} \leftarrow FindDayProducts(PR, Day)$;
    $PR \leftarrow PR \setminus PR_{day}$;
    $(s, PR_{Rejected}) = RPT(SM, PR_{Rejected}, h, PR, Day, Ql, Qu, GradeList)$;
    $(PR_{Candidates}, PR_{RejectedDay}) = FindPossibleProducts(PR_{Day}, SM)$;
    while $PR_{RejectedDay} \neq \emptyset$ **do**
        $IncreasePriority(PR_{RejectedDay})$;
        $PR_{Rejected} \leftarrow PR_{Rejected} \cup PR_{RejectedDay}$;
    end
    while $PR_{Candidates} \neq \emptyset$ **do**
        $(p, X) = FindBestProduct(PR_{Candidates}, SM)$;
        $RMF(SM, PR, PR_{Rejected}, Day, Ql, Qu, GradeList)$;
        if *there are sufficent mass in raw material silos* **then**
            $h = h + p_j$;
            $C_j = h$;
            $T = T + \max\{C_j - d_j, 0\}$;
            $UpdateMass(X, SM)$;
            $s \leftarrow s \cup \{p\}$;
            $PR_{Candidates} \leftarrow PR_{Candidates} \setminus \{p\}$;
        end
        else
            $IncreasePriority(p)$;
            $PR_{Rejected} \leftarrow PR_{Rejected} \cup \{p\}$;
            $PR_{Candidates} \leftarrow PR_{Candidates} \setminus \{p\}$;
        end
    end
    $Day{+}{+}$;
end
if $PR_{RejectedDay} \neq \emptyset$ **then**
    while $PR_{RejectedDay} \neq \emptyset$ **do**
        $RMF(SM, PR, PR_{Rejected}, Day, Ql, Qu, GradeList)$;
        $(s, PR_{rejeitados}) = RPT(SM, PR_{Rejected}, h, PR, Day, Ql, Qu, GradeList)$;
    end
end
$C_{\max} = h$;
$fo = CalculateFO(C_{\max}, T, \alpha)$;
return s, *fo*;

Algorithm 2 RPT

Input: Set SM of raw materials silos, set $PR_{Rejected}$ of rejected products, time h, set PR of products, analyzed day Day, minimum mass value Ql, maximum mass value Qu, lower and upper bounds for the parameters

Output: Solution s and set $PR_{Rejected}$

$(PR'_{Candidates}, PR'_{Rejected}) = FindPossibleProducts(PR_{Rejected}, SM)$;
while $PR'_{Candidates} \neq \emptyset$ **do**
    $(p', X) = FindPriorityProducts(PR'_{Candidates}, SM)$;
    $RMF(SM, PR, PR_{Rejected}, Day, Ql, Qu, GradeList)$;
    if *there are sufficent mass in raw material silos* **then**
        $h = h + p'_j$;
        $C'_j = h$;
        $T = T + \max\{C'_j - d'_j, 0\}$;
        $UpdateMass(X, SM)$;
        $s \leftarrow s \cup \{p'\}$;
        $PR'_{Candidates} \leftarrow PR'_{Candidates} \setminus \{p'\}$;
    end
    else
        $IncreasePriority(PR_{Rejected}[p'])$;
        $PR'_{Candidates} \leftarrow PR'_{Candidates} \setminus \{p'\}$;
    end
end
while $PR'_{Rejected} \neq \emptyset$ **do**
    $IncreasePriority(PR_{Rejected})$;
end
return $s, PR_{Rejected}$;

Algorithm 3 RMF

Input: Set SM of raw materials silos, set PR of products, set $PR_{Rejected}$ of rejected products, analyzed day Day, minimum mass value Ql, maximum mass value Qu, list of grades for each type of material

Output: Set SM

$EmptySilo = 0$;
$MaterialType = 0$;
$MPR_{NextDay} = 0$;
$MPR_{Rejected} = 0$;
$EmptySilo = FindEmptySilo(SM, Ql)$;
$MPR_{NextDay} = FindNextDayMaterial(PR, Day)$;
$MPR_{Rejected} = FindProductRejectedMaterial(PR_{rejeitados})$;
$MaterialType = MostRequestedType(MPR_{NextDay}, MPR_{Rejected}, PR, PR_{Rejected})$;
$SM = FeedSilo(EmptySilo, MaterialType, Qu, GradeList)$;
return SM;

considered in the model. The weights ws_a were set to a high value (bigM) for all silos.

We execute the DSS_ASPT algorithm considering equal importance to the makespan and total tardiness objectives, that is, the parameter α of Eq. (1) was set to 0.5.

Figure 2 shows the graphical interface of the decision support system. The user can define the number of raw material silos, the number of products to be generated, the planning horizon, the process flow rates, the maximum mass, and other parameters. Then, when activating the Run command, the system suggests a product schedule.

Table 5 presents the algorithm results. The first three columns report the instance identifier, the makespan in min-

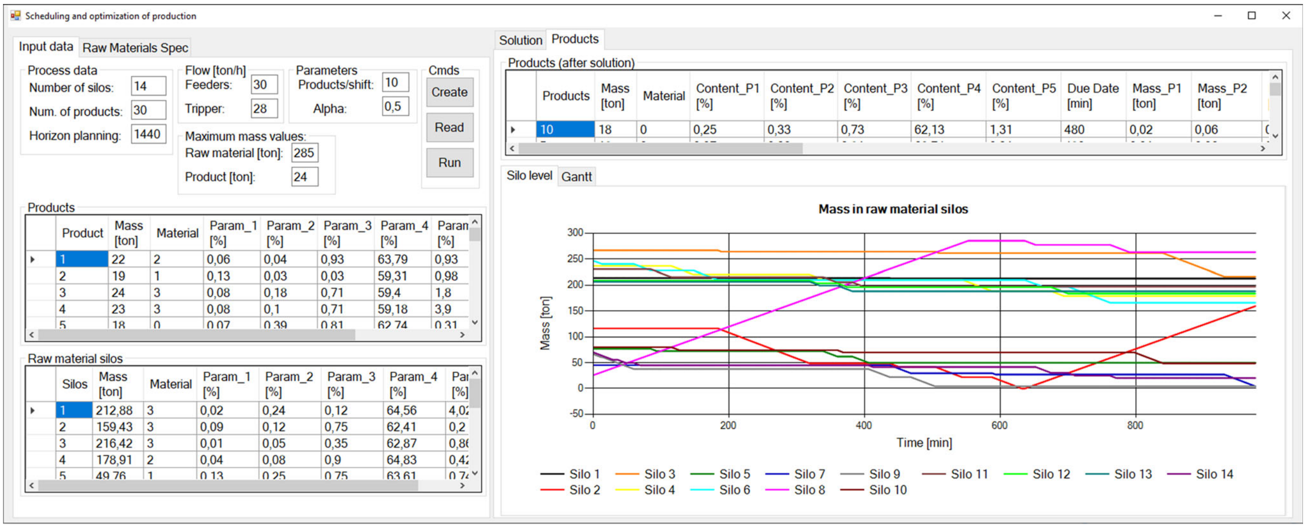


Fig. 2 User interface for the decision support system

Table 5 Results of the proposed algorithm per instance

Instance	Makespan (min)	Total tardiness (min)	Maximum tardiness (min)	Objective function (min)	Runtime (seg)	Average relative percentage deviation				
						A	B	C	D	E
1	660.24	0.00	0.00	330.12	5	0.10	0.16	0.03	0.00	0.08
2	2114.56	0.00	0.00	1057.28	16	0.18	0.12	0.04	0.01	0.13
3	2853.49	0.00	0.00	1426.75	21	0.14	0.14	0.03	0.01	0.06
4	8189.96	2379.56	834.87	5284.76	67	0.18	0.11	0.03	0.01	0.09
5	10,191.08	62,761.94	2564.33	36,476.51	157	0.14	0.21	0.04	0.01	0.09
6	12,526.98	76,006.57	3537.69	44,226.78	215	0.15	0.13	0.05	0.01	0.06
7	14,483.00	122,737.01	4178.04	68,610.01	267	0.15	0.15	0.07	0.01	0.04

utes, and the total tardiness in minutes, respectively. The fourth column shows the longest tardiness, in minutes, among the sequenced products. The value of the objective function generated by the solution is presented, in minutes, in the fifth column. Its sixth column reports the total execution time in seconds. In the last five columns, we report the Average Relative Percentage Deviation ($ARPD_b$) for the parameter b in each instance, calculated according to Eq. (13):

$$ARPD_b = \frac{\sum_{j \in PR} |te_{jb} - tr_{jb}|/tr_{jb}}{|PR|} \tag{13}$$

where te_{jb} is the value encountered for the parameter b in the product j.

Table 5 shows that algorithm DSS_ASPT sequences all products. However, the planning horizon is extrapolated in some instances. This extrapolation occurs in two instances with more products per work shift. Regarding the total tardiness, although this value was high in some instances, such as in instance 7, the maximum tardiness was less than three days. These facts may indicate an oversizing of the production target; fewer products should be produced per work shift in these instances. The deviations from the quality parameters varied according to the priority assigned to each one. Thus, parameters with the highest priority had the smallest deviations. In this case, control parameter D, which has the highest priority, had a maximum deviation of 1%. Parameter C, which has the second-highest priority, had a maximum deviation of 7%. The other parameters had a higher deviation, of up to 21%, to reach the target values. However, it is worth noting that the solution proposed by the algorithm satisfies the lower and upper bounds of the control parameters. We also highlight that the proposed algorithm consumes a low runtime, with the highest equal to 267 s.

Figure 3 shows the behavior of the mass of the materials present in the raw material silos throughout the sequencing. It is possible to observe the feeding behavior of the empty silos. It is also possible to observe the instants in which only rejected products are treated. In such instants, we observe

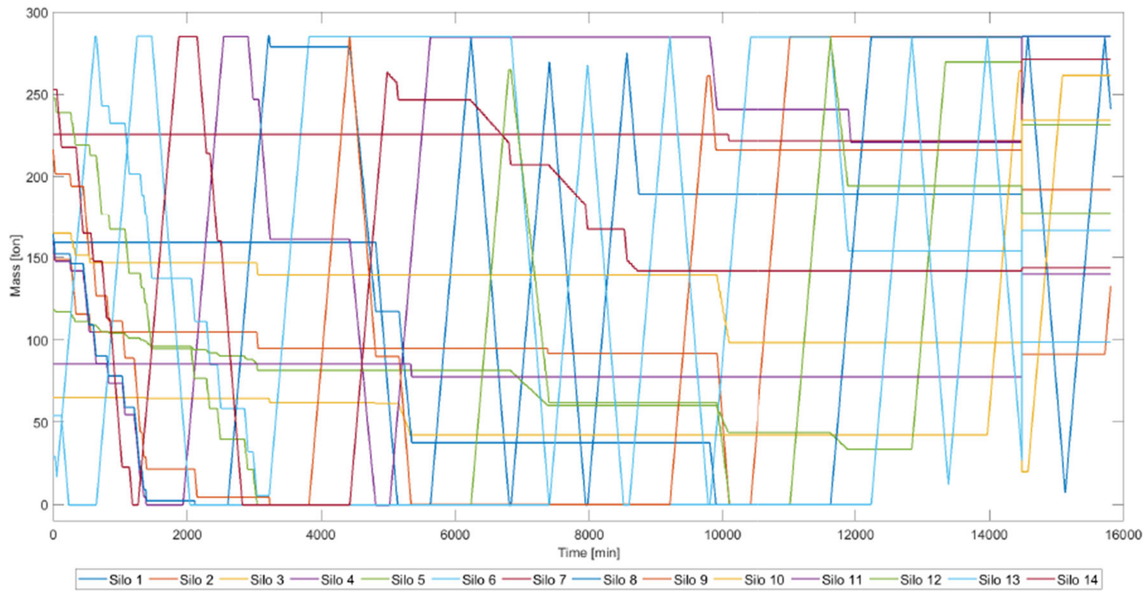


Fig. 3 Behavior of raw material silos for instance 7

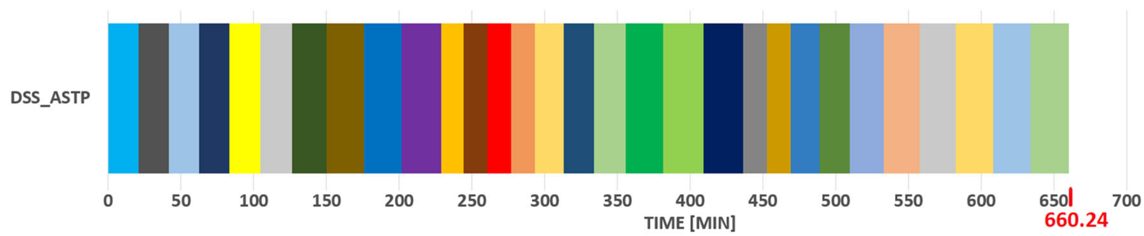


Fig. 4 Schedule generated for the instance with 30 products

that few products are generated due to the unavailability of raw materials. Yet, at the same time, the empty silo is fed with the most requested material type to meet the demand for rejected products.

Figure 4 illustrates through the Gantt chart the schedule generated by the developed algorithm for the instance 1. In this figure, each colored rectangle represents a product. The products are showed in the production sequence, and the width of each rectangle indicates the processing time of the respective product. In this sequencing, the DSS_ASPT generated a solution with a makespan of 660.24 min, respecting the desired production planning horizon of 1440 min.

7 Conclusions

This work deals with a production scheduling problem to fed an arc furnace in a mining-metallurgical industry. The objective of the schedule is to minimize the makespan and the total tardiness. To solve it, we developed a matheuristic algorithm called DSS_ASPT. First, we choose the products through a heuristic dispatch rule. According to this rule, we select the products with the shortest processing times among those whose due dates correspond to the work shift analyzed. Second, the proposed algorithm uses a goal linear programming formulation to solve the blending problem of raw materials, a sub-problem of the sequencing.

We use seven instances to test the proposed algorithm. These instances contain 30, 90, 210, and 300 products to be produced during the planning horizon in three daily work shifts. Four of them have 10 products per work shift, and three have 9 products. In these instances, the production planning horizons can be 1, 3, 7, and 10 days.

The results showed that the DSS_ASPT algorithm generates all the requested products, but the makespan can exceed the planning horizon, and there may be a delay in production. We can observe that the planning horizon is respected in instances with the least number of products per work shift. This result can be indicative of the ideal number of products per work shift in the industry under study. Regarding total tardiness, as the number of products to be generated grows, the delay in their production also increases. Regarding the quality parameters of the products, it is clear that the blending problem was successfully solved. The deviations from the quality parameters with the highest weights were minimal, and those from the other parameters were not high.

In addition to sequencing all required products, respecting production specifications, and reducing production delay, the developed algorithm has a low runtime. In this way, we validate its use to support decision-making in the company.

In the method proposed, a product is chosen to be inserted into the solution greedily at each step. The advantage of this type of algorithm is that a solution is found quickly and, in general, has good quality. On the other hand, the disadvantage is that greedy strategies do not always provide the best solutions to a problem. In this sense, we propose as future works to develop metaheuristic-based algorithms to generate even better solutions, such as the Iterated Local Search and Genetic Algorithm. In addition, we propose to evaluate the proposed method in instances of other similar industries and adapt it to handle other production processes that use, for example, more than one conveyor belt to transport the raw materials.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

Araujo, S. A., Arenales, M. N., & Clark, A. R. (2008). Lot sizing and furnace scheduling in small foundries. *Computers & Operations Research, 35*(3), 916–932. https://doi.org/10.1016/j.cor.2006.05.010.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*. New York: Wiley.

Baykasoğlu, A., & Ozsoydan, F. B. (2018). Dynamic scheduling of parallel heat treatment furnaces: A case study at a manufacturing system. *Journal of Manufacturing Systems, 46,* 152–162. https://doi.org/10.1016/j.jmsy.2017.12.005.

Bayu, F., Panda, D., Shaik, M. A., & Ramteke, M. (2020). Scheduling of gasoline blending and distribution using graphical genetic algorithm. *Computers & Chemical Engineering, 133*(106), 636. https://doi.org/10.1016/j.compchemeng.2019.106636.

Chanda, E. K. C., & Dagdelen, K. (1995). Optimal blending of mine production using goal programming and interactive graphics systems. *International Journal of Surface Mining, Reclamation and Environment, 9*(4), 203–208. https://doi.org/10.1080/09208119508964748.

Costa, F. P., Souza, M. J. F., & Pinto, L. R. (2005). Um modelo de programação matemática para alocação estática de caminhões visando ao atendimento de metas de produção e qualidade. *Revista Escola de Minas, 58*(1), 77–81.

Cota, L. P., Haddad, M. N., Souza, M. J. F., & Coelho, V. N. (2014). Airp: A heuristic algorithm for solving the unrelated parallel machine scheduling problem. In *2014 IEEE congress on evolutionary computation (CEC)*, pp. 1855–1862. https://doi.org/10.1109/CEC.2014.6900245.

Cota, L. P., Guimarães, F. G., Ribeiro, R. G., Meneghini, I. R., Oliveira, F. B., Souza, M. J. F., & Siarry, P. (2019). An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem. *Swarm and Evolutionary Computation, 51*(100), 601. https://doi.org/10.1016/j.swevo.2019.100601.

Franzoi, R. E., Menezes, B. C., Kelly, J. D., & Gut, J. A. (2019). Design for online process and blend scheduling optimization. In S. G. Muñoz, C. D. Laird, & M. J. Realff (Eds.), *Proceedings of the 9th international conference on foundations of computer-aided process design, computer aided chemical engineering*, Vol. 47, Elsevier, pp. 187–192. https://doi.org/10.1016/B978-0-12-818597-1.50030-8.

Gomes, A. C. L., Ravetti, M. G., & Carrano, E. G. (2021). Multi-objective matheuristic for minimization of total tardiness and energy costs in a steel industry heat treatment line. *Computers & Industrial Engineering, 151*(106), 929. https://doi.org/10.1016/j.cie.2020.106929.

Haddad, M. N., Cota, L. P., Souza, M. J. F., & Maculan, N. (2015). Solving the unrelated parallel machine scheduling problem with setup times by efficient algorithms based on iterated local search. In J. Cordeiro, S. Hammoudi, L. Maciaszek, O. Camp, & J. Filipe (Eds.), *Enterprise information systems*, Springer, Cham, pp. 131–148. https://doi.org/10.1007/978-3-319-22348-3_8.

Hernández, J. D., Onofri, L., & Engell, S. (2020). Optimization of the electric efficiency of the electric steel making process. *IFAC-PapersOnLine, 53*(2), 11895–11900. https://doi.org/10.1016/j.ifacol.2020.12.707.

Huang, X., Fan, X., Chen, X., Gan, M., Ji, Z., & Zheng, R. (2019). A novel blending principle and optimization model for low-carbon and low-cost sintering in ironmaking process. *Powder Technology, 355,* 629–636. https://doi.org/10.1016/j.powtec.2019.07.085.

Lin, F., Zhizhong, M., Yuan, P., & Fuqiang, Y. (2012). An improved multi-objective particle swarm optimization algorithm and its application in eaf steelmaking process. In *2012 24th Chinese control and decision conference (CCDC)*, pp. 867–871. https://doi.org/10.1109/CCDC.2012.6244134.

Lingshuang, K., Chunhua, Y., Shenping, X., & Gang, C. (2013). Stochastic optimization method based on HSS technique and expert knowledge for a metallurgical blending process. In *2013 Third international conference on intelligent system design and engineering applications*, pp. 1290–1293. https://doi.org/10.1109/ISDEA.2012.304.

Liu, S., & Zaraté, P. (2014). Knowledge based decision support systems: A survey on technologies and application domains. In *Group decision and negotiation. A process-oriented view*, Springer, Cham, pp. 62–72.

Moraes, E. F., Alves, J. M. C. B., Souza, M. J. F., Cabral, I. E., & Martins, A. X. (2006). Um modelo de programação matemática para otimizar a composição de lotes de minério de ferro da mina cauê da cvrd. *Revista da Escola de Minas, 59,* 299–306.

Olivier, L. E., & Craig, I. K. (2017). A survey on the degree of automation in the mineral processing industry. In *2017 IEEE AFRICON*, pp. 404–409. https://doi.org/10.1109/AFRCON.2017.8095516.

Pereira, C. S., Dias, D. M., Vellasco, M. M. B. R., Viana, F. H. F., & Martí, L. (2018). Crude oil refinery scheduling: Addressing a real-world multiobjective problem through genetic programming and dominance-based approaches. In *Proceedings of the genetic and evolutionary computation conference companion*, Association for Computing Machinery, New York, NY, USA, GECCO '18, pp. 1821–1828. https://doi.org/10.1145/3205651.3208291.

Pereira, C. S., Dias, D. M., Pacheco, M. A., Vellasco, M. M. B. R., Abs da Cruz, A. V., & Hollmann, E. H. (2020). Quantum-inspired genetic programming algorithm for the crude oil scheduling of a real-world refinery. *IEEE Systems Journal, 14*(3), 3926–3937. https://doi.org/10.1109/JSYST.2020.2968039.

Pinedo, M. L. (2012). *Scheduling: Theory, algorithms, and systems* (5th ed.). New York: Springer.

Rezakhah, M., Moreno, E., & Newman, A. (2020). Practical performance of an open pit mine scheduling model considering blending and stockpiling. *Computers & Operations Research, 115*(104), 638. https://doi.org/10.1016/j.cor.2019.02.001.

Ribeiro, C. C., Maniezzo, V., Stützle, T., Blum, C., Juan, A. A., Ramalhinho, H., et al. (2020). Preface to the special issue on matheuristics and metaheuristics. *International Transactions in Operational Research, 27*(1), 5–8.

Saberi-Aliabad, H., Reisi-Nafchi, M., & Moslehi, G. (2020). Energy-efficient scheduling in an unrelated parallel-machine environment under time-of-use electricity tariffs. *Journal of Cleaner Production, 249*(119), 393. https://doi.org/10.1016/j.jclepro.2019.119393.

Saboohi, Y., Fathi, A., Skrjanc, I., & Logar, V. (2019). Optimization of the electric arc furnace process. *IEEE Transactions on Industrial Electronics, 66,* 8030–8039. https://doi.org/10.1109/TIE.2018.2883247.

Shyamal, S., & Swartz, C. L. (2017). Optimization-based online decision support tool for electric arc furnace operation. *IFAC-PapersOnLine, 50*(1), 10784–10789. https://doi.org/10.1016/j.ifacol.2017.08.2338.

Wills, B. A., & Finch, J. A. (2015). *Wills mineral processing technology an introduction to the practical aspects of ore treatment and mineral recovery* (8th ed.). Waltham, MA: Elsevier.

Worldsteel. (2020). Steel statistical yearbook 2020: concise version.

Xu, T. J., Yang, P., & Liu, Z. Q. (2008). Mine ore blending planning and management based on the fuzzy multi-objective optimization algorithm. In *2008 International seminar on business and information management*, Vol. 2, pp. 38–41. https://doi.org/10.1109/ISBIM.2008.123.

Yuan, Y., Qu, Q., Chen, L., & Wu, M. (2020). Modeling and optimization of coal blending and coking costs using coal petrography. *Information Sciences, 522,* 49–68. https://doi.org/10.1016/j.ins.2020.02.072.

Zhang, D., Gao, X., Wang, M., Ye, Y., & Tong, J. (2019). The research of sintering ore blending based on profit maximization. In *2019 Chinese control and decision conference (CCDC)*, pp. 2220–2225. https://doi.org/10.1109/CCDC.2019.8833379.