# Optimizing Supply Chain Scheduling in Steel Mills: An Algorithm Leveraging Digital Twin Technology

Authors

Pedro Henrique Feres Campos, Digital Twin Specialist, Enacom, Belo Horizonte, MG, Brazil
pedro.campos@enacom.com.br

Fabricio Schiavon Kolberg, Digital Twin Developer, Enacom, Belo Horizonte, MG, Brazil

Matheus Henrique Lemes Faria, Digital Twin Specialist, Enacom, Belo Horizonte, MG, Brazil

Matheus de Oliveira Mendonça, Digital Twin and Optimization Manager, Enacom, Belo Horizonte, MG, Brazil
matheus.mendonca@enacom.com.br

Geraldo José Duarte, S&OP Consultant, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil

Milton Carlos Abel Pires, Supply Chain Coordinator, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil

Wilian Lopes Santos, S&OP Specialist, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil

Luiz Fabio Lobato Notini, S&OP Specialist, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil

Bruno Alvares, Supply Chain Manager, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil

Bruno da Silva Breder, Industry 4.0 Technical Manager, Gerdau, Fazenda do Cadete, Ouro Branco, MG, Brazil
bruno.breder@gerdau.com.br

This article presents a heuristic algorithm developed within a digital twin system to optimize scheduling in a complex steelmaking plant. The digital twin provides insights and anticipates bottlenecks, contributing to efficient scheduling. The algorithm addresses nonlinear constraints arising from technical, technological and financial limitations, ensuring a customized approach to production planning. The optimized scheduling aims to enhance steel production, minimize delays and maximize resource allocation within the supply chain. By leveraging mathematical optimization and digital twin technology, this solution achieved significant gains in a real steel mill, increasing monthly production by 1% without the need for infrastructure investment.

## Introduction

Optimizing production in a steel mill via operation scheduling is a challenging problem, and it has been approached in many ways in recent years.[1] An adequate solution can bring significant improvements to a mill's efficiency with no extra costs to infrastructure. This article presents a solution to the scheduling problem for Gerdau's steel mill in the Brazilian city of Ouro Branco.[2]

The scheduling problem that is addressed here is known as the steelmaking and continuous casting (SCC) problem,[3] which is modeled as a variation of the optimization problem known as the hybrid flow shop problem (HFSP).[4]

The steelmaking and casting process consists of three main steps: Steelmaking, refining and casting.[3] Each of these stages may have several machines working in tandem. For the Gerdau mill this work is based on, the following features are present:

i. Three casting machines for slabs, blooms and billets, each with their own processing times.

ii. Continuous production and intermittent production coexist. The first two steps (steelmaking and refining) have the individual charges as the minimum production unit (the "jobs" in the HFSP model). Casting, on the other hand, can be done continuously, in sequences of charges, provided the consecutive charges meet the criteria to be grouped together (similar chemical compositions, same dimensions, etc). These groups of continuously cast charges are referred to as batches, or sequences.

iii. Charges can have different refining routes, even when they are in the same casting sequence.

iv. A setup time is required between two sequences in the same casting machine, to prepare them for the next sequence.

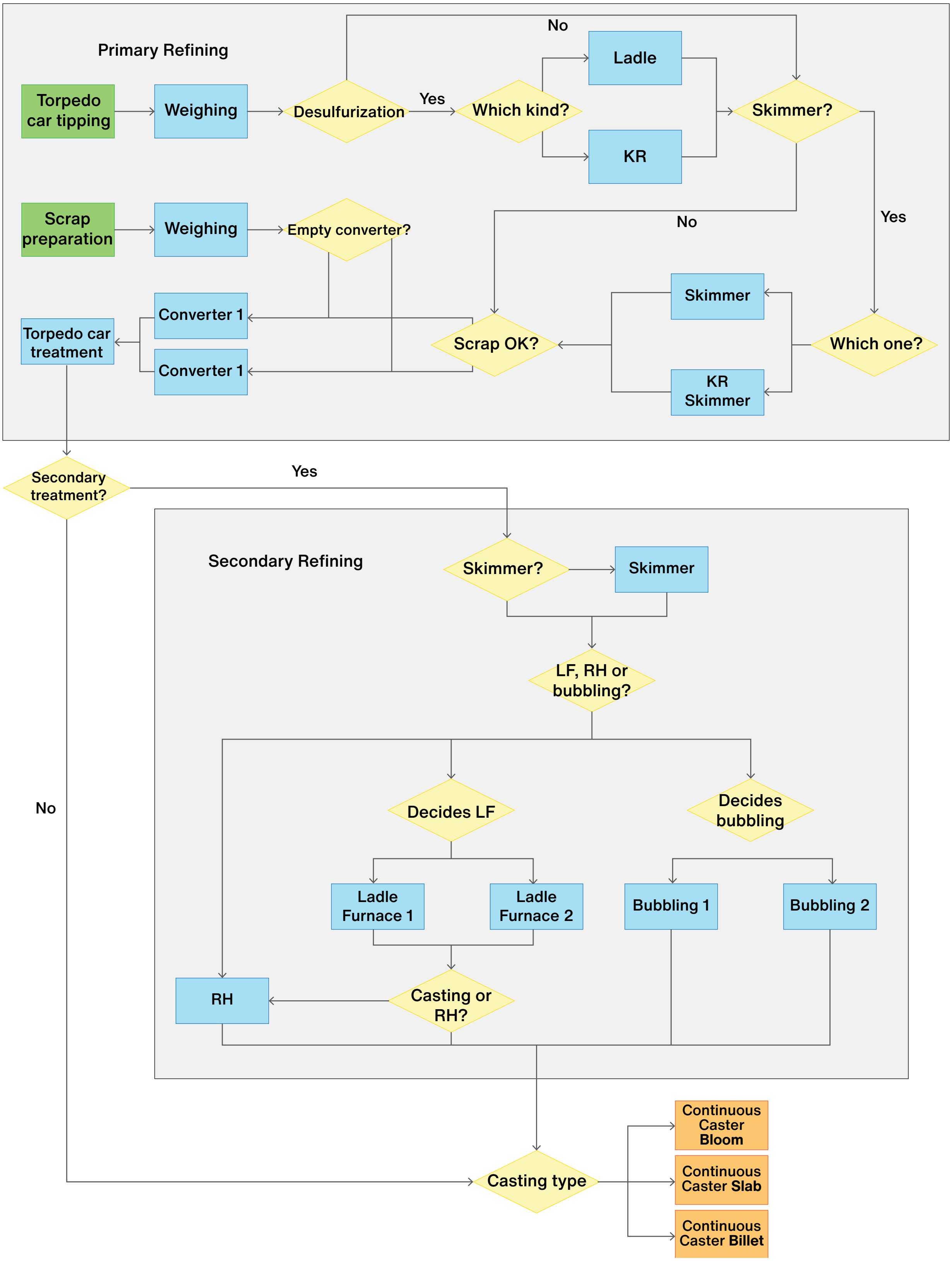
v. Charges have a set due date, and it is imperative that they do not suffer from excessive delays.

Those features make the Gerdau steel mill a particularly complex instance of the HFSP problem. Fig. 1 contains a flowchart describing the routes of steel products within the mill.

Our solution to the problem employs digital twin technology.[5]

Figure 1

Steelmaking and continuous casting process for the Gerdau Ouro Branco plant.



Digital twins are highly beneficial to industrial optimization in general[6] due to their ability to fully model a production line, and to output a precise step-by-step simulation of a complex industrial process, and they are employed in several different aspects of steel production.[7] For this application in particular, the digital twin is fundamental for identifying production bottlenecks and other undesired events to be solved by the heuristic algorithm.

This work explains the methods used in the scheduler, presents data from its execution over real-world inputs, and then concludes with a detailed discussion of the data, the findings they elucidate, and how the algorithm affects the performance of the Gerdau steel plant.

## Discussion

### Methodology

The algorithm for the optimization of the SCC problem is divided into two main subproblems, as seen in the flowchart in Fig. 2. The primary problem assembles charges into sequences (batches) and sorts the list of sequences to define their scheduling order. The secondary problem schedules the sequences into a timeline, in the order given by the primary problem, taking every machine that every charge must go through into consideration, and defining the beginning and end time for every operation.

The primary objective function is minimizing the pig iron spill. If the converters stay idle for too long during high steel production periods, hot pig iron needs to be spilled, which leads to a waste of raw material. Furthermore, since the blast furnace is the main bottleneck, spillage also leads to a longer total time to process all charges (makespan). The secondary objective is to minimize the delays of charges, i.e., finishing each charge as close to their due date as possible.

**Primary Problem:** As seen in Fig. 2, the primary problem is broken down into two stages, the first one being the "constructor" stage. In this stage, the charges are organized into batches (sequences), defining the jobs to be made within the same tundish. This process must take the following restrictions into account:

- Casting type and gauge: Batches must contain charges of the same casting type (bloom, billet, slab) and same gauge.
- Mix steel: There are rules determining which pairs of steel types can be included consecutively in the same batch.
- Minimum and maximum batch casting times: There are upper and lower bounds for the duration of continuous casting for a whole batch, depending on casting type.
- Minimum and maximum charge count: There are upper and lower bounds for the number of charges in the same batch, depending on casting type.
- Maximum width difference: There is an upper bound to the maximum difference of width between two charges in the same batch.
- Trapezoidal rule: The order of the charges within a batch must be such that its widths form a roughly trapezoidal shape. This is to avoid wear and tear on the gauge pieces.

Every sequence (batch) is sent into the caster within one tundish, and there exists a setup time between tundishes. The constructor aims to create the largest possible batches given the above restrictions, to minimize tundish changes and, therefore, make the makespan shorter and increase productivity.

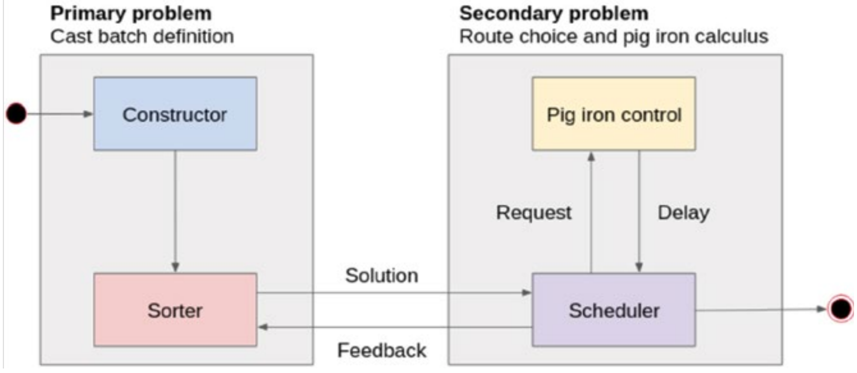
Once the batches are constructed, the "sorter" component orders them according to a series of different heuristics. Another important role of the sorter is to change the batch order given feedback from the secondary problem, seen in the following subsection.

**Secondary Problem:** The secondary problem is responsible for, in simple terms, scheduling the sequences (batches) in the order given by the primary problem, and simulating their processing in the timeline, returning feedback to the feedback loop (see the subsection titled Feedback Loop). The secondary problem must treat the following restrictions:

- Continuous casting: Charges from the same batch must go through the casting machines continuously, without delay between them.

Figure 2

The algorithm's architecture.



- Process times: Each machine has different processing times depending on the steel grade of the charge.
- Possible routes: For each charge, depending on its steel grade, there is a set of possible refining routes it can take before casting.
- Machine maintenance: The problem input may contain a list of maintenance times for certain machines in the steel plant, during which no operations using the machine can be done.
- Transit times: There is a list of minimum and maximum transit times between consecutive machines in the same casting route, taking maximum buffer times into consideration.
- Tundish change times: At the end of the casting of the charges in a tundish, there is a minimum waiting time before the next tundish can be processed.
- Gauge change times: If two consecutive batches of the same casting type are such that their gauges are different, there must be a period between them for the gauges to be changed in the machine.
- Pig iron limits: Pig iron is constantly produced during the operation of the steel mill, at varying rates, and is used by every converter operation (see Fig. 1). Pig iron spills occur to prevent the operational maximum from being violated. As for operational minimum violations, they are prevented by delaying converter operations.

***Scheduler Block —*** The scheduler block is responsible for allocating the sequences (batches) into the steel mill's timeline. It allocates sequences in the order given by the sorter, such that the casting of a sequence must always occur after the casting of the previous sequence of the same casting type. The allocation process begins by attempting to allocate a sequence's casting operation, and then works backwards to fit all of the sequence's individual charges' refining steps and converter operations before the casting. If an impossibility is found, such as:

- Pig iron minimum violation (converter steps reduce the pig iron amount in the converters),
- Unavailable machines (maintenance or conflicts with other operations),
- Gauge change period (current sequence must be allocated later to accommodate gauge change) or
- Transit times are not respected (the time between two refining steps for a charge is too long),

then the casting operation is pushed forward in time and the backward fitting of the refining steps starts over. The algorithm uses several different heuristics to ensure the final allocation of a sequence is feasible, and the earliest possible without breaking restrictions.

***Pig Iron Control —*** The pig iron control block is responsible for simulating the evolution of the pig iron levels throughout the steel mill's process, dealing with pig iron production, usage of pig iron in converters, and minimum and maximum pig iron level constraints. From the scheduler block, the exact starting point of every converter operation is received, and the pig iron level is reduced immediately (it is assumed the pig iron dumping time is negligible). The amount of pig iron to be spilled in a converter operation at a given time is given by Eq. 1:

$$\nu^- = \frac{\lambda \rho_{hmr}}{\eta_s \eta_p} \qquad \text{(Eq. 1)}$$

where

$\nu^-$ = the total pig iron used for the converter operation,

$\lambda$ = the volume of steel in one ladle,

$\rho_{hmr}$ = the current HMR (hot metal rate, the ratio between pig iron and scrap iron used in the operation) and

$\eta_s$ and $\eta_p$ = the steel efficiency and the pig iron efficiency, respectively (both are a number from 0 to 1).

All values are informed as parameters in the problem input, sometimes given alongside exception intervals as the values can change in certain periods.

As previously stated, that amount is discounted immediately at the start of a converter operation. What that implies is the following: let $\nu(t)$ be the current pig iron volume at time $t$ and let $\theta$ be the starting point of a converter operation. Then:

$$\nu(\theta + \varepsilon) = \nu(\theta) - \nu^- \qquad \text{(Eq. 2)}$$

where $\varepsilon$ is the smallest unit of time in the simulation (a minute, in this case).

The pig iron control block also models the minute-by-minute pig iron production coming from the blast furnace. Let $\nu^+(t)$ be the amount of pig iron produced in one minute starting at time $t$. This value is passed as a parameter in the input, once again with possible exception intervals, in case of blast furnace maintenance or low production periods. Eq. 2 can then be further refined as follows:

$$\nu(\theta + \varepsilon) = \nu(\theta) - \delta(\theta)\nu^- + \nu^+(\theta) \qquad \text{(Eq. 3)}$$

In Eq. 3, $\delta(t)$ is 1 if there is a converter operation starting at $t$, and 0 otherwise. Finally, the pig iron control must also control pig iron spill events, which occur when the operational maximum is violated. Let $\gamma$ be the size of an individual torpedo car, and $M$ be the operational maximum pig iron level. A spill event must remove a discrete amount of torpedo cars from the total volume, so every spill is of the form $k\gamma$ for some positive integer value of $k$. Let $\theta$ be a minute such that $\nu(\theta + \varepsilon)$ contains a pig iron maximum violation. A simplified view of the pig iron spillage model is as follows:

$$\nu'(\theta+\varepsilon) = \nu(\theta) - \delta(\theta)\nu^{-} + \nu^{+}(\theta)$$
$$\nu(\theta+\varepsilon) = \nu'(\theta+\varepsilon) - \left\lceil \frac{\nu'(\theta+\varepsilon) - M}{\gamma} \right\rceil \gamma \qquad \text{(Eq. 4)}$$

In Eq. 4, the $\lceil\ \rceil$ brackets denote the ceiling operation. The actual treatment of pig iron spilling has a small number of extra details, but they are unimportant for the purposes of this article.

One other thing the pig iron control does is interact with the scheduler to avoid minimum pig iron level violations. After the allocation of every sequence, the pig iron control is run, and if a minimum violation is detected, the allocation is redone.

Fig. 3 shows an example of a pig iron level graph, with converter operations and pig iron spills highlighted.

**Feedback Loop:** The feedback loop is a fundamental aspect of the algorithm, as it allows the algorithm to iteratively improve its solution. Every time a sorter execution is finished, the new batch order is sent to the secondary problem, and the secondary problem, once finishing the simulation of the new order, returns a feedback list to the sorter. The sorter then uses the list to attempt to improve upon the previous solution, and the loop starts anew.

The loop goes on for a number of iterations, and the best solution found is returned for analysis by the supply chain specialist at the end. The number of iterations is calculated based on heuristics that consider several factors, from the size of the input to the total expected running time.

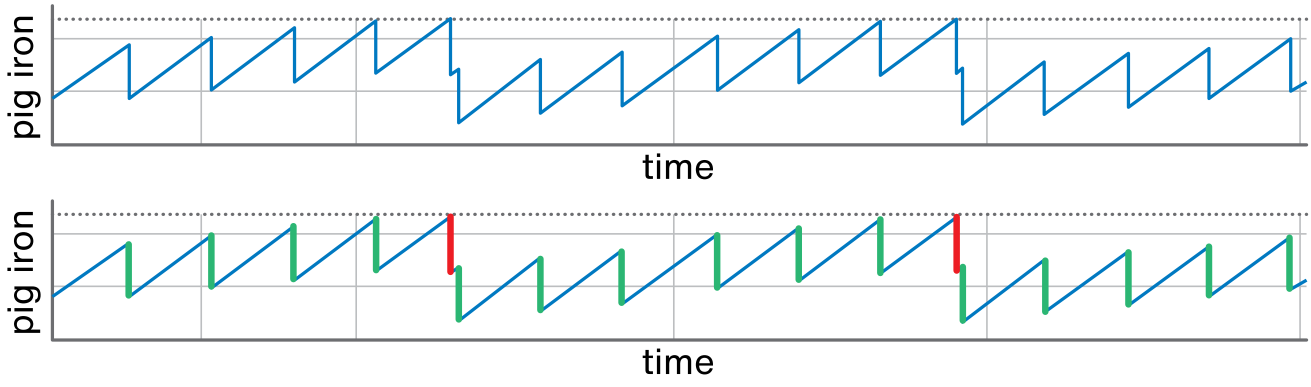
***Feedback and Events —*** As previously stated, the secondary problem returns a feedback list to the primary problem. The list contains, among other things, the pig iron time series, the exact starting times of all operations, and a list of undesirable events to be treated by the sorter. These events include, but are not limited to:

- Critical delay: When charges are finished a very long time after their due dates.
- Bad occupation: Situations where the converter occupation is suboptimal, resulting in long idle periods.
- Pig iron spill: A series of events pertaining to potentially avoidable pig iron spillage, including:
  - Converter unavailability: Situations where one converter is unavailable, and the other is idle for too long.
  - Casting unavailability: A casting machine is unavailable, leading to an idle period that can be prevented by pulling sequences from other casting types to the time period of the event.
  - Gauge changes: Two consecutive sequences of the same casting type are of different gauges, resulting in a long idle period for the casting machine. Sequences of other casting types must be pulled to fill the gap.
  - Secondary refinement unavailability: One or more secondary refinement machines are unavailable. The event can be corrected by attempting to fill the time gap with sequences that do not require the unavailable machines.

Every event accompanies a list of sequences affected by the event, as well as the time interval in which it occurs.

Figure 3

**Top: a graph showing the evolution of pig iron volume in a given period. Bottom: the same graph, with pig iron spills highlighted in red, and converter operations highlighted in green. The pig iron maximum is marked by a dotted line.**



This information is used by the sorter to decide how the list of sequences must be changed for the next iteration.

***Event Treatment —*** Once the feedback is received, the primary sorter will attempt to solve a specific event by changing the order of the sequence list. To avoid undoing the corrections done in earlier events, the sorter will only change the list from the earliest sequence affected by the event onwards.

Once the reordering is done, the new solution is sent to the secondary problem. One other piece of information that can be extracted from the feedback is whether the event was resolved. In case the new solution fails to solve the event, the sorter attempts to solve it again a limited number of times. If all attempts fail, the current event is considered irrelevant and the next event is attempted.

**Solution Output:** Once the algorithm concludes, it returns the best solution obtained, including information about each batch, the order in which they are processed, and a detailed simulation of the process, including the starts and ends of every single machine operation, and a minute-by-minute time series showing the pig iron levels. The information can then be used to plot a Gantt chart detailing the machine operations on the timeline, as seen in Fig. 4.

## Results

**Comparison With Brute Force:** This subsection presents a comparison between the algorithm and a brute force method that considers all possible batch permutations. Given that the number of possible permutations is factorial over the number of batches, a brute

Figure 4

**An example of a Gantt chart plotted with the solution to a real-world problem.**

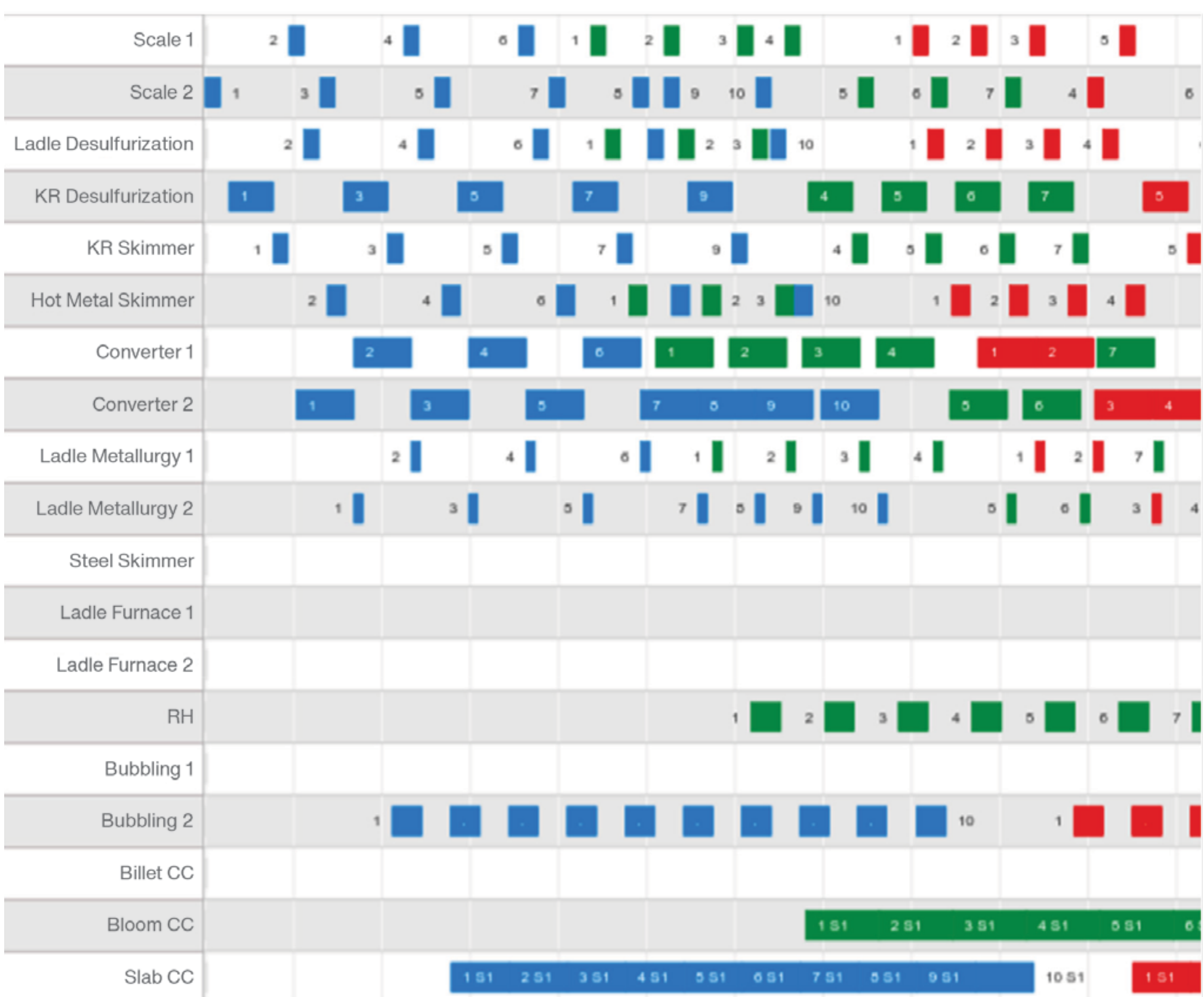
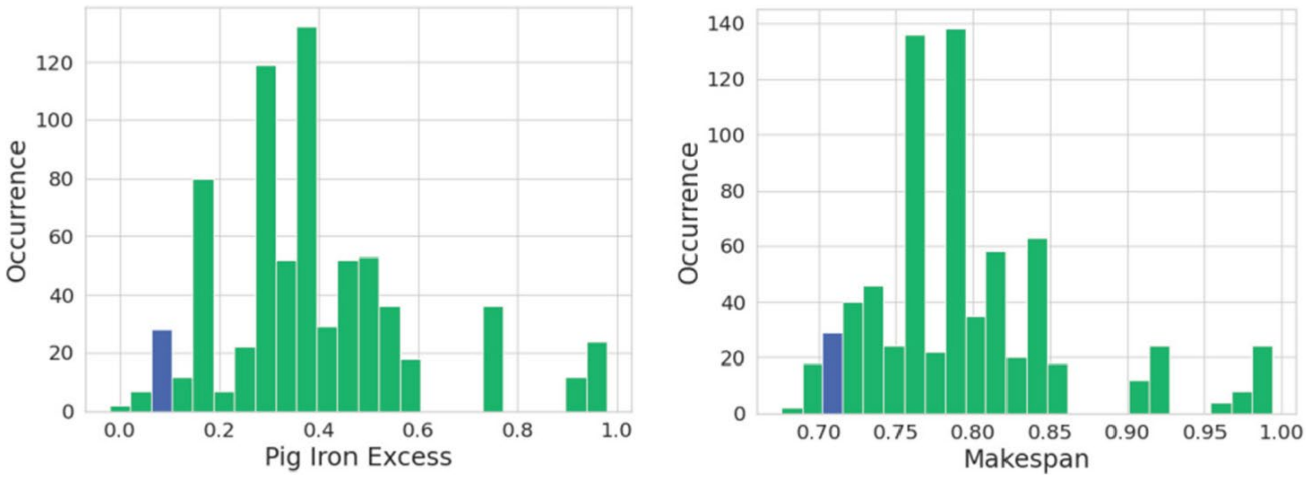


Figure 5

**Histogram containing the normalized values of pig iron excess and makespan for all permutations of the batches. The columns highlighted in blue are the result obtained by the heuristic algorithm.**




force approach is not advisable for this problem and is being used exclusively to demonstrate the algorithm's performance.

The scenario used for the comparison is comprised of six batches (three for slabs, two for blooms and one for billets) with a converter unavailability in it. The brute force method was used to determine the properties of all 720 (6!) permutations and compare them to the result given by the algorithm. Fig. 5 is a histogram showcasing the results.

The algorithm reaches its final solution after analyzing only three permutations, which shows its capability to obtain good results in a relatively short amount of time.

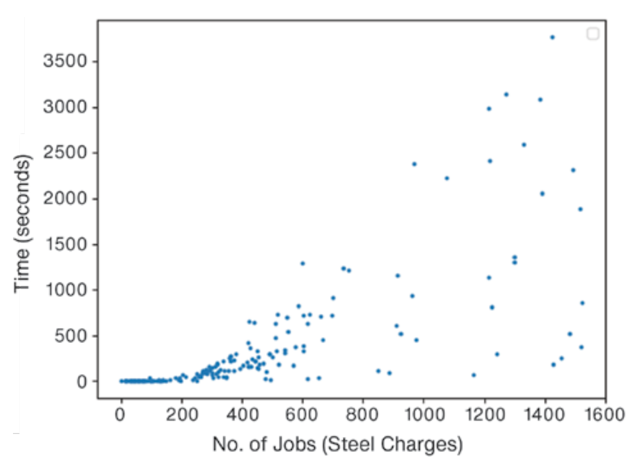
As the number of batches increases, the time differences between the algorithm and a brute force approach greatly increase. It is not uncommon for real-life scenarios to have over 100 batches. Applying the algorithm allows the steel mill operators to find a solution that is close to optimal without requiring impractically long computation times.

**Execution Times:** To test the expected time performance of the algorithm, a set of 460 real and realistic test cases of varying sizes were used, and it was checked how the execution time grows in relation to the number of jobs (i.e. number of steel charges). The results can be seen in Fig. 6. The tests were executed in dedicated AWS machines allocated specifically for usage by the steel mill operators.

As seen in the figure, even for test cases with over 1,000 steel charges, the algorithm runs in a little over an hour, making it significantly more efficient than a brute force approach would be. For comparison, a very conservative estimate for the brute force method on as little as 200 jobs puts it at over 1,000 hours (about 1.5 months).

Figure 6

**Relation between the number of jobs and execution time (in seconds) of the algorithm.**




## Conclusion

The algorithm presented here leverages digital twin technology to provide an efficient solution to the steel mill scheduling problem applied to the Gerdau plant in Ouro Branco. The solutions it provides include detailed instructions about how to order the charges to be processed, and when to allocate each necessary machine operation, without breaking the restrictions of the problem.

As seen in the results section, the algorithm runs at reasonable times, even for large inputs, and the solutions returned are of good quality. It is, therefore, a useful tool to aid in the decision-making process of the steel mill. Practical usage by steel mill specialists has brought improvements to the mill's productivity without extra infrastructure costs.

Future works involve an intelligent integration with other parts of the company, especially considering rolling mills that are supplied with the steel ingots produced in the steel mill — for example, which steel to prioritize considering coffin and campaign restrictions from the rolling mill. This aims to provide better supply chain control as a whole.

*This article is available online at AIST.org for 30 days following publication.*

## References


1. D.G. Menéndez, H.M. Palacios, F.O. Fernández and M.D. Piloñeta, "Scheduling in Continuous Steelmaking Casting: A Systematic Review," *ISIJ International,* Vol. 60, No. 6, 2020, pp. 1097–1107, https://doi.org/10.2355/isijinternational.ISIJINT-2019-574.
2. C.V.D. Carvalho, E.N. Almeida, T.H.N. Coelho, R.C.A. Filho and J.A.C. Cohn, "Redução da Perda Metálica na Aciaria da Gerdau de Ouro Branco," *ABM 2017 — Anais do Seminário de Aciaria, Fundição e Metalurgia de Não-ferrosos,* Vol. 48, No. 48, 2017, pp. 390–401, https://doi.org/10.5151/1982-9345-30387.
3. J. Yang, B. Wang, M. Guan, T. Li, S. Gao, W. Guo and Q. Liu, "Scheduling Model for the Practical Steelmaking-Continuous Casting Production and Heuristic Algorithm Based on the Optimization of 'Furnace-Caster Matching' Mode," *ISIJ International,* Vol. 60, No. 6, 2020, pp. 1213–1224, https://doi.org/10.2355/isijinternational.ISIJINT-2019-423.
4. R. Ruiz and J.A.V. Rodríguez, "The Hybrid Flow Shop Scheduling Problem," *European Journal of Operational Research,* Vol. 205, No. 1, August 2010, pp. 1–18, https://doi.org/10.1016/j.ejor.2009.09.024.
5. A. Sharma, E. Kosasih, J. Zhang, A. Brintrup and A. Calinescu, "Digital Twins: State of the Art Theory and Practice, Challenges, and Open Research Questions," *Journal of Industrial Information Integration,* Vol. 30, 100383, November 2022, https://doi.org/10.1016/j.jii.2022.100383.
6. M. Javaid, A. Haleem and R. Suman, "Digital Twin Applications Toward Industry 4.0: A Review," *Cognitive Robotics,* Vol. 3, 2023, pp. 71–92, https://doi.org/10.1016/j.cogr.2023.04.003.
7. Y. Zhang, M. Sukhram, I. Cameron and A. Rozo, "Industrial Perspective of Digital Twin Development and Applications for Iron and Steel Processes," *AISTech 2020 Conference Proceedings,* 2020, pp. 34–43. ✦


This paper was presented at AISTech 2024 — The Iron & Steel Technology Conference and Exposition, Columbus, Ohio, USA, and published in the AISTech 2024 Conference Proceedings.


CARB-RITE Company

**Serving the Steel Industry for Over 85 years**

**The Rite Way to Carburize Steel**

The Best in Recarburizers for Electric Furnaces, BOF's and Iron and Steel Foundries

**Quality**
**Service**
**Dependability**

Specializing in a Complete Line of Carbon Products:
Metallurgical Coke
Petroleum Coke
Anthracite Coal
Customized Screening, Blending and Packaging

Affiliate:
**Bognar and Company, Inc.**
**Pittsburgh, PA**

**(412) 344-9900**
**1-800-333-CARB**

**sales@ejbognar.com**