# Modeling and Solving the Steelmaking and Casting Planning and Scheduling Problem

**Davide Armellini · Paolo Borzone · Sara Ceschia · Luca Di Gaspero · Andrea Schaerf**

## 1 Introduction

One of the most complex production operations is the steelmaking and casting process. Indeed, many technological (physical, chemical, mechanical, . . . ) and business constraints are involved in the process and the size of the production batches makes the operation of a steelmaking plant quite costly. Consequently, a careful planning and scheduling of the plant with the aim of maximizing the daily/weekly throughput is of crucial importance to ensure productivity and competitiveness.

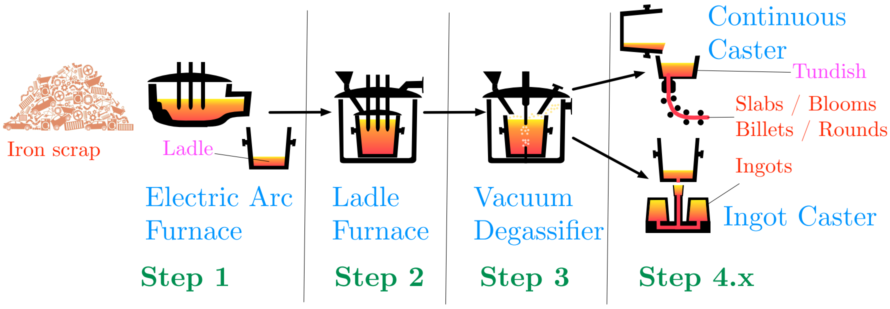


**Fig. 1** The steelmaking and casting production process.

A typical steelmaking and casting production plant is composed of several *lines*. A single line is outlined in Figure 1 and consists of four stages. At first the iron scrap

Davide Armellini and Paolo Borzone
Danieli Automation S.p.A.
via Bonaldo Stringher 4, I-33042 Buttrio (UD), Italy / p. Borgo Pila 39, I-16129 Genoa, Italy
E-mail: d.armellini@dca.it, p.borzone@dca.it
Sara Ceschia, Luca Di Gaspero, Andrea Schaerf
DPIA – Università degli Studi di Udine
via delle Scienze 206, I-33100 Udine, Italy
E-mail: sara.ceschia@uniud.it, luca.digaspero@uniud.it, andrea.schaerf@uniud.it

is melted in an *Electric Arc Furnace* (EAF), then the liquid metal is poured into a *ladle* that will be used to contain the steel in all the following processing steps. The next step is performed in the *Ladle Furnace* (LF), where chemical additives are added to the iron for obtaining the desired alloy. Afterwards, the gas content of the metal has to be reduced in order to avoid explosions during the casting process in a *Vacuum Degassifier* (VD) unit.

The last production step, called *casting*, might differ based on the type of semifinished product required. Indeed, in the case of "long pieces" (i.e., slabs, rounds, blooms or billets) the production is sent to a *Continuous Casting* (CC) machine, whereas "short pieces" (e.g., ingots) are forged by an *Ingot Casting* (IC) machine. These semifinished products, then, could be subject to further processing (e.g., hot rolling), which might be planned and scheduled somehow independently from the casting one.

The main physical constraint of this initial part of the steelmaking process is the practical impossibility to buffer jobs between different processing steps because of the cooling of the liquid metal in case of waiting. As a consequence, the jobs should be scheduled in a *just-in-time* fashion.

A similar version of the problem was presented by Fanti *et al* [2], who propose an integrated system consisting of a database, an optimization engine, a simulation module and an user interface for the pure scheduling problem. The optimization engine models the scheduling as a hybrid flow shop using a MILP solver.

In our previous work [1], in order to better capture our real-world specification we made the following modifications, with respect to [2]:

1. we considered the possibility that a job switches from one line to another at any stage;
2. we considered the *border data*, coming from the previous scheduling stage (the plant runs for 24h a day); in particular, for each machine, we register the time when it is available, the section of last job and its steel grade; in addition, we consider the status of the ladle with respect to pollutants;
3. we used the *throughput* as objective function, rather than the makespan, as we schedule for a fixed horizon, but flexible number of jobs.

In this work, in order to obtain a model that captures the essential features of the problem and can be shared with other researchers for comparison purposes, we decided to make further modifications. In essence, on the one side, we simplify the problem statement, so as to remove low level details specific for the situation at hand; on the other side, we generalize the problem including features collected from the literature that where not included in the models of [2] and [1].

In detail, the main differences with respect to our previous model in [1] are:

4. we consider possible standstills of single machines for limited times (e.g., for maintenance); as a consequence we have to resort heavily to the possibility that the production is not executed by a single line, but goes through machines belonging to different lines (at a price of longer moving times and possible bottlenecks);
5. we remove the detailed management of ingot wagons and stripping areas for the ingot casting machines, as the number of ingot production is normally limited and it does not have a strong impact on the overall performance;
6. the condition to perform a shorter setup (called *fly-tundish*) for CC machines has been modified; the feasibility of this procedure technically involves the heating and moving times of the tundish, which is an intermediate container that is needed to

**Table 1** Instance features.

| Feature | Description | Min | Max |
|---|---|---|---|
| jobs | single charge of melted metal | 110 | 308 |
| steel grades | chemical composition of the steel of the job (influences the setup time) | 39 | 98 |
| ladles | container of the molten steel moved through the plant till the end of the process | 10 | 10 |
| appointments | time window for the completion time of a specific job | 0 | 34 |
| standstills | machine stop due to maintenance services or temporary machine breakdown | 0 | 3 |
| pollutants | chemical element present in the fluid metal that pollutes the ladle | 2 | 2 |
| horizon | planning period (in minutes) | 720 | 2880 |

pour the content of the ladle; in this work, it is modeled in a simpler, more practical way: there must by a minimum number of jobs between two fly-tundish operations (fixed for each CC);

7. we simplify the cleaning process of the ladles, considering only two levels of cleanness (clean and dirty), but we take into account a generic set of possible pollutants (not just one as in [1]), so that the notion of clean ladle is multidimensional.

Remarkably, point 3 above remains the most peculiar difference with respect to [2] since it involves a change of perspective on the objective function. Given that we have to schedule the production for the next-time horizon, the goal is also to select the jobs, from a large job set, that have to be processed during the horizon, assign them to machines and plan the sequencing and timing (start and end time). Therefore, the problem in its essence is both a planning and scheduling problem.

## 2 Instances

We collected 60 real-world instances coming from a mid-sized steel-making plant, under different production conditions. The plant comprises two production lines followed by three CC machines, and two IC machines as casters. The main features of the instances are showed in Table 1, in terms of minimum and maximum number of jobs, steel grades, ladles, appointments, standstills, pollutants and horizon length.

We plan to release these instances (properly anonymized) in the near future, along with our best solution, for future comparisons. At the same time, we are preparing a visualization and validation tool that certifies the feasibility of the solution and its costs.

## 3 Solution Method

In essence, the problem is a complex variant of the *hybrid flow-shop* problem (see, e.g., [3] for a review) with sequence-dependent setup times and heterogeneous processing times. We tackle it by means of local search, modeled as follows:

- The *search space* is an indirect representation of the schedule by means of the sequence (i.e., the permutation) of all the available jobs. Each job has also a set of additional data stating on which machine it has been assigned for each specific processing step. The mapping between the job sequence and the actual execution times of each job is obtained by applying a chronological, right-justification greedy algorithm. All the jobs whose execution time exceed the time-horizon are part of the *tail* of the solution, which consists of the jobs considered as unscheduled.
- The *neighborhood* relation is defined as the set of swaps of the processing order of two jobs and, possibly, the reassignment of the jobs to different machines (for some processing step). The neighborhood is refined by removing moves that are clearly ineffective, such as for example moves that swap jobs that are both in the tail of the state.
- The *cost function* takes into account the difference between the upper bound of scheduled jobs and the number of jobs actually scheduled in the solution. In addition, it considers the violations of some hard constraints, as a ladle that is not clean as required and a job that misses an appointment.

The experimental analysis with different metaheuristics (hill climbing, simulated annealing, and tabu search) is still ongoing. We will report the results as soon as we complete the tuning and the experimental comparison.

## References

1. Davide Armellini, Paolo Borzone, Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. A metaheuristic approach for scheduling steelmaking and casting plants. In *Proceedings of the 12th Metaheuristics International Conference (MIC 2017)*, pages 1–3, Barcelona, Spain, July 4-7 2017.
2. Maria Pia Fanti, Giuliana Rotunno, Gabriella Stecco, Walter Ukovich, and Stefano Mininel. An integrated system for production scheduling in steelmaking and casting plants. *IEEE Transactions on Automation Science and Engineering*, 13(2):1112–1128, April 2016.
3. Rubén Ruiz and José Antonio Vázquez-Rodríguez. The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1):1–18, 2010.