Production Optimization and Scheduling in a Steel Plant: 
Hot Rolling Mill

Matteo Biondi*, Dr. Sleman Saliba**, 
Dr. Iiro Harjunkoski**

* ABB S.p.A, 16153 Genova, Italy  
**ABB AG, Corporate Research, 68526 Ladenburg, Germany  
(Tel: +49 6203 716014; e-mail: iiro.harjunkoski@de.abb.com)

Abstract: In this presentation we investigate the production scheduling of a hot rolling mill in a steel plant. The production schedule must adapt to strict production rules derived from metallurgic and physical constraints. Successful manual planning requires therefore in practice thorough process knowledge and years of operating experience. In our optimization-based approach, we first build parts of the rolling programs using intelligent heuristics and compose these parts to fully feasible programs by solving a min-cost-flow problem. In a second step, the built programs are scheduled using a mixed integer linear programming formulation in order to obtain an optimal schedule that violates as few order due dates as possible. Finally, we describe how to embed the current system within a standard IT environment, connecting the business level to the plant layer.

Keywords: Scheduling algorithms, Optimization, Steel industry, Heuristics, Mathematical programming

1. INTRODUCTION

Production scheduling in the steel industry has been recognized as one of the most difficult industrial scheduling problems. Many different and often contradicting constraints must be taken into account while defining a feasible and, possibly, optimal schedule for the production.

In one of the most typical production configurations, the steel-making process can be roughly subdivided into three parts: the melt shop, where melt steel is produced and cast into semi-finished products (slabs), see e.g. Harjunkoski I. and Grossmann I.E. (2001); the hot rolling, in which slabs are transformed by means of a mechanical and thermal process into final- or intermediate products (coils, billets, wires,…); cold rolling and finishing line operations can achieve customers’ specifications for final dimensions, surface quality and mechanical properties. These production steps are highly interconnected; the ideal situation would be to comprise all of them into one optimization model. However, this is currently not yet possible, both due to the computational and modeling limitations.

In our present work, we will focus on the problem of production scheduling on the hot rolling mill.

2. HOT ROLLING

2.1 The process

The hot strip mill typically consists of several processing stages (reheating furnace, roughing mill, finishing mill, down coiler), on which the slabs need to be processed sequentially; see e.g. Lenhard, J.G (2007) and Ginzburg V.B. and Ballas R. (2000) for a detailed description of the process. After the slabs have been heated to a temperature of around 1200°C in the reheating furnace, slabs are processed in a roughing mill that first reduces the thickness to around 50 mm. Then the slabs are processed in a finishing mill consisting of several sequential stands, during which the slabs (at the exit defined “strips”) reach their final thicknesses (typically 1-6 mm). Finally, the strips are cooled down by water jets and coiled in the down coiler.

2.2 Scheduling challenges

The rolling is performed through plastic deformation of the steel by subsequent pairs of work rolls. The input material, slabs, eventually become thinner and longer. Because of the high pressure, hard material and extreme temperatures, the work rolls will wear, especially at the edges of the strip, where a small “mark” is engraved during the rolling of several successive slabs. Due to the surface quality, it is therefore allowed to only reduce the width of successive coils; else the formed marks could cause visual defects on the surface of following coils. It is also important to couple steel qualities with similar hardness together to ensure an even product quality and minimize the risk of cobbling during threading of the material in the rolling stands. Additionally, only limited changes in the final thickness can be performed between consecutive orders to ensure process stability. These facts contribute to the complexity of the production rules that need to be strictly followed. These production rules are based...
on physical and metallurgical facts, as well as local experience and quality requirements.

Additionally, orders that are produced in the hot rolling mill need to meet customer due dates, if the product is sold right after the rolling mill, or internal due dates, if the coils are to be further processed in other sections of the steel plant (e.g. cold rolling mill). In the latter case, the product mix in the hot rolling mill has to also be balanced in order to be able to “feed” different parallel down-stream processes and thus enable optimal capacity utilization.

Due to the intriguing complexity and the variety of plant designs in metals hot rolling only few mathematical programming approaches with applications to real world steel plants have been published. Lopez et al. (1998) suggested a heuristic based on Tabu Search, which was successfully applied to Dofasco, a Canadian steel producer, but failed to be applied to other steel plants. Most recently, Zhao et al. (2009) applied a two-stage scheduling method to the hot rolling area of Baosteel, China.

3. SCHEDULING ALGORITHM

A production schedule for the hot rolling mill consists of a set of production campaigns (rolling programs), which are composed of a finite number of slabs/coils. The hot rolling scheduling problem consists of creating feasible rolling programs and sequencing them on the plant.

A pure single-step mathematical programming approach can neither capture all the relevant problem aspects nor meet the performance criteria. Therefore, a two-step approach combining heuristic-mathematical programming methods has been developed. The steps are:

1. Build hot rolling programs
2. Sequence the built rolling programs

The programs built in step 1 should be as long as possible and contain as many orders as possible meeting production and quality rules. The building procedures of feasible rolling programs takes into account all necessary rules for allowed width and thickness changes, as well as metallurgical and physical constraints related to subsequent coil compatibilities.

The procedure first applies a construction heuristics to form program bodies. The program bodies are denoted by \( b_i, i = 1,..,n \). Additionally, we introduce two special vertices, a source \( s \) and a sink \( t \). Therefore, \( V = \{ b_1,..,b_n, s, t \} \). For each body \( b_i \) that can be followed by body \( b_j \) in a feasible program, we introduce an arc \( a = (b_i,b_j) \). Moreover, we connect the source \( s \) to each body that can be the first body in a program with an arc. Each body that can be the last body in a program is connected with the sink \( t \) by an arc.

Each arc connecting two bodies has an associated cost \( p(b_i,b_j) \) that corresponds to the negated profit of the bodies \( b_i \) and \( b_j \) and the profit of combining the two bodies to the same program. The arcs connecting the source and the sink have no additional cost.

Moreover, we also associate with each arc \( a \in A \) a capacity \( c(a) \) denoting the maximum amount that can flow on the arc. In our problem, the capacity of each arc is at most one unit of flow.

The objective is now to minimize the cost of the flow from the source to the sink:

\[
\min \sum_{i,j=1}^{n} p(b_i,b_j) f(b_i,b_j) \tag{1}
\]

Capacity constraints ensure feasible flows by setting the binary capacity parameter \( c(a) \) to 1 for allowed combinations and prohibiting flows directly from source to the sink.

\[
f(b_i,b_j) \leq c(b_i,b_j) \quad \forall i, j \tag{2}
\]

\[
f(s,b_j) \leq c(s,b_j) \quad \forall j \tag{3}
\]

\[
f(b_i,t) \leq c(b_i,t) \quad \forall i \tag{4}
\]

Flow conservation constraints ensure that the flow coming into the vertex and the flow leaving the vertex are the same.

\[
f(s,b_j) + \sum_{i=1}^{n} f(b_i,b_j) - \sum_{k=1}^{n} f(b_j,b_k) - f(b_j,t) = 0 \quad \forall j \tag{5}
\]

Uniqueness constraints ensure that the incoming flow for each vertex must be less than or equal to 1, resulting in that each program section can be used only once.

\[
f(s,b_j) + \sum_{i=1}^{n} f(b_i,b_j) \leq 1 \quad \forall j \tag{6}
\]

A max-flow problem is solved upfront to define the optimal number of flows.

\[
\sum_{i=1}^{n} f(s,b_i) = \text{MaxFlowValue} \tag{7}
\]

\[
f(b_i,b_j) \in \mathbb{R}^+ \quad f(s,t) = 0
\]

This flow problem is repeated in a prioritized sequence for each type of targeted programs, i.e. how many sections a program is composed of. A more comprehensive overview on
the algorithmic concepts of solving minimal cost flow problems can be found in Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993).

Using the described approach for building hot rolling programs, we can ensure that all production requests that were not included in a program can neither form a valid program by themselves nor be added to already built programs.

The concept of building skeletons and filling the skeletons to form program parts ensures that we always consider the most valuable production requests first. Valuable production requests are e.g. coils with early due dates or coils with minimal finishing thicknesses.

Moreover, the utilization of the minimum cost flow problem for composing program parts to full programs ensures that the most valuable program parts are selected and that the combination of program parts is optimal in terms of similar due dates and common steel properties. Valuable program parts are e.g. bodies that contain a high number of valuable coils and that form a long sequence of production requests in kilometers.

Therefore, our approach results in a feasible rolling program meeting the quality requirement, while maximizing the number of rolled production requests and the value of the production requests, as well as minimizing the work roll changes and the use of waste material.

The built programs are then sequenced, which is a more traditional scheduling-type of problem. An MILP formulation of the problem is proposed taking into account due date and production mix constraints. The formulation is an extension of the slot-based approach by Pinto, J.M. and Grossmann, I.E. (1995) and will not be further discussed here.

Considering production requests of one month (about 2000 – 5000 coils), the program building procedure takes less than 30 seconds of computational time. The second step, sequencing the programs, requires more computational effort. Restricting the computational time of the sequencing MILP to 120 seconds yields sufficiently good sequences. Therefore, we can ensure a total computational time of strictly less than three minutes.

Since we consider the orders for up to several weeks (e.g. production requests in the order book for the next month), in addition to the above benefits, the scheduling department gains a better insight into the order book and the additional material needed to fill the gaps in the order portfolio. The visualization of the production plan for the next weeks and the highlighting of key performance indicators enable the schedulers to plan and react more accurately to the business plan and therefore, improve the productivity of the plant.

Figure 1. Building hot rolling programs algorithm: orders for the hot rolling mill are organized into rolling program sections.

Figure 2. Rolling program sections are combined into feasible rolling programs by solving a flow problem on graph.

4. CONNECTION TO DCS AND ERP SYSTEMS

The scheduling system described above has been implemented as a web-service and can be easily embedded within a standard IT infrastructure, combining data both from the ERP and the DCS levels. As an example, the scheduling client may receive the production orders from an ERP system and the plant related information from the DCS. The client then calls the scheduling server with an XML file containing the production orders including all needed parameters following the B2MML (Business to Manufacturing Markup Language, ISA-95) standard. For plant related information such as local production rules, an ad-hoc structure has been implemented. The resulting schedule is also provided as an ISA-95 compliant XML file. This architecture of the proposed system provides through its flexibility easy connectivity to any control system. The scheduling server has always a standardized interface and can thus adapt to different environments and production rules only by
modifications on the client. Figure 3 shows a simplified communication overview.

![Figure 3](image_url)

Figure 3. The scheduling system is implemented as a Web-service. Information can be gathered from ERP and DCS systems and sent to the scheduling server.

Once the schedule has been optimized and transferred back to the client through XML, it can be displayed in various ways. Figure 4 shows eight rolling programs in the sequence from left-to-right. Within one rolling program, the various steel families and program sections are displayed in different colors, and the length (km) of programs is directly proportional to the picture.

![Figure 4](image_url)

Figure 4. Screenshot of the GUI client. The typical “coffin-shape” of hot-rolling programs is shown. The vertical size corresponds to the program length and horizontal to the coil widths.

5. RESULTS AND CONCLUSIONS

In this work we presented a method for production planning on a hot rolling mill that creates optimized hot rolling programs and schedules. The optimized production schedule is obtained through the construction and sequencing of hot rolling programs. The algorithm combines efficiently intelligent heuristics and mathematical optimization. Moreover, the presented solution ensures a high transparency of the planning process through configurable and adaptable planning rules in B2MML compliant XML-files. Finally, the solution is implemented as a web service and can be easily integrated into any existing IT environment using standard interfaces in ERP and DCS systems. Through this flexibility the method can be used either as stand-alone or connected to other solutions. In the presented approach only the rolling programs are defined and e.g. the re-heating furnace planning is not considered. Owing to this and the possibility to pre-specify and enforce rolling program sequences, it is possible to use the method to support hot charging, which can result in significant energy savings.

Some benefits of the presented hot rolling scheduling system are:

- It is possible to plan and have an overview of a full month’s orders (e.g. 5000 coils) at one instance resulting in highly optimal solutions
- A full overview also helps e.g. in checking whether the production from the order book is well balanced (in terms of dimensions, qualities,…)
- Slab yard can efficiently be included in the planning helping to reduce the overall slab-yard inventory
- The optimization results in longer programs and fewer setups, reducing the total production time and costs
- By modifying the set of rules and parameters, new planning strategies can be easily implemented, e.g. when introducing new products or quality requirements
- Re-planning takes seconds instead of hours, which makes it possible to efficiently react to e.g. changes in order book or unusual plant condition (delays on the production line, broken equipment,…).

The described algorithm combining heuristics and mathematical optimization methods has been tested on a large number of real problems. A typical problem size may include 3000-5000 coil orders and the corresponding rolling programs can be generated in the range of seconds. Depending on the problem size, the sequencing of programs may take somewhat longer. By enabling an option to allow the introduction of auxiliary orders, the approach can ensure that all coils are included into programs and in some instances up to 95% of the available coils could be included in rolling programs, by strictly following the specified rules. This can be considered to be an outstanding result that over performs the manual capabilities as well as many existing alternative approaches.

REFERENCES


