A Decomposition Approach for the Scheduling of a Steel Plant Production

Iiro Harjunkoski and Ignacio E. Grossmann*

Department of Chemical Engineering
Carnegie Mellon University, Pittsburgh, PA 15213

Email: iiro@andrew.cmu.edu

February 2001 / Rev. July 2001

Abstract

In this paper we present a decomposition strategy for solving large scheduling problems using mathematical programming methods. Instead of formulating one huge and unsolvable MILP problem, we propose a decomposition scheme that generates smaller programs that can often be solved to global optimality. The original problem is split into subproblems in a natural way using the special features of steel making and avoiding the need for expressing the highly complex rules as explicit constraints. We present a small illustrative example problem, and several real-world problems to demonstrate the capabilities of the proposed strategy, and the fact that the solutions typically lie within 1-3% of the global optimum.

Keywords: Mixed Integer Programming, Steel Making, Scheduling, Heuristics, Disaggregation.

1. Introduction

Large-scale combinatorial problems arise frequently in the area of scheduling and process synthesis. The main goal of scheduling is to assign and sequence a number of jobs into the equipment of a given production facility within a certain time frame such that each job is completed before its due date. The time is critical also in two other aspects: the equipment should normally not process more than one job at a time, and in multistage production, the previous stage of a job needs to be completed before proceeding to the next one. To meet these requirements, either a discrete or continuous time model needs to be applied. Scheduling problems can further be classified by a number of criteria and these are discussed

* Author to whom all correspondence should be addressed. Email: grossmann@cmu.edu

in Pinto and Grossmann (1998) and Shah (1998). The main difficulty in these types of problems lies in the number of discrete variables that is needed for a valid representation of the model. Because of this, Mixed-Integer Programming (MIP) approaches may require exponential computation times and often fail to even find a feasible solution (Pekny and Reklaitis, 1998). This trend can be observed in most common approaches when the problem size is increased to real-size scales. The approaches include the uniform time discretization (Kondili et al., 1993; Pekny and Zentner, 1994), the time-slot based approach (Pinto and Grossmann, 1995), and continuous time approaches (Zhang and Sargent, 1996; Mockus and Reklaitis, 1996; Ierapetritou and Floudas, 1998).

A number of strategies to overcome this problem have been proposed. Heuristic methods using for instance genetic search (Rubin and Ragaz, 1995) may give a feasible solution, but cannot guarantee the quality of the result. Combining Constraint Logic Programming (CLP) methods with Mixed Integer Linear Programming (MILP) has been proposed by Jain and Grossmann (2001) for a certain class of scheduling problems. A third category of methods combines heuristics with mathematical programming methods which results in a reduction of discrete variables and/or an improved relaxation of the original problem (Elkamel et al., 1997; Lee et al., 1997; Roslöf et al., 2001). The approach proposed in this paper belongs to the last category. Here, we apply a continuous time representation for modeling a large-scale scheduling problem.

Production scheduling in the steel industry has been recognized as one of the most difficult industrial scheduling problems. It involves a number of stages, each of which has many critical production constraints. There have been several studies only on part of the problems, e.g. continuous casting (Lally et al., 1987) and batch annealing process, including crane operations (Moon and Hrymak 1999). Most of the methods use expert systems, heuristics, or fuzzy logic approaches to generate feasible solutions. A brief overview on expert systems is given by Dorn and Doppler (1996) followed by a more detailed study on a scheduling expert system (Dorn and Shams, 1996). A thorough overview of the process and computerized scheduling solutions including OR-techniques, AI methods and combinations of these two is given in Lee et al. (1996). Apart from optimization, simulation has also been used to verify the chosen production strategy, and to identify possible bottlenecks and machine inconsistencies.

Modeling and solving a steel-shop scheduling problem involves handling a large number of complicated chemistry-, geometrical- and scheduling rules. This means in practice that the production is highly sequence dependent. Due to this complexity using a straightforward MILP approach may require excessive computation time, even for a problem with only 10 jobs. Instead, MILP can be complemented by heuristics and in this case the fact that the jobs (often named as heats) need to be specifically grouped into sequences for the last processing stage, continuous casting, motivates a decomposition strategy. The decomposition strategy proposed in this paper consists in first partitioning the customer orders into groups of heats with similar properties, each of which is optimized as a jobshop scheduling problem (Rich and Prokopakis, 1986). Next, each of the groups are optimally scheduled as a flowshop problem (Birewar and Grossmann, 1989), and finally an LP and/or MILP method is used to properly account for setup times and to optimize the allocation of some parallel equipment. While this decomposition strategy is not guaranteed to yield the global optimal schedule, it allows the solution of very large-scale problems. Whereas the traditional MILP approaches fail to solve in reasonable time problems with 8-10 products, it will be shown that the proposed decomposition strategy can be successfully applied to the scheduling of close to 100 products which equals to almost one week production. Moreover, a theoretical lower bound can be calculated to assess the quality of the solutions, which lie within 1-3% of the global optimum.

2. Problem definition

The steel-making process studied consists of two electric arc furnaces (EAF), where the melt steel is combined with scrap, and thereafter taken to argon oxygen decarburization (AOD) and ladle metallurgy facility (LMF) units. Finally, the melt steel is solidified in a continuous caster under strictly constrained conditions. The process is illustrated in Fig. 1. There are several complicating factors in the problem such as sequence dependent setup times, maintenance of equipment and production time limitations. Problems arising from the metal chemistry, as well as plant geometry are not directly considered in the model. Thus, a working scheduling model must not only consider equipment availability, but also the temperature requirements of each product at each stage, chemistry constraints, for instance between subsequent products and material properties, both in the products, as well as in the equipment that operate under extreme conditions.

The continuous casting involves solidifying of the melt steel to steel slabs of a prespecified width and thickness. When a continuous steel flow is broken, the caster needs maintenance and the caster mold needs to be replaced, which involves costs and a delay in production. This happens, for instance, when the product type is changed. If the two subsequent products are very similar, it may be possible to mix them and go on without stopping, but in the other case the caster needs to be stopped for service. Moreover, the caster can only be run continuously for a limited number of products due to the extreme conditions. Because of this, the continuous casting can be considered as one of the main challenges in steel production planning, and even getting feasible solutions is not trivial.

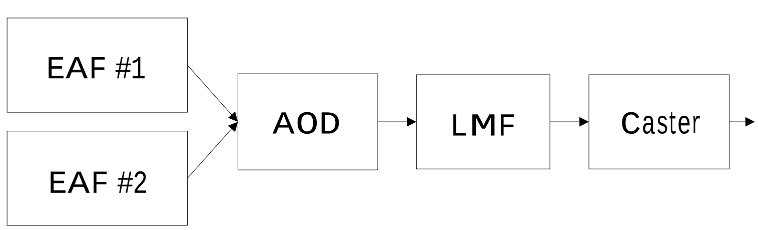

Figure 1. The example process

The products of the steel making process are defined by their grade, that is basically a product quality description, including both chemical and physical properties. Each grade has a given production recipe including strict specifications of temperature, chemistry and processing times at different stages. Grades are further subdivided into sub-grades, that have minor differences to the actual grade (e.g. lower carbon content) Example: grade 555 may have a low-carbon sub-grade 555L. Given the grade constraints, production equipment and a number of customer orders (heats), the problem is to find a schedule that minimizes the makespan following the recipe for each product grade. The different grade qualities and dimensions pose chemical, as well as physical restrictions that need to be considered in the planning stage. The model that follows is, however, a simplification of the real case, but captures the critical features.

In the following, we will use the term products for individual heats or orders, and grades for a product type. Also, the term group will be used to express a sequenced subset of products. In order to tackle the scheduling problem some assumptions are made. Grades and respective sub-grades with the same slab thickness can be cast in the same sequence, although the order is specified by certain rules. Each product is characterized by its grade, slab width and -thickness. Many of the complicated chemistry rules can be embedded into parameters and need not be explicitly considered in the model.

3. Solution strategy

The proposed decomposition strategy involves presorting and four major optimization steps (three MILP and one LP) as illustrated in Fig. 2. The following steps are involved in the proposed decomposition strategy:

1. Presorting the products into product families
2. Optimal disaggregation of product families into groups
3. Scheduling each of the groups independently
4. Scheduling of all groups
5. LP-improvement problem

The main issues for each step will first be discussed together with an introduction of the major concepts. This is followed by a mathematical formulation, where we first present the general case and thereafter discuss some specific details, that may for instance, require addition or modification of certain constraints.

3.1 Presorting the products

The products (orders) need to be presorted in order to classify them into product families that have the same grade and thickness. This is done as follows. The products are ordered sequentially according first to their casting speed, next to due date, then to slab width, followed by the grade and finally slab thickness. Each grade is also ordered according to their "sub-grades" which have to follow certain ordering rules. Briefly, the ordering is done in increasing importance such that, for example, after the last and most important sorting of the thickness, all products with the same thickness follow each other (see Appendix I). After the ordering, the sorting takes place simply by scanning through the list of products starting from the top and joining subsequent products that are similar within given tolerances. This "manual" sorting procedure provides an upper limit to the number

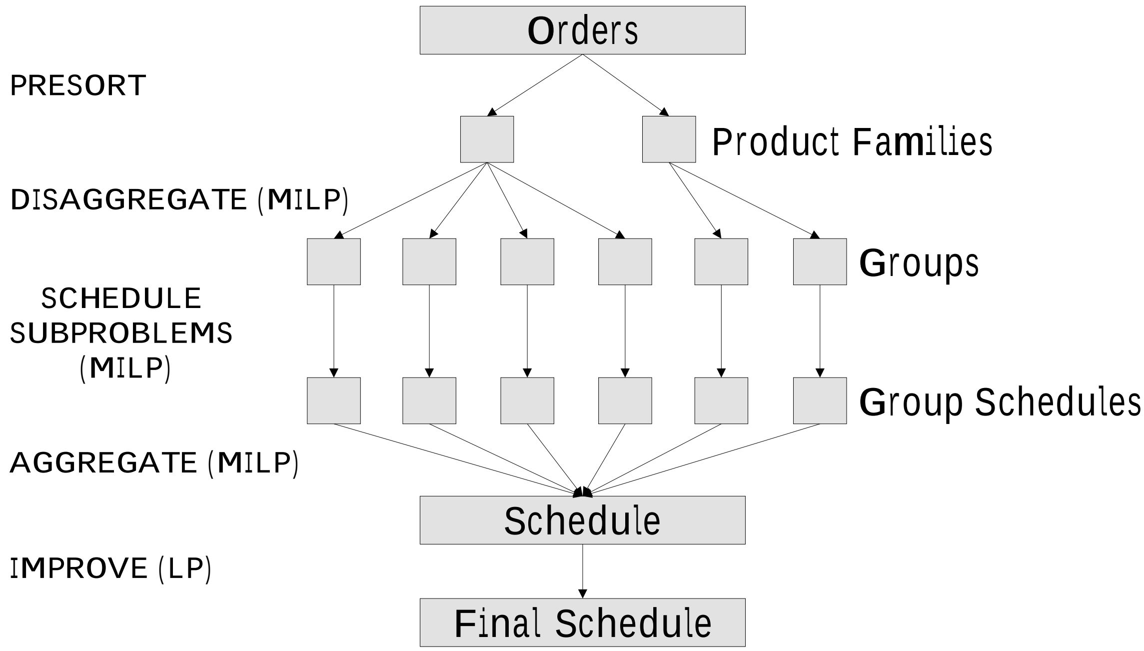

Figure 2. Decomposition strategy

of sequences (or groups) needed for the continuous casting. The two lists below illustrate a simple case with the unsorted products on the left and presorted on the right hand side, which contains 5 product families.

Before sorting			*After sorting*		
Grade	**Thickness**	**Width (in)**	**Grade**	**Thickness**	**Width (in)**
777	5.0	40.0	555	5.0	42.5
666L	5.0	38.5	555	5.0	37.0
555	5.0	42.5	666L	5.0	38.5
777L	5.0	43.0	666	5.0	33.0
777	6.0	45.0	777L	5.0	43.0
666	5.0	33.0	777	5.0	40.0
777	6.0	44.0	666	6.0	50.0
555	5.0	37.0	777	6.0	45.0
666	6.0	50.0	777	6.0	44.0

3.2 Optimal disaggregation of products into groups

After the presorting step the products are ordered such that a product cannot be followed by any of its predecessors in the sorted list, except if it has exactly the same properties (grade, sub-grade, width and thickness). An upper bound for the number of groups needed for covering all the products is obtained from the presorting. Having completed the presorting step, each product family is disaggregated into smaller, solvable problems consisting of groups. The motivation for this disaggregation lies in the last processing stage, the continuous casting, which is done under the following assumption. A number of similar grades can be cast in a sequence (at most M_{max} products) after which the caster needs maintenance, typically this consists of a caster mold change that takes TM_{cst} minutes. A sequence cannot contain different slab thicknesses and grades, and in order to ensure the required product quality, the sub-grades in each sequence need to be ordered in a certain way, for instance low carbon products should always precede high carbon products. Also, the products need to be cast in decreasing width within each sequence, and the slab width difference between two subsequent products within a group is restricted (Δ_w).

This motivates the need for developing a formulation for the grouping. The matching between products and groups, as well as the exact casting sequences are obtained as a result from this problem. Each family of grades (grades and their corresponding sub-grades) are grouped separately under the following conditions:

- the number of groups should be minimized since each group change imposes a TM_{cst} minute changeover time
- different grades and slab thicknesses cannot be mixed in a group
- an overall goal is to run longer caster sequences

The grouping problem for each product family can be formulated as follows. The objective function (1) minimizes the number of groups needed, i.e. the sum of variables, z_g, that are equal to one if any product $i \in I$ has been assigned to the group $g \in G$, or else equal to zero,

$$\min_{x_{ig}, z_g, y_{ii'g}, q_{ig}} \sum_{g \in G} z_g \tag{1}$$

Although the variables z_g are 0-1, they can be treated as continuous as will be seen below. Defining x_{ig} as the variable for assigning product i in group g, constraint (2) requires that a product i can be assigned to only one group g (logical expression: $\underline{\vee}\, x_{ig}$), and constraint (3) sets an upper limit to the number of products in a group, M_{max}, as well as, defining the relationship between the variables x_{ig} and z_g.

$$\sum_{g \in G} x_{ig} = 1 \qquad \forall i \in I \tag{2}$$

$$\sum_{i \in I} x_{ig} \leq M_{max} \cdot z_g \qquad \forall g \in G \tag{3}$$

Constraints (4)-(6) define a continuous variable, $y_{ii'g}$, to be equal to one if both products i and i' are in the group g, else zero, which logically is equivalent to $x_{ig} \wedge x_{i'g} \Leftrightarrow y_{ii'g}$. This condition can be formulated as the inequalities,

$$y_{ii'g} \geq x_{ig} + x_{i'g} - 1 \qquad \forall i, i' \in I, i < i', \ \forall g \in G \tag{4}$$

$$y_{ii'g} \leq x_{ig} \qquad \forall i, i' \in I, i < i', \ \forall g \in G \tag{5}$$

$$y_{ii'g} \leq x_{i'g} \qquad \forall i, i' \in I, i < i', \ \forall g \in G \tag{6}$$

The products in a group need to meet several requirements. The decreasing casting width is enforced by a logical expression, $y_{ii'g} \Rightarrow (W_i \geq W_{i'})$ for $i < i'$, given in Eq. (7), where W_i is the given width of order i. In order to be able to handle the more complicated constraints linearly, a precedence matrix, $P_{ii'}$ is formed to denote if product i' can be cast directly after product i ($P_{ii'} = 1$) or not ($P_{ii'} = 0$). Equation (8) states that all products except one (the last in the group) has to be followed by a suitable product, the logic equivalence of which is $(x_{ig} \wedge \neg q_{ig}) \Rightarrow (\vee y_{ii'g})$ for $P_{ii'} = 1, i < i'$. Constraint (9) makes sure that the sum of relaxation variables for this, q_{ig}, does not exceed one for any active group.

$$(W_i - W_{i'}) \cdot y_{ii'g} \geq 0 \qquad \forall i, i' \in I, i < i',\ \forall g \in G \tag{7}$$

$$\sum_{i' \in I, i' > i} P_{ii'} \cdot y_{ii'g} \geq x_{ig} - q_{ig} \qquad \forall g \in G,\ \forall i \in I \tag{8}$$

$$\sum_{i \in I} q_{ig} \leq z_g \qquad \forall g \in G \tag{9}$$

$$x_{ig} \in \{0, 1\}$$

$$0 \leq z_g, y_{ii'g}, q_{ig} \leq 1$$

$$y_{ii'g} = 0 \qquad \forall i, i' \in I, i \geq i'$$

This formulation requires only one set of binary variables, x_{ig}. The other variables can be treated as continuous, ranging between 0 and 1. Because of the pre-ordering some of the $y_{ii'g}$ variables can be fixed. The formulation can further be tightened through elimination of multiple solutions by forcing the first groups to be active ($z_{g+1} \Rightarrow z_g$) and ordering the groups by decreasing number of products. This is done in Eqs. (10) and (11).

$$z_{g+1} \leq z_g \qquad \forall g \in G, g < |G| \tag{10}$$

$$\sum_{i \in I} (x_{i,g+1} - x_{ig}) \leq 0 \qquad \forall g \in G, g < |G| \tag{11}$$

The formulation given in Eqs. (1)-(11) is capable of disaggregating orders optimally into groups for a typical one week set of orders, where no more than 30-35 products appear per grade. The result of this step is not only an optimal grouping strategy that minimizes the casting setup times, but also the correct casting order, as illustrated in Fig. 3, in which a grade with its two sub-grades (vertical and horizontal lines) containing 9 presorted products are divided into 2 groups, or casting sequences.

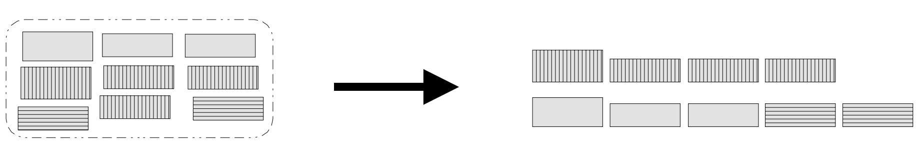

Figure 3. The disaggregation step

3.3. Scheduling each of the groups independently

After solving the optimal grouping problem, the next step of the proposed strategy is to schedule the production independently for each group. This step is performed using a jobshop scheduling formulation. It should, however, be noted that since the casting sequence has already been fixed at the previous step, the same production sequence often applies to the rest of the production. This makes the problem fairly efficient to solve, and the main purpose of solving the individual group schedules lies in defining the required time window for each equipment, as well as handling the case of parallel units. Also, this step takes into account certain restrictions, e.g. the maximum hold-times of the heats. Depending on the grade, the heats should not spend more than a certain time, the hold-time, between the AOD tap and the beginning of the casting in order to keep the heat temperatures within a required range. Otherwise, the production of a grade may fail and may not meet its quality requirements without additional heating. An example of this step can be seen in Fig. 4. In the following, a simplified formulation is first given without the parallel EAF-units, followed by a discussion and the constraints that are needed for handling two parallel EAF furnaces.

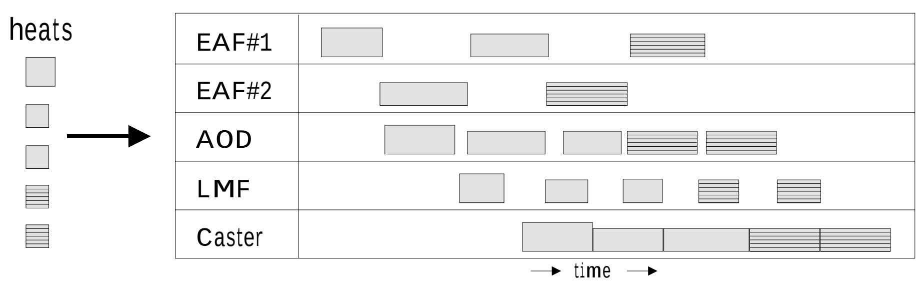

Figure 4. The scheduling of groups

The objective is to schedule the products $i \in I$ into machines $m \in M$ while minimizing an objective that combines the makespan, in-process times (the difference between start-time of the first and last machine) and the positive slack-variables representing hold-time violations. The coefficients C_1 and C_2 in the objective function (12) are weighted such that C_2 dominates to reduce the violation of hold-times. The coefficient C_1 need not be large since the makespan is more important, than the in-process times.

$$\min_{t_{im}, Y_{ii'm}, t^{MS}, sl} \quad t^{MS} + C_1 \cdot \sum_{i \in I} \left(t_{i,|M|} - t_{i,1} \right) + C_2 \cdot sl \tag{12}$$

The makespan is defined in constraint (13). The precedence constraints between the products are determined using a binary variable $Y_{ii'm}$ to indicate that product i precedes i' in machine m. These are defined in Eqs. (14) and (15), where U is a valid upper bound.

$$t_{im} + \tau_{im} \leq t^{MS} \qquad \forall\, i \in I,\ m = |M| \tag{13}$$

$$t_{im} \geq t_{i'm} + \tau_{i'm} + T_m^s + T_m^{cl} - U \cdot Y_{ii'm} \qquad \forall i, i' \in I, i \neq i',\ \forall\, m \in M, m < |M| \tag{14}$$

$$Y_{ii'm} + Y_{i'im} = 1 \qquad \forall\, i, i' \in I, i \neq i',\ \forall\, m \in M \tag{15}$$

Constraint (14) involves machine setup and clean-up times $T_m^s + T_m^{cl}$ that can easily be customized, e.g. for sequence dependent setup times. The production constraint enforcing the order of operations is given in Eq. (16), and the hold-time is defined in Eq. (17).

$$t_{i,m+1} \geq t_{im} + \tau_{im} + T_{m+1}^s \qquad \forall\, i \in I,\ \forall\, m \in M, m < |M| \tag{16}$$

$$t_{i,CST} - (t_{i,AOD} + \tau_{i,AOD}) \leq H_{i,max} + sl \qquad \forall\, i \in I \tag{17}$$

Finally, the previously sequenced continuous casting can be enforced through Eq. (18).

$$t_{i+1,m} = t_{im} + \tau_{im} + T_m^s \qquad \forall\, i \in I, i < |I|,\ m = |M| \tag{18}$$

$$Y_{ii'm} \in \{0, 1\}$$

$$t_{im}, t^{MS}, sl \geq 0$$

The simplified formulation in Eqs. (12)-(18) does not consider parallel EAF units or variable processing times. Incorporating variable processing times is straightforward and does not involve nonlinearities, while the processing times can just be changed to positive variables. Handling the two parallel electric arc furnaces requires introduction of new binary variables and complicating big-M constraints. The new variables are marked with the superscript EAF. Two binary variables are needed: Z_{im}^{EAF} to assign each product to one of the two furnaces, and $Y_{ii'}^{EAF}$ to define the precedence of the EAF step, since electricity can only be fed into one furnace at a time.

The sequencing constraint in (14) needs to be changed for the furnaces. In Eq. (19), the additional big-M terms ensure that only products in the same EAF-unit are compared ($Y_{ii'm} \wedge Z_{im}^{EAF} \wedge Z_{i'm}^{EAF} \Rightarrow t_{i'm} \geq t_{im} + \tau_{im} + T_m^s + T_m^{cl}$). Constraint (20) states that exactly one furnace is assigned to each product.

$$t_{i'm} \geq t_{im} + \tau_{im} + T_m^s + T_m^{cl} - U\left(1 - Y_{ii'm} + 1 - Z_{im}^{EAF} + 1 - Z_{i'm}^{EAF}\right)$$

$$\forall\, i, i' \in I,\ \forall\, m \in M^{EAF} \tag{19}$$

$$\sum_{m \in M^{EAF}} Z_{im}^{EAF} = 1 \qquad \forall\, i \in I \tag{20}$$

Additional constraints are needed because of the electricity restriction above. The time T^{el} stands for the electricity feed and overlapping is hindered in Eq. (21) followed by the EAF-precedence variable definition in Eq. (22).

$$\sum_{m \in M^{EAF}} (t_{i'm} - t_{im}) \geq T^{el} - U\left(1 - Y_{ii'}^{EAF}\right) \qquad \forall\, i, i' \in I, i \neq i' \tag{21}$$

$$Y_{ii'}^{EAF} + Y_{i'i}^{EAF} = 1 \qquad \forall\, i, i' \in I, i < i' \tag{22}$$

Constraint (15) needs to be relaxed for the furnaces to handle the case where only one product is processed on each EAF. This is enforced through Eq. (23). The relationship between the general precedence variables and the EAF assignment variables is defined in Eq. (24),

$$\sum_{m \in M^{EAF}} (Y_{ii'm} + Y_{i'im}) \leq 1 \qquad \forall\, i, i' \in I, i \neq i' \tag{23}$$

$$Y_{ii'm} \leq Z^{EAF}_{i''m} \qquad \forall\, i, i' \in I, i \neq i', (i'' = i) \vee (i'' = i'),\ \forall\, m \in M^{EAF} \tag{24}$$

Because of the two parallel EAF-units, the production constraint for the following AOD-unit needs to be redefined. This is done in Eq. (25) which is valid if the start-time for the unused EAF is set to zero, as in Eq. (26).

$$t_{i,AOD} \geq T^s_{AOD} + \sum_{m \in M^{EAF}} \left(t_{im} + \tau_{im} \cdot Z^{EAF}_{im}\right) \qquad \forall\, i \in I \tag{25}$$

$$t_{im} \leq U \cdot Z^{EAF}_{im} \qquad \forall\, i \in I, m \in M^{EAF} \tag{26}$$

$$Y^{EAF}_{ii'}, Z^{EAF}_{im} \in \{0, 1\}$$

Introduction of the new variables and constraints in (19)-(25) makes the relaxation poorer and increases the search space. In some cases this may result into difficulties when solving the problems. Depending on the specific instance, the model can be tightened, for instance, by alternating the furnaces, as in Eq. (27), which basically is almost as powerful as fixing the EAF assignment a priori. In practice, this is a desired strategy which is often used and hence, the parallel equipment should not be a complicating factor in this optimization step. Also, another practical issue is to define the same ordering for the LMF (zero-wait) as for the caster. This is expressed in Eq. (28) and is also motivated by the fact that often the product is taken to the caster immediately from the LMF, which also decreases the hold-times.

$$Z^{EAF}_{im} + Z^{EAF}_{i+1,m} = 1 \qquad \forall\, i \in I, i < |i|, m = 1 \tag{27}$$

$$\sum_{i' \in I} (Y_{ii'm} - Y_{i+1,i'm}) = 1 \qquad \forall\, i \in I, i < |I|, m = |M| - 1 \tag{28}$$

Even if the jobshop model is fairly complex, the problem can be solved efficiently using the simplifications above. For less than 10 products, most of the groups can be scheduled optimally in a few seconds. This problem is solved separately for each group (see Fig. 2), and as a result we obtain valid schedules with optimal assignments and production sequences. From these schedules the time occupation of each unit can be extracted, which allow us to form a large-scale problem, where the optimal order of the groups is sought.

3.4. Scheduling of all groups

Having found the optimal strategy for the grouping of the products (disaggregation), and thereafter scheduled independently each group using the previous MILP approach, the groups are ordered in the next step using the slot-based approach by Birewar and Grossmann (1989). The main idea is to focus on the scheduled groups instead of the individual products. The principle is illustrated in Fig. 5. The figure shows how the equipment requirements are extracted from the group. The groups will then form time blocks which are finally ordered in a way that minimizes the makespan and tardiness/earliness. The due-date for each group is the earliest one of its products and the time window for each equipment is defined from the start of the first product to the end of the last one. Apart from only minimizing the makespan, certain sequencing issues are considered here. The two main issues are considering special grade requirements due to purity constraints, and reducing the number of caster mold thickness changes.

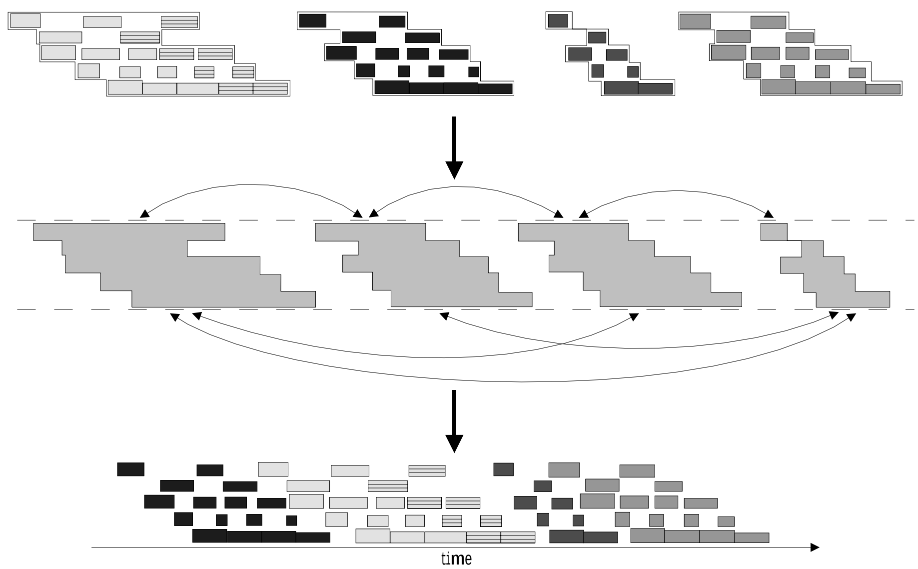

Figure 5. The aggregation step

As shown in Fig. 5, this step is performed by only looking at the groups without considering individual products. The problem can be solved as a flowshop scheduling problem, since there is no parallel equipment and all the groups should follow the same production path. After the slot-ordering has been performed, the complete production order is known, and can be extracted by combining the results from this and the previous step.

The objective function (29) minimizes the makespan, tardiness, earliness and mold thickness changes. The values of the coefficients (C_1, C_2, C_3) depend on the problem, but generally the tardiness should be penalized more than earliness. These are expressed by continuous variables. The coefficient C_3 for a thickness change that is detected here by positive slack-variables depends on the problem, but should be fairly high to avoid unnecessary setup times and costs.

$$\min_{z_{gl},t^S_{ml},t^E_{ml},t^{MS},z_{gg'l},t^{early}_l,t^{tardy}_l,sl^+_l,sl^-_l} t^{MS} + \sum_{l \in L} \left(C_1 \cdot t^{tardy}_l + C_2 \cdot t^{early}_l + C_3 \cdot (sl^+_l + sl^-_l) \right) \tag{29}$$

Only one set of binary variables is required for the slot-assignment of the groups. Constraints (30-31) define the new binary variables z_{gl}, i.e. each group g can only be placed into one slot l, and each slot l must be assigned to exactly one group g. This is also illustrated in Fig. 6. Also, a continuous variable $w_{gg'l}$ needs to be defined in Eq. (32) to identify subsequent groups ($z_{gl} \wedge z_{g',l+1} \Rightarrow w_{gg'l}$).

$$\sum_{g \in G} z_{gl} = 1 \qquad \forall\, l \in L \tag{30}$$

$$\sum_{l \in L} z_{gl} = 1 \qquad \forall\, g \in G \tag{31}$$

$$w_{gg'l} \geq z_{gl} + z_{g\prime,l+1} - 1 \qquad \forall\, g, g' \in G,\ \forall\, l \in L, l < |L| \tag{32}$$

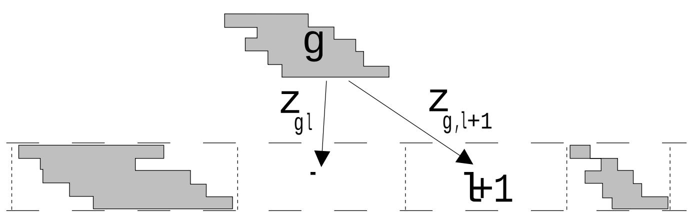

Figure 6. The slot-assignment variable

The following constraints define the time boundaries for each slot. Constraint (33) simply states that the end time, t^E_{ml}, of a machine in a group is the start time, t^S_{ml}, plus the total machine time $T^D_{g,m}$ for the group, determined in the previous scheduling step. The start time of the first machine, $t^S_{1,l}$, is used as a reference point in Eq. (34) that defines the start times of the remaining equipment, utilizing fixed delays from the previous step, $T^{ST}_{g,m}$, between the start times of the first machine and the remaining machines.

$$t^E_{ml} = t^S_{ml} + \sum_{g \in G} T^D_{g,m} \cdot z_{gl} \qquad \forall\, m \in M,\ \forall\, l \in L \tag{33}$$

$$t^S_{ml} = t^S_{1,l} + \sum_{g \in G} T^{ST}_{g,m} \cdot z_{gl} \qquad \forall\, m \in M, m \geq 2,\ \forall\, l \in L \tag{34}$$

Equation (35) defines that the operations in the next slot should not start before the completion of the previous one followed by a machine specific setup-time. The sequence, as well as mold thickness changes in the caster, are considered in Eq. (36).

$$t^S_{m,l+1} \geq t^E_{ml} + T^s_m \qquad \forall\, m \in M, m < |M|,\ \forall\, l \in L, l < |L| \tag{35}$$

$$t^S_{m,l+1} \geq t^E_{ml} + T^C_{grp} + \sum_{g,g' \in G, g \neq g'} T^{C,thk}_{gg'} \cdot w_{gg'l} \qquad m = |M|,\ \forall\, l \in L, l < |L| \tag{36}$$

Constraints (37) and (38) consider the electricity feed restriction and the makespan is defined in (39). Constraint (40) defines the earliness and tardiness, both of which are represented by positive continuous variables and the RHS term will be equal to the due date of the group assigned to the slot. In this way the problem will never be infeasible because of due date violations.

$$t^S_{2,l+1} \geq t^S_{1,l} + T^{el} \qquad \forall\, l \in L, l < |L| \tag{37}$$

$$t^S_{1,l+1} \geq t^S_{2,l} + T^{el} \qquad \forall\, l \in L, l < |L| \tag{38}$$

$$t^{MS} \geq t^E_{ml} \qquad \forall\, m \in M, l = |L| \tag{39}$$

$$t^E_{ml} + t^{early}_l - t^{tardy}_l = \sum_{g \in G} T^{due}_g \cdot z_{gl} \qquad \forall\, l \in L, m = |M| \tag{40}$$

The last two constraints arise from the mold thickness change. This is controlled through a thickness code C^{th}_g of each group. The problem considered here has only two possible thicknesses that are here represented by thickness codes 0 and 1. In constraint (41) either of the positive slack-variables will be equal to one in the case of a change, depending if we change from thickness code 1 to 0 or vice versa (LHS equals 1 or -1). Constraint (42) restricts the total number of thickness changes.

$$\sum_{g \in G} \left(C^{th}_g \cdot z_{gl} - C^{th}_g \cdot z_{g,l+1} \right) = sl^+_l - sl^-_l \qquad \forall l \in L, l < |L| \tag{41}$$

$$\sum_{l \in L} \left(sl^+_l + sl^-_l \right) \leq N^{th}_{max} \tag{42}$$

$$z_{gl} \in \{0, 1\}$$

$$t^S_{ml}, t^E_{ml}, t^{MS}, w_{gg'l}, t^{early}_l, t^{tardy}_l, sl^+, sl^- \geq 0$$

The formulation for the aggregation (ordering of the groups) in Eqs. (29-42) can be complemented by a number of constraints regarding, for instance, special product grades that poses certain casting requirements. Some special grades with a low nickel content, for example, cannot be cast arbitrarily, but need to be preceded by suitable grades that can be called "wash grades" since their function is to clean up the equipment from undesired substances. Because of this complexity it is often beneficial to do the casting of special grades in one long sequence, i.e. let all the groups with the grade follow each other. This can be achieved by a few additional constraints. In the following example, let the special grade be 555 and the suitable wash grade 555*. The set of groups containing the grade is referred to by G^{555}.

Here, it should be recalled that the products are sorted such that all groups with the same grade (and thickness) are adjacent. In Eq. (43) all special grades are forced to be in subsequent slots. Constraint (44) focuses on the first special grade and positions the selected "wash grade" before it,

$$z_{gl} = z_{g+1,l+1} \qquad \forall\ l \in L, l < |L|,\ \forall\ g \in G^{555}, g+1 \in G^{555} \tag{43}$$

$$z_{555^*,l} = z_{g+1,l+1} \qquad \forall\ l \in L, l < |L|,\ \forall\ g \notin G^{555}, g+1 \in G^{555} \tag{44}$$

Additional information can be embedded through similar constraints, such as a set of groups can be ordered according to their size (number of products). Also, the formulation allows the addition of other critical constraints and many of these will, in fact, decrease the search space by incorporating partial decisions done off-line.

The problem given in Eqs. (29)-(44) is solvable for up to about 30 groups (on average 80-120 heats), and the solution gives the order in which the groups should be produced. Since the internal order of each group has already been optimized, we know after solving this step the complete order of production. Nevertheless, since this problem does not directly account for individual products, it provides only an upper bound for the makespan and may contain time gaps at the transition between the groups. Therefore, the solution can be improved by forming a scheduling problem with fixed sequencing and assignment and solve it as an LP.

3.5. LP-improvement problem

As discussed above, at this stage both the sequences and assignment of the products are known. Therefore, the formulation can be simplified by only writing constraints for subsequent heats. Solving this problem will eliminate those slack times, that were formed through the disaggregation into groups (see the improvement Δt in Fig. 7). At this step, most of the more specific issues are considered such as the hold-time, possible maintenance and changeover times.

The objective is to minimize the makespan, the total processing time of each product as well as hold-time violations. All elements needed for this formulation have already been presented, mainly in the section of scheduling the groups. The main difference is that now we only have three variables: makespan, starting times and hold-time slacks. All discrete variables have been eliminated. The objective was already given in Eq. (12). The makespan can be defined as in Eq. (13) and the timing constraints are similar to Eq. (14). Furthermore, the hold-time violations in Eq. (17) apply also to this problem. Continuous casting is also enforced by Eq. (18). Here, we do not need any precedence variables or constraints, so constraint (21) can be modified to control the electricity demand of EAF simply by removing the big-M terms. With these modifications the general mathematical model for this step is given by:

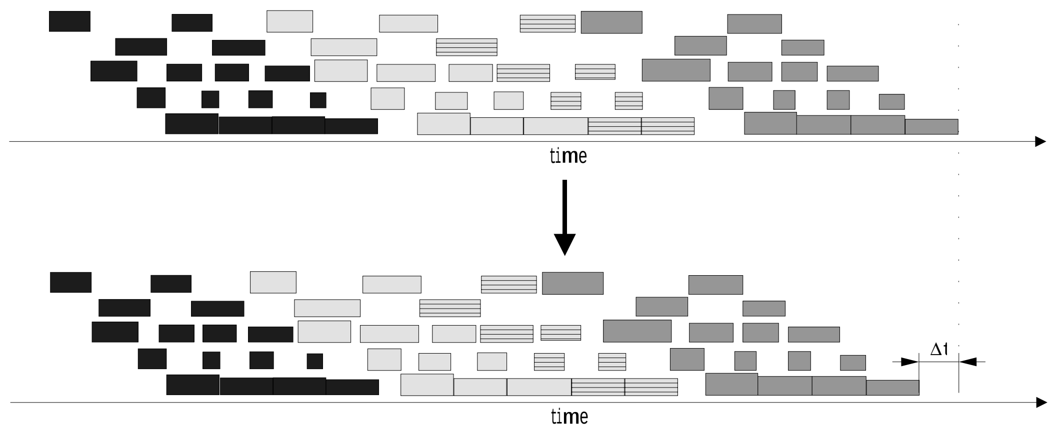

Figure 7. Improving the solution

$$min\ (12)$$

subject to

$$(13), (14), (17), (18), (21)$$
$$t^{MS}, t_{im}, sl \geq 0$$

The above problem is an LP that can be efficiently solved, typically within a fraction of a second. The main idea of this step is to improve the solution by closing some gaps that are formed by the grouping strategy and give a more realistic estimate of the total makespan and timing. Often this step results in a makespan which is 5-15% better than the one obtained in the previous step.

An additional optimization step can be added which "frees" the parallel equipment and determines an optimal assignment for those. In this example problem, the only parallel equipment are the two EAFs. The benefits of this step can be observed in cases where the parallel equipment act as a bottleneck, or when one of the equipment is unavailable due to maintenance. To solve this step, additional binary variables are needed for the EAF assignment. In this case, where we only have two parallel units, we would only need one binary variable per product, for which the value zero (0) refers to one of the units and value one (1) to the other. All constraints that do not consider the EAF stage are the same as in the previous LP. Thus, we only need to redefine a few constraints and introduce constraints for definition of the binary variable. The final step can be ignored if a detailed machine schedule is not needed, since it does not change the production order. Even though the previous LP provides both an upper bound and a feasible solution to this final MILP problem, solving the MILP may take a considerable amount of time, when the number of products (=number of binary variables) is large because of the big-M constraints.

4. Summary

The decomposition strategy is illustrated in Fig. 8. The modularity makes it possible to solve some parts of the problem independently without affecting the rest of the problem.

For instance, if a new product arrives, the grouping needs to be repeated only for those product grades that the new product belongs to. Furthermore, the second step of solving group schedules also needs to be performed only for the updated ones. The third step, which is computationally the most demanding, needs to be re-optimized in most cases, since changed operational parameters or due-dates etc. may either make the old solution infeasible or significantly worsen it and question the optimality.

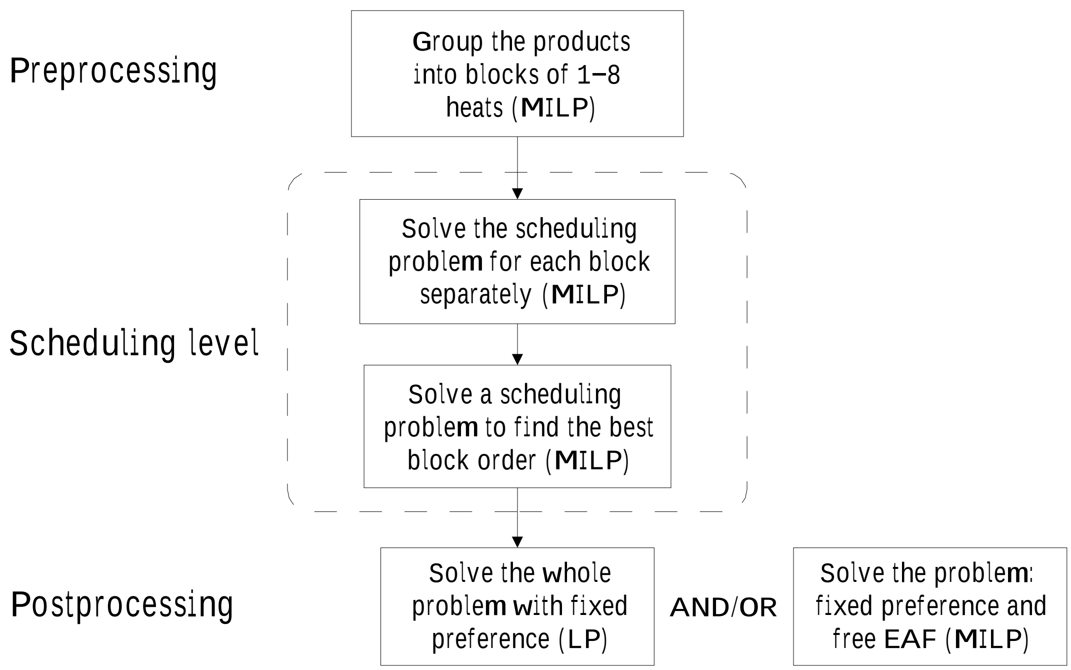

Figure 8. The decomposition strategy

The steel scheduling problem is very complex and obtaining a feasible solution can be very difficult for real industrial problems. Evaluating the optimality gap for the proposed strategy is difficult because there is no single MILP model from which a valid lower bound can be obtained. However, a rigorous lower bound for the makespan can be determined by considering the sum of the production and transition times, which corresponds to a valid lower bound. To obtain this theoretical lower bound for the makespan, we first need to determine the bottleneck stage and sum all production and setup times for this, and thereafter add a minimum production and setup time for the remaining stages. Since the optimal grouping strategy minimizes the casting setup times (section 3.2), we know that the lower bound for the makespan is valid. However, since we are not considering possible equipment conflicts, a physically realizable solution may have a higher objective.

5. Example

An illustrative example is first presented, followed by a set of large-scale scheduling problems.

5.1. Small scale example

To illustrate the steps of the decomposition strategy, a small problem with 12 products is considered. In this simplified example, we have two different grades (100,101), the grade 101 having three subgrades (101A, 101B and 101C). If these subgrades are placed into the same sequence they must be cast in the order 101A $\rightarrow$ 101B $\rightarrow$ 101C $\rightarrow$ 101. The grade, slab width and -thickness, casting and AOD times (minutes) of the products are given in Table 1 in random order.

Table 1. Unsorted orders

Grade	Width	Thick	Time	AOD
101	31.8	7.500	98.4	81.0
100	50.2	6.125	65.0	85.0
100	49.8	6.125	65.0	85.0
101	31.5	7.500	98.4	81.0
101B	37.6	6.125	92.0	83.0
100	49.2	6.125	65.0	85.0
101	49.8	7.500	60.0	81.0
101A	49.8	7.500	60.0	87.0
101	35.2	6.125	86.2	81.0
101A	34.2	7.500	81.1	87.0
101C	32.2	7.500	92.6	81.0
101B	27.5	7.500	110.0	83.0

In the first step, the products are presorted into product families by those properties that are critical for casting, namely the width, grade and thickness. The goal is to get the products as close as possible to a feasible casting order. As can be observed in Table 2, the presorting procedure results in 3 product families (separated by the thicker lines) and an upper bound of 8 for the number of groups considered. Here, the products are also numbered for reference.

Table 2. Presorted orders

Prod	Grade	Width	Thick	Time	AOD	Sequence
P1	100	50.2	6.125	65.0	85.0	1
P2	100	49.8	6.125	65.0	85.0	1
P3	100	49.2	6.125	65.0	85.0	1
P4	101B	37.6	6.125	92.0	83.0	2
P5	101	35.2	6.125	86.2	81.0	2
P6	101A	49.8	7.500	60.0	87.0	3
P7	101A	34.2	7.500	81.1	87.0	4
P8	101B	27.5	7.500	110.0	83.0	5
P9	101C	32.2	7.500	92.6	81.0	6
P10	101	49.8	7.500	60.0	81.0	7
P11	101	31.8	7.500	98.4	81.0	8
P12	101	31.5	7.500	98.4	81.0	8

The optimization problem for disaggregation into groups is then formulated based on the presorting. In this case, three MILP problems are formulated with the product families {1–3}, {4–5} and {6–12} (the products are divided according to their grade and thickness) using Eqs. (1)–(11). In this step all casting rules need to be considered. This is partly done by formulating a precedence matrix, $P_{ii'}$ that defines which products may follow each other. This matrix for products 6–12 is given in Table 3.

Table 3. Precedence matrix

	P6	P7	P8	P9	P10	P11	P12
P6	0	0	0	0	1	0	0
P7	0	0	0	1	0	1	1
P8	0	0	0	0	0	0	0
P9	0	0	0	0	0	1	1
P10	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	1
P12	0	0	0	0	0	0	0

From the table, it can be seen for instance that product 7 can be followed by products 9, 11 or 12 in a sequence. The solutions for the three product families are shown in Table 4. Note that families 1 and 2 involve only one group each, and family 3 has been partitioned into three groups. The products are given in the correct casting order within each group.

Table 4. Optimally grouped orders

Prod	Grade	Width	Thick	Time	AOD	Sequence
P1	100	50.2	6.125	65.0	85.0	1
P2	100	49.8	6.125	65.0	85.0	1
P3	100	49.2	6.125	65.0	85.0	1
P4	101B	37.6	6.125	92.0	83.0	2
P5	101	35.2	6.125	86.2	81.0	2
P7	101A	34.2	7.500	81.1	87.0	3
P9	101C	32.2	7.500	92.6	81.0	3
P11	101	31.8	7.500	98.4	81.0	3
P12	101	31.5	7.500	98.4	81.0	3
P6	101A	49.8	7.500	60.0	87.0	4
P10	101	49.8	7.500	60.0	81.0	4
P8	101B	27.5	7.500	110.0	83.0	5

The following step includes a detailed scheduling of each group using the jobshop formulation (12)–(28). This is done separately for the products {1–2–3}, {4–5}, {7–9–11–12}, {6–10} and {8}. Here we assume a fixed EAF time of 110 minutes per product of which 90 minutes is the electricity feed time (T^{el}). A constant LMF time of 17 minutes is further assumed. The setup time (transition time between equipment) is 5 minutes, except for the

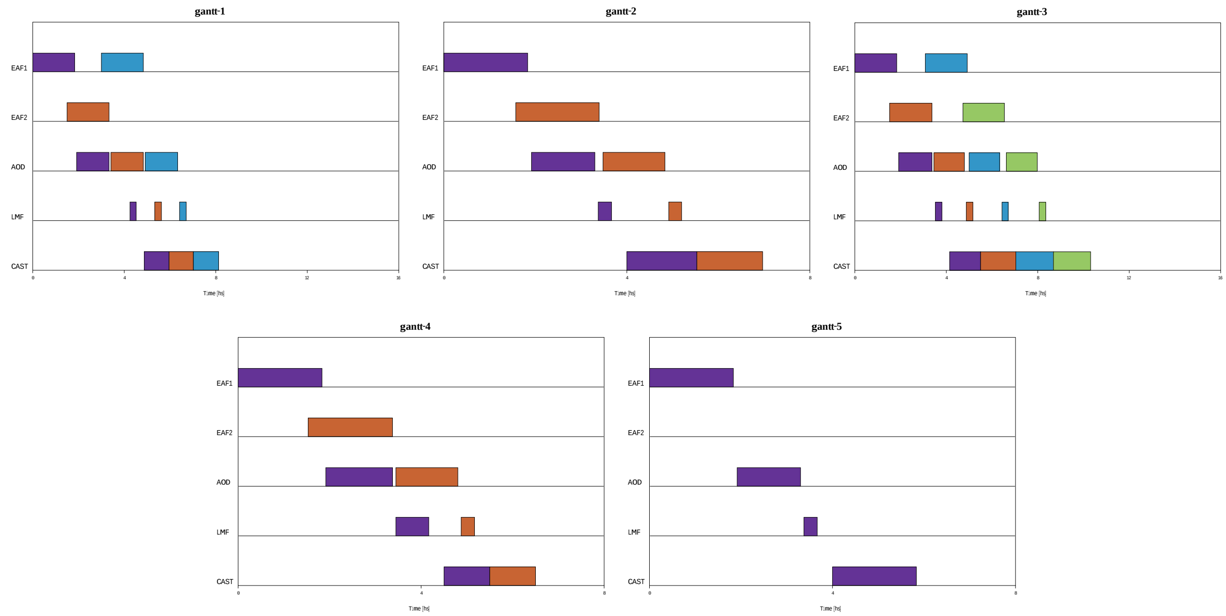

Figure 9. Solutions from the group scheduling problems

caster for which it is 20 minutes. No clean-up times are considered. The results of the 5 scheduling problems are presented as Gantt-charts in Fig. 9.

The results are mainly used for extracting information on equipment occupancy. That is, the start-time of the first job and the end-time of the last one at each equipment are registered, and used to formulate a slot-ordering problem in which the order between the groups is determined as illustrated in Fig. 5. In this example we do not have any special grades so the flowshop formulation in (29)–(42) is applied. In the caster, the setup time between groups is assumed to be 60 minutes and a thickness change adds additional 30 minutes to this. The optimal order of the groups is $2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 5$ and the slot-ordering problem results in a makespan of 26 hours 23 minutes. The final LP reduces the makespan to 24 hours 43 minutes (1483.7 mins), which represents a 6% improvement. This final solution is presented in the Gantt chart of Fig. 10. The complete solution procedure of this small example problem required about 1.5 CPU-s using GAMS/XPRESS-MP on a 667MHz PIII running Redhat Linux.

When evaluating the optimality for this example problem the casting turns out to be the bottleneck. The sum of the casting times is 973.7 minutes, which added with the minimum changeover times ($3 \cdot 60 + 90 = 270$ min) results in 1243.7 minutes for the casting process. If we add to this the EAF-time (110 min), LMF-time (17 min), the shortest AOD-time (81 min) and the required transition times ($2 \cdot 5 + 20 = 30$ min) we obtain the theoretically shortest possible makespan for the production 1481.7 minutes (24 hours 41.7 minutes), which differs only 2 minutes from the predicted solution of 24 hours 43 minutes! This means that the solution lies within 0.1% of the global optimum, which is an indication of that we have found a global or near global optimal solution to the small-scale example problem.

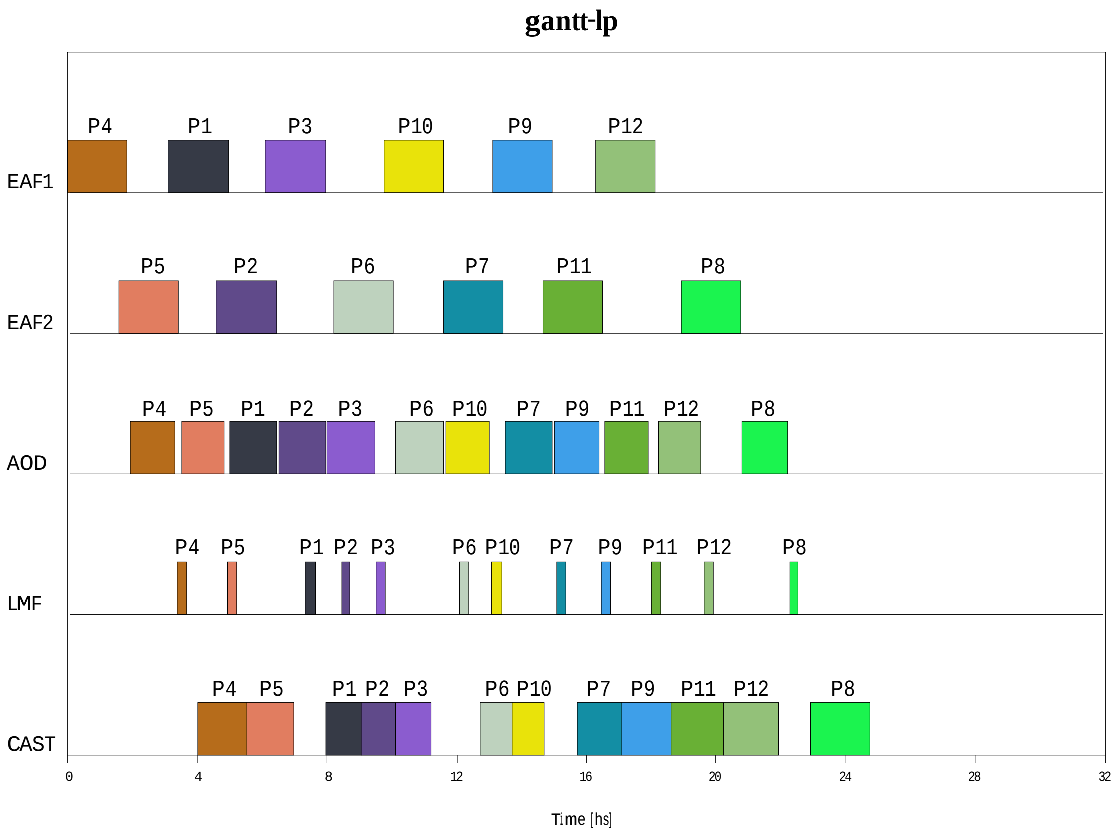

Figure 10. Final schedule of the small example problem

5.2. Industrial-scale examples

A number of real-size problems are considered in this section which are based on a one week schedule of a steel plant with about 80 orders. For these problems we report in Table 5 the number of grades and subgrades, the number of products and groups needed (casting sequences), the theoretical lower bound, the predicted makespan (days:hours:minutes) and the solution time for two different optimality tolerances in GAMS (5 and 10%). As can be seen from Table 5 the predicted schedules lie between 1% and 3% of the theoretically optimal makespan. These solutions are shorter than one week, which indicates a considerable improvement of the schedules.

Information of the subproblems for example problem 1 is displayed in Table 6. The problems were solved using GAMS-19.5/XPRESS-MP on a PIII, 667MHz PC running Redhat Linux. The termination criteria for each subproblem was an optimality tolerance of 5% (10%) or a maximum of 10000 CPU-s (166.7 mins). In most of the MILP problems either one of the grouping problems or the slot-ordering problem was terminated by the time limit, but in most of these cases, a good solution was found early in the search. Thus, the resulting MILP solution is very good even if optimality cannot be proved. The most important fact is that we can obtain a feasible solution to each of the problems, which is not trivial due to the large number of complicating production constraints.

Table 5. Results of industrial scale problems

Problem	Grades/ subgrades	Products	Groups	Lower bound	Makespan[1]	CPU-min[1]	Makespan[2]	CPU-min[2]
1	9/20	82	25	5:11:22	5:13:07	171.7	5:13:52	7.2
2	10/17	80	20	5:05:52	5:08:44	334.5	5:08:53	168.7
3	10/18	86	24	5:18:19	5:21:02	172.7	5:21:45	133.6
4	9/17	84	21	5:12:34	5:14:42	173.6	5:15:14	9.7
5	9/16	83	19	5:10:53	5:13:33	169.9	5:14:46	90.7

[1] optimality tolerance 5% [2] optimality tolerance 10%

Table 6. Results of example problem 1

Subproblem	Problems solved	Time range (CPU-s)	Avg. time (CPU-s)	Variables (0-1) (largest)	Constraints (largest)
1. Presorting	1	N/A	N/A	–	–
2. Grouping	12	0–174.7	19.4	1361 (160)	4304
3. Scheduling	25	0–15.54	1.26	283 (224)	544
4. Slot-ordering	1	–	10029.75	15977 (618)	15643
5. LP-improvement	1	–	0.10	658 (–)	815

The presorting does not require the solution of mathematical programming problems and is performed in a fraction of seconds. Table 6 reveals that possible bottlenecks in the computation can be found in subproblems 2 and 4 that are the most CPU-consuming problems. It is interesting to note that, for instance, formulating all 82 products in example 1 as a single scheduling problem using the model for subproblem 3, results in 74,000 equations and 34,000 variables of which more than 33,000 are discrete. Obviously such a large problem is virtually impossible to solve. Although the proposed strategy has not yet been fully applied on-site, simulations with Automod (Banks, 2000) have shown that a production increase of 2-4% can be expected compared to the present manual schedules that require 4-5 working days.

6. Conclusions

A decomposition algorithm for solving complex scheduling problems in the steel making industry has been presented. The algorithm relies on disaggregating the original problem and solving a sequence of smaller MILP problems for each group, followed by an MILP for all the groups. The rest of the solution steps need to be solved only once if the problem conditions do not change.

Numerical results have shown that the proposed approach can be successfully applied to industrial scale problems. While global optimality cannot be guaranteed, comparison with theoretical estimates indicate that the method produces solutions within 1-3% of the global optimum. It should be noted, however, that since we are dealing with a short-term scheduling problem, obtaining good feasible solutions is of more relevance than obtaining exact optimal solutions.

Finally, it should be noted that the general structure of the proposed approach naturally would allow the consideration of other types of problems, especially such where the physical problem provides a basis for decomposition. The modularity also provides increased flexibility by making it possible to customize and modify the solution strategy where needed.

Acknowledgments

The authors would like to thank the Center for Advanced Process Decision-making (CAPD) at Carnegie Mellon University and Kvaerner Metals for their financial support.

Nomenclature

Indices

g	index of group
i	index of products
l	index of slot
m	index of machines

Sets

G	set of groups
I	set of products
L	set of slots
M	set of machines
M^{EAF}	set of EAF

Parameters

C_1, C_2, C_3	cost coefficients in the objective function
C_g^{th}	thickness code for products (0 or 1)
Δ_w	maximal width difference between subsequent products within a group
$H_{i,max}$	maximal hold-time for product i
U	big-M constant with upper values for various purposes
M_{max}	maximal number of products in a group
N_{max}^{th}	maximum number of thickness changes
$P_{ii'}$	precedence matrix: 1 if product i' is allowed to follow product i, else 0
T^{el}	electricity feed time for one EAF
T_g^{due}	due (order) date for group g
T_{gm}^{D}	duration of processing group g in machine m
T_{gm}^{ST}	start time of group g in machine m
$T_{gg'}^{C,thk}$	changeover-time between different thicknesses
TM_{cst}	maintenance time between groups in the caster
T_{grp}^{C}	changeover-time between groups
T_m^{cl}	clean-up time of machine m

T_m^s	setup time for machine m
W_i	slab width of product i
τ_{im}	production time of product i on machine m

Variables

$Y_{ii'm}$	binary sequencing variable between products i and i' on machine m
$Y_{ii'}^{EAF}$	binary sequencing variable for products i and i' on the parallel EAF-units
Z_{im}^{EAF}	binary variable to couple each product i to one EAF-unit.
q_{ig}	continuous variable for last product in a group
sl	a slack variable for hold-time violations
sl_l^+	positive slack variables for mold change (slab thickness)
sl_l^-	negative slack variables for mold change (slab thickness)
t^{MS}	makespan
t_{im}	start-time of product i on machine m
t_l^{early}	earliness of the products in slot l
t_l^{tardy}	tardiness of the products in slot l
t_{ml}^E	end time of operation of machine m in slot l
t_{ml}^S	start time of operation of machine m in slot l
x_{ig}	binary variable to assign product i to group g
$y_{ii'g}$	continuous variable indicating that both products i and i' are in group g
z_g	continuous variable for the use of group g
$w_{gg'l}$	continuous variable for the sequence of groups g and g' in slots l and $l+1$
z_{gl}	binary variable assigning group g to slot l

References

Banks J. (2000). Getting Started with Automod. *http://www.autosim.com*

Birewar D.B. and Grossmann I.E. (1989). Incorporating Scheduling in the Optimal Design of Multiproduct Batch Plants. *Computers chem. Engng.*, **13**, pp. 141–161

Dorn J. and Doppler C. (1996). Expert Systems in the Steel Industry. *IEEE Expert*, **11**, 1, pp. 18–21

Dorn J. and Shams R. (1996). Scheduling high-grade steelmaking. *IEEE Expert*, **11**, 1, pp. 28–35

Elkamel A., Zentner M., Pekny J.F. and Reklaitis G.V. (1997). Decomposition Heuristic for Scheduling the General Batch Chemical Plant. *Engineering Optimization*, **28**, pp. 299–330

Finke G. and Jiang H. (1997). A variant of the permutation flow shop model with variable processing times. *Discrete Applied Mathematics*, **76**, 1-3, pp. 123–140

Ierapetritou M. and Floudas C.A. (1998). Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *I&EC Res*, **37**, pp 4341–4359.

Jain V. and Grossmann I.E. (2001). Algorithms for Hybrid MILP/CLP Models for a Class of Optimization Problems. *To appear in INFORMS Journal of Computing.*

Kondili E., Pantelides C. and Sargent R.W.H. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations - I. MILP Formulation. *Computers chem. Engng.*, **17**, pp. 211–227.

Lally B., Biegler L. and Henein A. (1987). A Model for Sequencing a Continuous Casting Operation to Minimize Costs. *Iron & Steelmaker*, **10**, pp. 53–70

Lee H.S., Murthy S.S., Haider S.W. and Morse, D.V. (1997). Primary production scheduling at steelmaking industries. *IBM Journal of Research and Development*, **40**, 2, pp. 231–252

Lee Y.H., Bhaskaran K. and Pinedo M. (1997). A Heuristic to Minimize the Total Weighted Tardiness with Sequence-Dependent Setups. *IIE Transactions*, **29**, pp. 45–52.

Mockus L. and Reklaitis G.V. (1996). Continuous Time Representation in Batch/Semi-Continuous Processes: Randomized Heuristic Approach. *Computers chem. Engng.*, **20**, pp. S1173–S1177.

Moon S. and Hrymak A.N. (1999). Scheduling of the Batch Annealing Process – Deterministic Case. *Computers chem. Engng.*, **23**, pp. 1193–1208

Pekny J.F. and Reklaitis G.V. (1998). Towards the Convergence of Theory and Practice: A Technology Guide for Scheduling/Planning Methodology. *AIChE Symposium Series*, **94**, 320, pp. 91–111.

Pekny J.F. and Zentner M.G. (1994). Learning to Solve Process Scheduling Problems: the Role of Rigorous Knowledge Acquisition Frameworks. *In Foundations of Computer Aided Process Operations, Rippin D.W.T., Hale J.C., Davis J.F. (Eds.), CACHE, Austin (TX)*, pp. 275–309.

Pinto J.M. and Grossmann I.E. (1995). A Continuous Time Mixed-Integer Linear Programming Model for Short Term Scheduling of Multistage Batch Plants. *I&EC Research*, **34**, pp. 3037–3051.

Pinto J.M. and Grossmann I.E. (1998). Assignment and Sequencing Models of the Scheduling of Process Systems. *Annals of Operations Research*, **81**, pp. 433–466.

Rich S.H. and Prokopakis G.J. (1986). Scheduling and Sequencing of Batch Operations in a Multipurpose Plant. *Ind. Eng. Chem. Process Des. Dev.*, **25**, pp. 979–988.

Roslöf J., Harjunkoski I., Westerlund T. and Isaksson J. (2001). Solving a Large-Scale Industrial Scheduling Problem Using MILP Combined with a Heuristic Procedure. *paper accepted to European Journal of Operational Research*

Rubin P.A. and Ragaz G.L. (1995). Scheduling in a Sequence Dependent Setup Environment with Genetic Search. *Computers Ops. Res.*, **22**, pp. 85–99.

Tang Lixin, Liu Jiyin, Rong Aiying and Yang Zihou (2000). A Mathematical Programming

Model for Scheduling Steelmaking Continuous Casting Production. *European Journal of Operational Research*, **120**, 2, pp. 423–435.

Shah N. (1998). Single- and Multisite Planning and Scheduling: Current Status and Future Challenges. *AIChE Symposium Series*, **94**, 320, pp. 75–90.

Zhang X. and Sargent R.W.H. (1996). The Optimal Operation of Mixed-Production Facilities – a General Formulation and Some Approaches for the Solution. *Computers chem. Engng.*, **20**, pp. S897–S904.

APPENDIX 1 - preordering algorithm

The preordering algorithm takes into consideration the casting rules and requirements and forms an initial casting order. Here, the main parameter of a product is its grade.

Rules:

- Only one grade and its subgrades can be in the same sequence
- Casting should be done in sequences of 1 to N products
- A sequence can only contain products with the same thickness
- Within each sequence, casting is done in decreasing width
- The due dates in a sequence should not vary more than within a specified tolerance
- The difference between casting speeds of subsequent products may be limited

Having a list of products and their main properties, the following algorithm can be applied.

Algorithm:

1. Order the products according to increasing speed
2. Order the products according to increasing duedate
3. Order the products according to decreasing width
4. Order the products according to increasing grade
5. Order the products according to increasing thickness