

# Data-Driven Approach to Grade Change Scheduling Optimization in a Paper Machine

Hossein Mostafaei, Teemu Ikonen, Jason Kramb, Tewodros Deneke, Keijo Heljanko, and Iiro Harjunkoski\*



Cite This: *Ind. Eng. Chem. Res.* 2020, 59, 8281–8294



Read Online

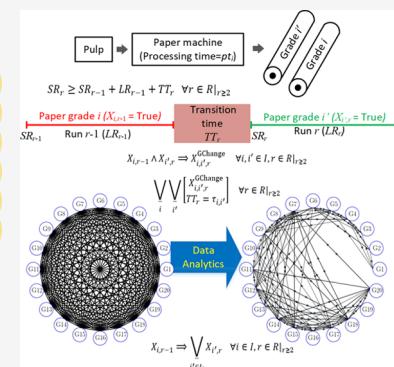
ACCESS |

Metrics & More

Article Recommendations

Supporting Information

**ABSTRACT:** This paper proposes an efficient decision support tool for the optimal production scheduling of a variety of paper grades in a paper machine. The tool is based on a continuous-time scheduling model and generalized disjunctive programming. As the full-space scheduling model corresponds to a large-scale mixed integer linear programming model, we apply data analytics techniques to reduce the size of the decision space, which has a profound impact on the computational efficiency of the model and enables us to support the solution of large-scale problems. The data-driven model is based on an automated method of identifying the forbidden and recommended paper grade sequences, as well as the changeover durations between two paper grades. The results from a real industrial case study show that the data-driven model leads to good results in terms of both solution quality and CPU time in comparison to the full-space model.



## 1. INTRODUCTION

Modern industrial production processes are typically complex, multistage processes, and much research has been conducted in order to improve efficiencies of these processes using analytical techniques, such as mathematical optimization. The application scope for optimization ranges from large-scale supply network planning to short-term production scheduling of individual machines. On the broader scope, Papageorgiou provides a review of supply chain management and optimization. For short to medium terms, Maravelias and Sung<sup>2</sup> review the methodologies on production planning and scheduling.

Scheduling can be considered as a key element of enterprise-wide optimization<sup>3</sup> and has been an area of intense research activity over the last 20 years. The major feature in the modeling of process-oriented scheduling problems is the time representation, which can be discrete or continuous.<sup>4,5</sup> This subject has been widely studied in many scheduling applications<sup>6–10</sup> and concerns the division of the time horizon of interest into a specific number of slots. The discrete-time representation divides the time horizon into slots of fixed duration, whereas the continuous-time representation relaxes such an assumption. The major advantage of discrete-time approaches against their continuous-time counterparts is their simplicity and a tighter LP relaxation.<sup>11</sup> As discrete-time approaches need many more time slots to account for accurate data, generating larger problems, the continuous-time representation will be used in this paper.

The objective of this work is to propose an optimization framework for single-stage production scheduling of paper

reels in a paper machine producing several paper grades with sequence-dependent changeovers. Changeovers occur when two different tasks are processed in the same unit in consecutive process operations. The changeovers are often associated with changing the operating conditions or with the cleaning of the equipment. There are many publications that have addressed the production scheduling of multiproduct plants with sequence-dependent changeovers.<sup>12–15</sup> Oh and Karimi<sup>12</sup> presented a novel mixed integer nonlinear programming (MINLP) formulation to solve the single machine economic lot-scheduling problem on a single multiproduct production facility. Liu et al.<sup>13</sup> developed a hybrid discrete-continuous-time mixed integer linear programming (MILP) model for the medium-term planning of single-stage multiproduct plant with one processing unit. In their model, the planning horizon is composed of a discrete number of weeks with a known demand, and each week is modeled by a continuous-time approach.

The problems, occurring in the paper industry, share similarities with many other production scheduling problems. For example, when changing the paper grade being produced on a paper machine, the production process does not stop, but the paper being produced during the transition may not meet

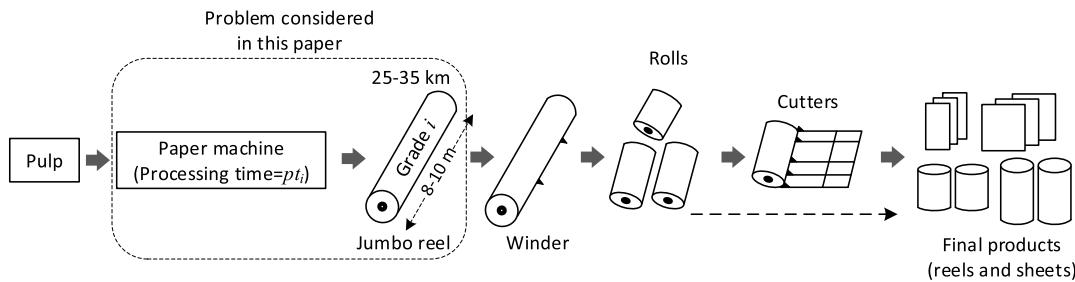
Received: December 18, 2019

Revised: February 28, 2020

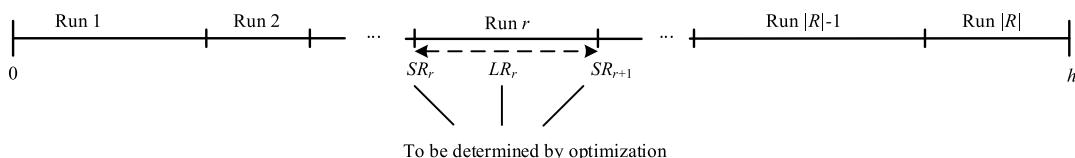
Accepted: March 13, 2020

Published: March 13, 2020





**Figure 1.** An overview of the papermaking process.



**Figure 2.** Continuous-time grid for proposed formulation.

customer quality criteria and is typically converted back to pulp for reuse. The time required to make the transition between the products depends on the starting and ending grades of the grade change. These sequence-dependent changeover times, or costs, are a common feature of paper production scheduling models.

Keskinocak et al.<sup>16</sup> developed a relatively comprehensive scheduling approach for paper production on a company scale, including order allocation to mills in addition to the actual production scheduling (including sequence-dependent setup times) and trimming on individual machines. Santos and Almada-Lobo<sup>17</sup> addressed the combination of paper production scheduling with sequence-dependent setup times and material flows in an integrated pulp and paper mill with a two-week planning horizon. Examples of sequence-dependent setup costs can also be found from outside the paper industry. For example, Tang et al.<sup>18</sup> presented a scheduling solution for the iron and steel industry using a multiple traveling salesman approach.

This article has been motivated by a real-world application from the Finnish paper making industry. The specific goal is to explore how historical data can be used to narrow down the decision space of a rigorous scheduling model addressing the sequence-dependent changeovers. The contribution of this work is threefold: first, we develop a continuous-time MILP model for the production scheduling of different paper grades in a real-world paper machine. The model allows the consideration of multiple intermediate due dates in the production process. Second, we use automatic methods to derive the changeover durations between two paper grades as well as define the forbidden and recommended changeovers. Third, we use the past “best practices” to guide the optimization in order to reduce the search space of the full-space MILP model. We show that by combining data analytics and rigorous scheduling models, more efficient decisions can be made while incorporating the rule-based operations into the optimization model.

In the next section, we give a brief description of the papermaking process. Section 3 develops a novel MILP formulation based on generalized disjunctive programming<sup>19–23</sup> for the production scheduling of different paper grades on a paper machine. Section 4 applies data analytics techniques to extract information from historical data to

support the full-space model developed in Section 3. Section 5 provides numerical examples to validate the proposed decision support tool, and conclusions are given in Section 6.

## 2. PROBLEM STATEMENT

Figure 1 presents a schematic overview of the papermaking process.<sup>24,25</sup> First, the paper machine is fed by pulp, the main ingredient of paper, to produce large rolls of paper called jumbo reels, which are often of fixed width for each paper machine. The jumbo reels then continue to another machine called winder to be cut into rolls of smaller dimensions. The rolls either go straight to the customer or are cut into sheets at the cutters. In the end, the final packaged products are dispatched to customers or distribution centers. In this paper, we will only focus on the production of different grades of jumbo reels that are typically around 8–10 m wide and 25–35 km long.

Because of market requirements and global competition, more paper grades are manufactured nowadays, and consequently frequent grade changes are inevitable in the paper machine. The grade change is a transition from one set of operating conditions (e.g., speed, stock flow, and steam pressure) to another and requires from few minutes to hours depending on the subsequent grades. As the paper machine is never idle, the paper produced during the transition may not be sellable due to quality reasons and is typically recycled again into the pulper. Therefore, minimizing the grade change transition can lead to a significant reduction in production loss and the efforts and energy needed to recycle off-spec waste.

Given the number of jumbo reels needed to be produced during a specific time horizon, we aim to sequence the production of paper grades in the paper machine while minimizing the grade change transitions. It is a challenging objective function to deal with because its linear relaxation is typically zero, often far away from the actual MILP solution. It is worth mentioning that, in practice, paper machine operators usually use a cyclic scheduling strategy to meet customer orders and consequently, historical process data can be used to reduce the domain of variables and constraints, which has a profound impact on the CPU time of the full-space MILP model.

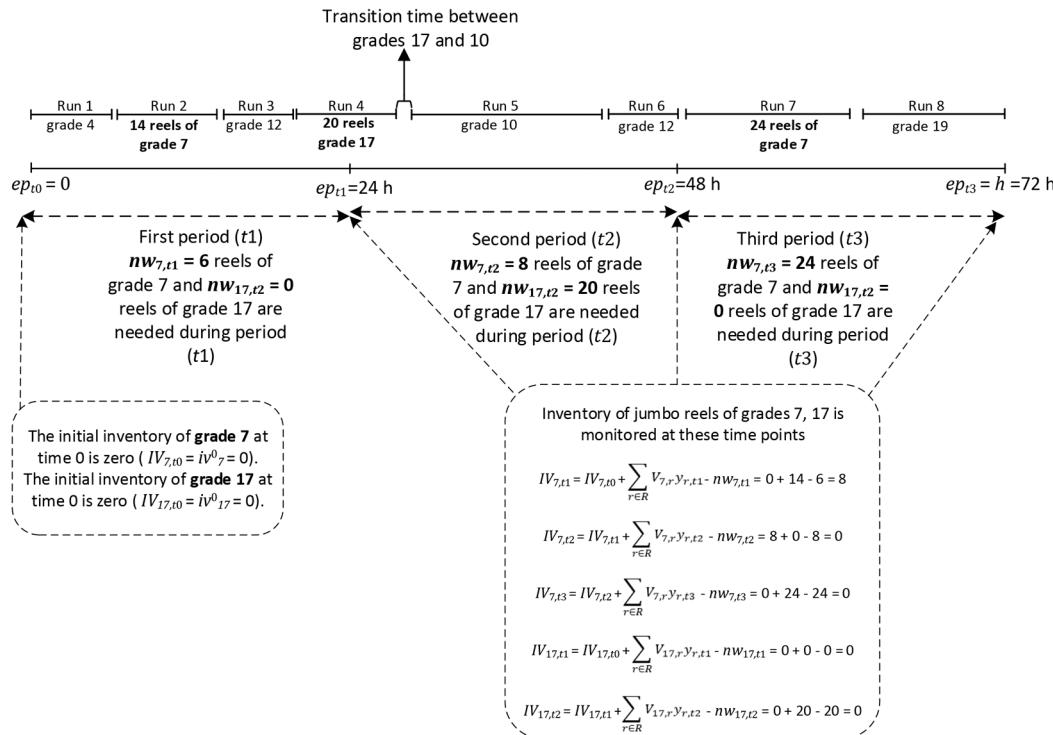


Figure 3. Simple example illustrating constraint 24.

### 3. MATHEMATICAL FORMULATION

We now present a continuous-time MILP formulation for production scheduling of jumbo reels in a paper mill. Such an MILP model (hereafter referred to as the full-space model) is capable of finding the optimal sequence of paper grades in a paper machine in a single step.

**3.1. Sequencing Production Runs.** We adopt a continuous-time formulation with the common reference grid given in Figure 2. It is made of a set of chronologically ordered  $r \in R = \{1, 2, \dots, |R|\}$  production runs. Therefore, run  $r$  must be performed after run  $r - 1$ . As the continuous variable  $SR_r$  represents the starting time of production run  $r$  and  $LR_r$  its duration, we will have eq 1. Equation 2 enforces that all production runs are located within the given time horizon  $h$ . Note that the start time of the first run is a known datum, that is,  $SR_1 = 0$ .

$$SR_r \geq SR_{r-1} + LR_{r-1} \quad \forall r \in R|_{r \geq 2} \quad (1)$$

$$\sum_{r \in R} LR_r \leq h \quad (2)$$

**3.2. Size and Length of Each Run.** Let the Boolean variable  $X_{i,r} = \text{True}$  indicate if run  $r$  is processing grade  $i$ , and let  $X_r^{no\ i} = \text{True}$  if no grade is being processed during run  $r$ . In the former case, the processing of grade  $i$  is associated with the production of a certain number of jumbo reels ( $V_{i,r}$ ). In the latter case, the variables  $V_{i,r}$  will be equal to zero. Notice that the disjunction in eq 3 is exclusive, meaning that at most one grade can be processed during each run  $r \in R$ .

$$\bigvee_{i \in I} [v_{\min,i} \leq V_{i,r} \leq v_{\max,i}] \bigvee [V_{i,r} = 0 \ \forall i \in I], \quad \forall r \in R \quad (3)$$

Equations 4 and 5 are obtained using the convex hull reformulation of disjunction 3. Note that the binary variable  $x_{i,r}$

that has a one-to-one relationship with the Boolean variable  $X_{i,r}$  can be assigned to  $X_{i,r}$  to transform disjunction 3 to an MILP representation.

$$\sum_{i \in I} x_{i,r} \leq 1 \quad \forall r \in R \quad (4)$$

$$v_{\min,i} x_{i,r} \leq V_{i,r} \leq v_{\max,i} x_{i,r} \quad \forall i \in I, r \in R \quad (5)$$

Equation 6 computes the length of a run  $r \in R$  processing grade  $i$ .

$$LR_r = \sum_{i \in I} p_{i,r} V_{i,r} \quad \forall r \in R \quad (6)$$

The number of production runs ( $|R|$ ) as a tuning parameter of the proposed model should be postulated by the user. Because the resulting problem for  $|R|$  is a relaxation of the one for  $|R| - 1$ , a typical procedure is to increase the cardinality of the set  $R$  by one until no better schedule is discovered. In case the number of predefined runs is greater than the optimal number of runs to be performed, the problem will result in some dummy runs ( $X_r^{no\ i} = 1$  or  $\sum_{i \in I} X_{i,r} = 0$ ). To reduce the symmetry, dummy runs should be confined to the end of ordered set  $R$ , see eq 7.

$$X_{r-1}^{no\ i} (\neg \bigvee_{i \in I} X_{i,r-1}) \Rightarrow X_r^{no\ i} (\neg \bigvee_{i \in I} X_{i,r}) \quad \forall r \in R|_{r \geq 2} \quad (7)$$

This can be reformulated into eq 8.

$$\sum_{i \in I} x_{i,r} \leq \sum_{i \in I} x_{i,r-1} \quad \forall r \in R|_{r \geq 2} \quad (8)$$

**3.3. Grade Change Transition and Forbidden Grade Changes.** When two consecutive runs process two different grades, a transition period occurs between the runs. If grade  $i$  has been processed at run  $r - 1$ , and run  $r$  is processing grade

$i'$ , there will be a transition time  $TT_r (= \tau_{i,i'})$  between the two runs. We have thus the following conditions

$$X_{i,r-1} \wedge X_{i',r} \Rightarrow X_{i,i',r}^{\text{GChange}} \quad \forall i, i' \in I, r \in R|_{r \geq 2} \quad (9)$$

$$\bigvee_i \bigvee_{i'} \left[ \begin{array}{l} X_{i,i',r}^{\text{GChange}} \\ TT_r = \tau_{i,i'} \end{array} \right] \quad \forall r \in R|_{r \geq 2} \quad (10)$$

The reformulation of logic proposition 9 and the convex hull reformulation of the disjunction 10 result in the following constraints

$$x_{i,i',r}^{\text{GChange}} \geq x_{i,r-1} + x_{i',r} - 1 \quad \forall i, i' \in I, r \in R|_{r \geq 2} \quad (11)$$

$$TT_r = \sum_{i \in I} \sum_{i' \in I} \tau_{i,i'} x_{i,i',r}^{\text{GChange}} \quad \forall r \in R|_{r \geq 2} \quad (12)$$

Note that constraints 11 and 12 can be replaced by a single inequality equation in constraints 13 by combining eqs 11 and 12 ( $TT_r = \sum_{i \in I} \sum_{i' \in I} \tau_{i,i'} x_{i,i',r}^{\text{GChange}} \geq \tau_{i,i'} x_{i,i',r}^{\text{GChange}} \geq \tau_{i,i'}(x_{i,r-1} + x_{i',r} - 1)$ ).

$$TT_r \geq \tau_{i,i'}(x_{i,r-1} + x_{i',r} - 1) \quad \forall i, i' \in I, r \in R|_{r \geq 2} \quad (13)$$

Note that the binary variable  $x_{i,i',r}^{\text{GChange}}$  in eqs 11 and 12 can be treated as a continuous 0–1 variable. This is simply because (i) when both  $x_{i,r-1}$  and  $x_{i',r}$  take one, the constraint 11 enforces  $x_{i,i',r}^{\text{GChange}}$  to be 1, and (ii) when at least one of the variables  $x_{i,r-1}$  and  $x_{i',r}$  is zero; consequently  $x_{i,i',r}^{\text{GChange}}$  must be zero. The latter is due to the fact that variable  $TT_r$  has a profound impact on the value of objective functions in eqs 29 and 30 which are minimized. Except Ex 2, we will use constraints 11 and 12 in our implementations.

The start time of run  $r$  should be after completing the end of the previous run  $r - 1$  and the subsequent changeover operation between the two runs

$$SR_r \geq SR_{r-1} + LR_{r-1} + TT_r \quad \forall r \in R|_{r \geq 2} \quad (14)$$

Because of prohibitively long transition times, some grades cannot be processed in consecutive runs. This condition is imposed by the logic proposition 15, which is reformulated into eq 16

$$\neg(X_{i,r-1} \wedge X_{i',r}) \quad \forall i, i' \in \{I|fs_{i,i'} = 1\}, r \in R|_{r \geq 2} \quad (15)$$

$$x_{i,r-1} + x_{i',r} \leq 1 \quad \forall i, i' \in \{I|fs_{i,i'} = 1\}, r \in R|_{r \geq 2} \quad (16)$$

**3.4. Inventory and Due Date Constraints.** Let us suppose that the horizon of interest is partitioned into a number of time periods  $t \in T = \{t0, t1, t2, \dots\}$  and  $ep_t$  is end of period  $t$  (note that  $ep_{t=0} = 0$ , see Figure 3). The aim is to produce a predefined number of reels  $i$  during each time period. Clearly, when a run starts, it should complete in exactly one of the time periods. If run  $r$  is completed during time period  $t$  ( $Y_{r,t} = \text{True}$ ), its completion time  $SR_r + LR_r$  should satisfy  $ep_{t-1} \leq SR_r + LR_r \leq ep_t$ . These conditions can be described through the following disjunction

$$\bigvee_t \left[ \begin{array}{l} Y_{r,t} \\ SR_r + LR_r \leq ep_t \\ SR_r + LR_r \geq ep_{t-1} \end{array} \right] \quad \forall r \in R \quad (17)$$

The convex hull reformulation of the disjunction gives rise to the following due dates timing constraints

$$\sum_{t \in T} y_{r,t} = 1, \quad \forall r \in R \quad (18)$$

$$SR_r + LR_r \leq \sum_{t \in T} ep_t y_{r,t} \quad \forall r \in R \quad (19)$$

$$SR_r + LR_r \geq \sum_{t \in T} ep_{t-1} y_{r,t} \quad \forall r \in R \quad (20)$$

Note that it is not necessary for a run that has started at period  $t$  to be completed in the same run. This is because eqs 19 and 20 are only applied to the completion time ( $SR_r + LR_r$ ) of a run  $r$  and not to the start time of that run ( $SR_r$ ), meaning that  $y_{r,t} = 1$  can only lead to  $ep_{t-1} \leq SR_r + LR_r \leq ep_t$ . For instance, run 4 processing grade 17 in Figure 3 completed at time period 2 starts during the first time period.

Alternative due date timing constraints can be developed using the big-M constraints in eqs 21–23. They describe that if run  $r$  is processing a grade, it should complete in one of the time periods and its completion time should satisfy  $ep_{t-1} \leq SR_r + LR_r \leq ep_t$ . If, run  $r$  is a dummy run, its completion time is relaxed. As constraints 18–20 result in a tighter linear relaxation, they will be used in our implementations.

$$\sum_{t \in T} y_{r,t} = \sum_{i \in I} X_{i,r} \quad \forall r \in R \quad (21)$$

$$SR_r + LR_r \leq ep_t + (h - ep_t)(1 - y_{r,t}) \quad \forall r \in R, t \in T \quad (22)$$

$$SR_r + LR_r \geq ep_{t-1} y_{r,t} \quad \forall r \in R, t \in T \quad (23)$$

The inventory balance at the end of each time period is defined in eq 24, stating that the number of reels of grade  $i$  at the end of time period  $t$  ( $IV_{i,t}$ ) is equal to the number of reels  $i$  at period  $t - 1$  plus the number of reels produced during time period  $t$  minus the numbers sent to the downstream processor ( $nw_{i,t}$ ). To satisfy the market demand on time, the number of jumbo reels of grade  $i$  sent to the downstream processor should be exactly equal to  $nw_{i,p}$ , the number of jumbo reels of grade  $i$  that needs to be produced during period  $t$ . The initial inventory of each grade is a known parameter as stated by eq 25.

$$IV_{i,t} = IV_{i,t-1} + \sum_{r \in R} V_{i,r} y_{r,t} - nw_{i,t} \quad \forall i \in I, t \in T|_{t \geq 1} \quad (24)$$

$$IV_{i,t} = iv_i^0 \quad \forall i \in I, t = 0 \quad (25)$$

Note that the bilinear term  $V_{i,r} y_{r,t}$  ( $= VY_{i,r,t}$ ) can simply be linearized by the following equations.

$$VY_{i,r,t} \leq v_{\max,i} y_{r,t} \quad \forall i \in I, r \in R, t \in T \quad (26)$$

$$VY_{i,r,t} \leq V_{i,r} + v_{\max,i}(1 - y_{r,t}) \quad \forall i \in I, r \in R, t \in T \quad (27)$$

$$VY_{i,r,t} \geq V_{i,r} - v_{\max,i}(1 - y_{r,t}) \quad \forall i \in I, r \in R, t \in T \quad (28)$$

Figure 3 is a simple example illustrating constraints 18–25. The time horizon  $h = 72$  h has been divided into three time periods  $t1, t2$ , and  $t3$ , with each period being composed of one day (24 h). There are a total of 8 production runs during the scheduling horizon, and each run is completed in exactly one

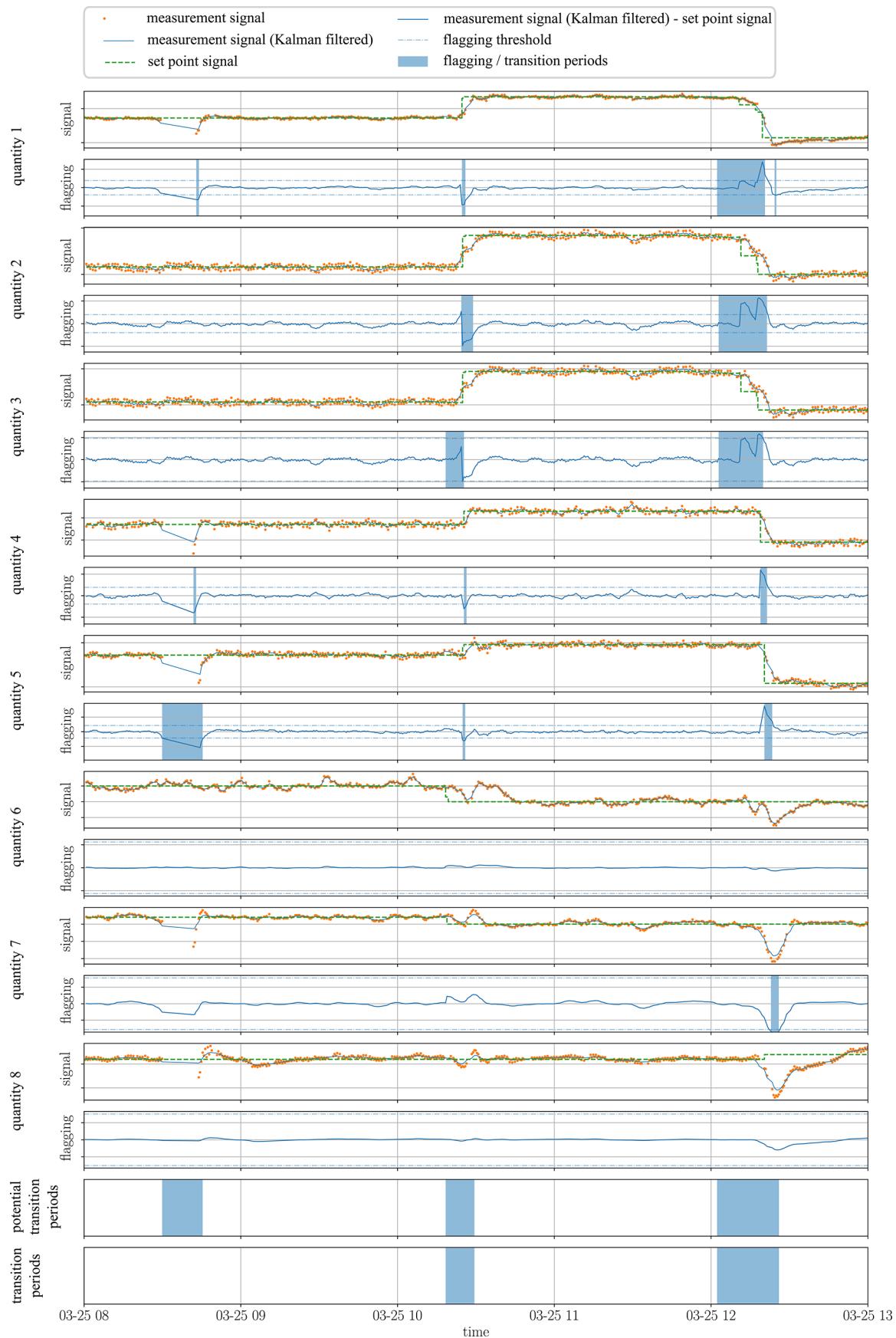


Figure 4. Example time window of transition period identification.

time period. Let us assume that from the planning level, we need 6, 8, and 24 jumbo reels of grade 7 and 0, 20, and 0 of grade 17 during periods  $t_1$ ,  $t_2$ , and  $t_3$ , respectively. As can be seen from Figure 3, 14 reels of grade 7 are produced during the first period and none during the second one. The initial inventory of grade 7 is zero, and during the first period 6 reels of it should be sent to the downstream processor. Thus, at time 24 h, the inventory of grade 7 will be 8 reels, which are used in the second period. The production of grade 7 is restarted at time point 48 h to satisfy 20 units of its demand needed during the third period. Run 4 processing grade 17 starts during period  $t_1$  and finishes within the second period  $t_2$ . Note however, we can only calculate the number of reels of grade 17 at the end of run 4, which happens within the second period, meaning that the inventory of grade 17 at the end of period  $t_1$  will be 0. In other words, the 20 units of grade 17 produced during run 4 will be only available at the end of the second period.

**3.5. Objective Function.** Two alternative objective functions are considered: minimum makespan in eq 29 and minimum cost in eq 30 including inventory cost, grade change transition cost, and the cost for executing a production run.

$$\min z = MS \quad (29)$$

$$\min z = \sum_{t \in T} \sum_{i \in I} ciV_{i,t} + \sum_{r \in R} \sum_{\substack{i \in I \\ r \geq 2}} ctT_r + \sum_{r \in R} \sum_{i \in I} crx_{i,r} \quad (30)$$

For the objective of minimization of makespan, the continuous variable  $MS$  should be the upper bound on the end time of last production run ( $r = |R|$ ), as given in eq 31.

$$MS \geq SR_{|R|} + LR_{|R|} \quad (31)$$

The full-space model for cost minimization consists of eqs 2, 4–6, 8, 11, 12, 16, 18–20, 24–28, and 30. For the makespan minimization, we replace eq 30 with eqs 29 and 31.

#### 4. DATA ANALYTICS

Historical process data contains information that is useful for defining the future operation of the process. The paper machine we consider in this work produces 20 groups of grades, each of which has a specific paper weight. Further, each of these groups contains various grades with different colors, brightness, coating material, and so on. However, in order to keep the size of the problem manageable, we only consider the groups of grades and refer to them simply as grades. In this section, we analyze a dataset recorded on the paper machine during one year of operation from the beginning of July 2017 to the end of June 2018. We extract two types of information from the dataset: (i) the number of pairwise grade transition occurrences and (ii) their average duration. We consider eight signals, which correspond to paper weight and moisture and coating weight measurements, in different parts of the paper machine.

A historical dataset of such process may either be labeled (i.e., the grade identifiers are recorder) or unlabeled. In this work, we consider the worse of these situations, that is, an unlabeled dataset. We demonstrate that useful information can still be extracted, even if the grade identifiers are unknown, using data clustering. In the following, we describe our signal processing approach to identify the grade transition periods (Section 4.1) and our clustering approach to summarize the information into useful tables (Section 4.2).

**4.1. Signal Processing.** The dataset contains both measurement and set point signals for each of the measured quantities. The set point signal is the target value at which the control system aims to retain the measurement signal. The sampling interval of the measurement signals is around 30 s. However, these samplings are not synchronized and contain idle periods, during which no measurements are recorded. We apply the Kalman filter to the measurement signal, in order to obtain a representative continuous signal. Figure 4 shows the measurement, its Kalman filtered and the set point signal for all eight measured quantities during a representative operation window of 5 h. For Kalman filtering and general data processing, we use the Python modules Pykalman<sup>26</sup> (version 0.9.5) and Pandas<sup>27</sup> (version 0.23.0), respectively.

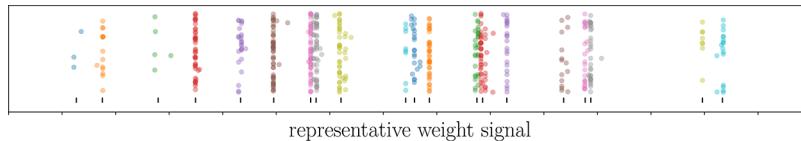
Our procedure to identify the transition periods from these signals is the following. First, for each measured signal, we flag time periods where the absolute difference between the Kalman filtered measurement and the set point signal exceeds a threshold of  $r_a = 5\%$  anomaly, that is, 1.96 times the standard deviation. In addition, we flag all time periods where the set point signal remains unchanged for less than a predefined minimum duration of a batch,  $t_{batch}$ . We use the value of  $t_{batch} = 30$  min in this work. The operators of the paper machine indicate this as the minimum time to produce a single grade. Figure 4 shows the flagged time periods for the measured quantities.

Second, the time periods at which any of the signals is flagged are marked as potential transition periods (see the second last subplot of Figure 4). Here, as the sampling of measurement signals are not synchronized, the missing values are forward filled. In reality, all signals might temporarily lay within the anomaly threshold of  $r_a$  soon after the set point change but exceed the threshold after a short time (cf., e.g., a controlled variable overshoot). Therefore, we merge two potential transition periods into one, if they are separated by less than a settling time of  $t_{settle} = 2$  min. In other words, the end of a potential transition period is marked as the time point, from which onward, no flagging occurs for 2 min.

Potential transition periods may include instances where a disturbance causes the flagging, instead of an intended grade change. In order to exclude such instances, we, finally, mark those of the potential transition periods, during which the set point signal of the representative paper weight signal changes, as transition periods (see the last subplot of Figure 4). The procedure avoids marking short-term disturbances falsely as transition periods. An example of such an instance is the first potential transition period in Figure 4. It is worth noticing that, in our approach, if a disturbance causes flagging within the settling time  $t_{settle}$  of a transition period, it is included in the transition period. Such disturbance, occurring soon after a grade transition, might well be caused by the transition. Therefore, its inclusion in the transition period is also appropriate.

Thus, our signal processing approach involves three hyperparameters, which need to be chosen before analyzing a dataset. The hyperparameters are the anomaly threshold  $r_a$ , the minimum batch duration  $t_{batch}$  and the settling time  $t_{settle}$ .

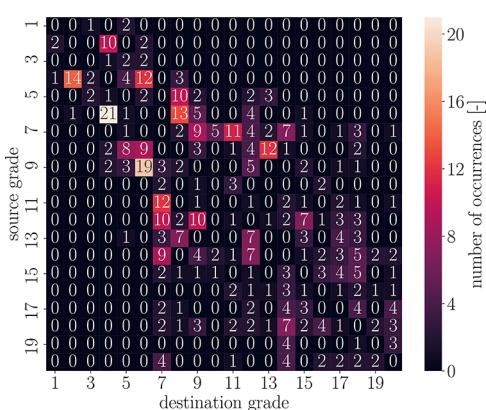
**4.2. Data Clustering.** As a result of the signal processing, described in the previous section, we obtain the start and end times of the transition periods. The time windows in between the transition periods are referred to as production periods. We enumerate the production periods between the transition periods and determine the average value for the representative



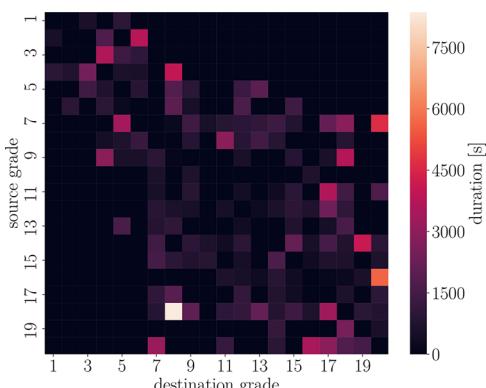
**Figure 5.** Clustering of production batches based on the average value of the representative paper weight signal. The centers of the clusters are marked by vertical black ticks. The colors indicate the points that belong to the same cluster.

paper weight during each period. We then cluster these values using  $k$ -means clustering<sup>28</sup> with a priori defined number of 20 clusters. The  $k$ -means clustering is implemented using the Python module Scikit-learn<sup>29</sup> (version 0.19.1). Figure 5 visualizes the clustered production periods, according to their representative paper weight, as well as the centers of the clusters.

We then assign each transition their source and destination grades. The grade identifiers are arranged in an increasing order of representative paper weight. Finally, we filter transitions with the same source and destination grade or unrealistically long duration ( $>3$  h). With this procedure, we are able to identify 498 grade transitions in the dataset. Figures 6 and 7 show matrices of transition occurrences from a source to a destination grade and their average durations, respectively.



**Figure 6.** Occurrences of changes from a source to a destination grade.



**Figure 7.** Average transition durations from a source to a destination grade.

**4.3. Constraints Derived from Data Analysis.** The main goal of this work is to show how data analytics methods can support the solution of large-scale scheduling models. Figure 8a shows all possible sequences between grades (some

sequences are forbidden), whereas Figure 8b is a directed graph based on the grade change table in Figure 6. It can be seen that historical data help to transform a dense graph into a sparse one, by omitting a significant number of links (sequences) between grades.

Taking into consideration the sparse graph, one can see that if, for instance, run  $r$  is processing grade 1, then the next run should process either grade 3 or grade 5. Now if we assume that  $I_i$  is a set of grades that can follow grade  $i$ , constraint 33 which is derived from logic proposition 32 can be added to the problem formulation.

$$x_{i,r-1} \Rightarrow \bigvee_{i' \in I_i} x_{i',r} \quad \forall i \in I, r \in R|_{r \geq 2} \quad (32)$$

$$x_{i,r-1} \leq \sum_{i' \in I_i} x_{i',r} \quad \forall i \in I, r \in R|_{r \geq 2} \quad (33)$$

By incorporating constraint 33 in the full-space model, we can guarantee that no forbidden sequence will occur during the production sequence. Note that constraint 33 also ensures that any dummy runs will place at the start of the production sequence. Moreover, constraint 33 enables us to reduce the domain of variable  $x_{i,i',r}^{\text{GChange}}$ , replacing  $i' \in I$  by  $i' \in I_r$ .

$$x_{i,i',r}^{\text{GChange}} \geq x_{i,r-1} + x_{i',r} - 1 \quad \forall i \in I, i' \in I_r, r \in R|_{r \geq 2} \quad (34)$$

$$TT_r = \sum_{i' \in I_r} \sum_{i \in I} \tau_{i,i'} x_{i,i',r}^{\text{GChange}} \quad \forall r \in R|_{r \geq 2} \quad (35)$$

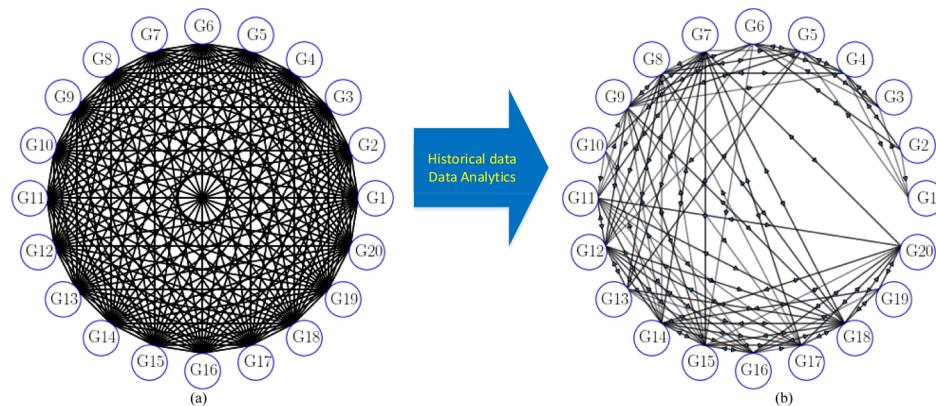
Note that again we can replace constraints 34 and 35 by a single inequality constraint 36. Except Ex 2, we will use constraints 34 and 35 in our implementations.

$$TT_r \geq \tau_{i,i'} \times (x_{i,r-1} + x_{i',r} - 1) \quad \forall i \in I, i' \in I_r, r \in R|_{r \geq 2} \quad (36)$$

Note that hereafter the full-space model with constraints 33–35 and without constraints 8 and 16 is called data-driven model. Therefore, the data-driven model for cost minimization consists of eqs 2, 4–6, 14, 18–20, 24–28, 30, and 33–35. For the makespan minimization, constraint 30 is replaced by 29 and 31.

## 5. COMPUTATIONAL RESULTS

To illustrate the capabilities of the proposed data-driven model, a case study based on an industrial size test case with different demand scenarios has been considered. In all cases, the paper machine produces 20 different types of paper grades (G1–G20), and the processing time of a jumbo reel of paper is 16.70 min ( $p_{ti} = 0.2778$  h). The MILP models were implemented in GAMS 24.9.1 and solved with CPLEX 12.7.1 running in parallel deterministic mode using up to four threads. The hardware consisted an Intel i5-7300U (2.60



**Figure 8.** Data analysis helps to omit a significant number of links between grades.

**Table 1. Number of Jumbo Reels Needed during the Scheduling Horizon**

	grades																			
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	
40	28	120	24	170	45	40	80					218	51	45			40	48	83	

GHz, 8 GB of RAM), running Windows 10, 64-bit operating system. The termination criteria were either a relative optimality tolerance of  $10^{-6}$  or a maximum computational time of 3 h for data-driven model and 5 h for the full-space model.

Remark: Note that, our observation on historical data indicates that no two consecutive runs have processed the same grade and this is why the diagonal elements of grade change table in Figure 7 are zero. This limitation, which is not considered in the full-space model, will be relaxed for the data-driven model in Ex 1 and Ex 4, replacing 0 with 1 in the diagonal of grade change table in Figure 6.

**5.1. Ex 1.** In this example, the aim is to manufacture a certain number of jumbo reels during a horizon of 300 h. The number of grades that should be produced is given in Table 1. The grade change transition duration between two grades (based on Figure 6) is given in Table 2. For a pair of grades that there is no link between them in the sparse graph in Figure 8b, the grade transition time is set to 3 h. Some grade transitions indicated with the cross mark ( $\times$ ) in Table 2 are forbidden. We assume that the minimum and maximum number of jumbo reels that can be produced during a process run are 4 and 240, respectively. Moreover, the fixed cost for performing a run is \$1000, and the cost resulting from a 1 h transition is \$10,000. The unit inventory cost is set to zero. Note also that in this example, we allow two consecutive runs to process the same grade, leading to replacing 0 with 1 in the diagonal of the occurrence matrix in Figure 6.

Here, we first solve Ex 1 for makespan minimization using data-driven model with both big-M and convex hull reformulations of due dates timing constraints. As can see from Table 1, 14 paper grades should be manufactured. Because each run can process one grade at a time, the minimum number of process runs  $|R|$  is fourteen ( $|R| \geq 14$ ). The results in Table 3 indicate that both big-M and convex hull need the same number of discrete and continuous variables as well as constraints for the same number of process runs. However, the convex hull reformulation performs quite well when we use constraints 34 and 35 instead of eq 36. The convex hull also exhibits a tighter LP relaxation, as expected. In

the remainder of this paper, we will use the convex hull type of due dates timing constraints.

Table 4 shows the model statistics and computational results for the full-space and data-driven models. For makespan minimization, it can be observed that both full-space and data-driven models yield the same solution with 14 runs and confirm it with 15 runs. However, the data-driven model is much faster, that is, about three times faster with 14 runs and six times faster for 15 runs. From the results, both formulations need the same number of binary variables, but there is a considerable difference in the number of continuous variables and constraints. The latter is because in the data-driven model, the set  $I$  in the domain of changeover time constraint 11 and variable  $x_{i,j,r}^{G\text{Change}}$  is replaced by  $I_b$ , that avoids duplicating the number of continuous variables and constraints.

For cost minimization, both models generate the same cost of \$41,961.1 with 14 runs in a reasonable CPU time. The data-driven model performs quite well with 15 processing runs, being able to improve the solution to \$40,213.8 in 231.2 s. This represents a one-order-of-magnitude time saving compared to the full-space model, which needs about 2 h to reduce the cost to \$40,213.8. To prove the solution optimality, both models with 16 runs are again solved. From Table 4, it follows that the same optimum is found with both models. However, the data-driven model is roughly two times faster than the full-space model.

Figure 9 depicts optimal schedules for Ex 1 using data-driven and full-space models. The sequences of grades in both models when minimizing cost are identical (see the middle part of Figure 9). When minimizing the makespan, the grade sequences are quite different, but correspond to the same makespan and total changeover transition times. This indicates that the problem can suffer from symmetric solutions.

**5.2. Ex 2.** Ex 1 is tackled again in Ex 2, but this time, we add an intermediate due date for demand at time 168. Therefore, Ex 2 considers two due dates for grades to be manufactured, with the first one being at 168 and the second one at 300. The number of grades that should be produced in each period is given in Table 5. All the values presented in Table 5 (also in Table 7) are generated based on historical data. In this

**Table 2.** Grade Change Transitions

grades	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
G1	0.000	3.000	0.161	3.000	0.278	3.000	x	x	x	3.000	3.000	x	x	x	3.000	3.000	3.000	3.000	3.000	
G2	0.148	0.000	3.000	0.477	3.000	1.047	3.000	3.000	3.000	3.000	x	x	3.000	3.000	x	3.000	x	3.000	3.000	
G3	0.000	3.000	0.000	1.012	0.371	0.283	3.000	3.000	3.000	3.000	x	x	x	3.000	3.000	3.000	x	x	x	
G4	0.261	0.209	0.686	0.000	0.170	0.163	3.000	1.111	3.000	3.000	3.000	3.000	3.000	3.000	3.000	x	x	3.000	3.000	
G5	0.001	0.385	0.198	0.000	0.249	0.249	3.000	0.497	0.256	3.000	3.000	0.358	0.547	3.000	x	3.000	3.000	x	3.000	
G6	0.000	0.269	3.000	0.260	0.058	0.000	3.000	0.558	0.150	3.000	x	0.458	3.000	x	0.387	3.000	x	3.000	x	
G7	3.000	3.000	3.000	3.000	0.948	3.000	0.000	0.059	0.383	0.153	0.240	0.259	0.301	0.368	0.212	3.000	0.594	0.804	x	
G8	x	3.000	3.000	3.000	0.119	0.187	0.341	3.000	0.000	0.227	3.000	0.810	0.253	0.405	0.239	3.000	0.001	0.228	x	
G9	x	3.000	3.000	3.000	0.819	0.159	0.163	0.270	0.029	0.000	3.000	0.158	3.000	0.234	3.000	0.161	1.028	x	x	
G10	x	3.000	x	3.000	3.000	0.164	3.000	0.189	0.000	0.004	3.000	3.000	3.000	3.000	0.197	3.000	3.000	3.000	3.000	
G11	3.000	x	3.000	x	3.000	0.151	3.000	0.246	3.000	0.000	0.111	3.000	0.096	0.258	3.000	1.013	0.404	3.000	0.476	
G12	3.000	x	x	3.000	3.000	0.252	0.170	0.153	3.000	0.141	0.000	0.078	0.171	0.265	0.196	0.659	0.282	3.000	3.000	
G13	3.000	3.000	x	3.000	0.450	3.000	0.226	0.288	3.000	3.000	0.106	0.000	3.000	0.201	3.000	0.138	0.436	3.000	3.000	
G14	3.000	x	3.000	3.000	x	0.392	3.000	0.269	0.181	0.104	0.269	3.000	0.000	0.600	0.149	0.421	0.201	1.144	0.269	
G15	3.000	3.000	3.000	x	3.000	0.419	0.258	0.200	0.244	3.000	0.139	3.000	0.461	0.000	0.098	0.194	0.177	0.001	0.183	
G16	x	3.000	x	3.000	x	3.000	3.000	3.000	3.000	0.181	0.151	0.082	0.244	0.108	0.000	0.041	0.067	0.161	1.566	
G17	x	3.000	3.000	x	3.000	x	0.298	0.533	3.000	3.000	0.316	3.000	0.215	0.101	3.000	0.000	0.179	0.001	0.276	
G18	3.000	x	3.000	x	3.000	0.227	2.322	0.566	3.000	0.281	0.317	0.600	0.222	0.377	0.188	0.911	0.000	0.258	0.215	
G19	3.000	3.000	x	3.000	x	x	x	x	3.000	3.000	0.318	3.000	3.000	3.000	0.702	0.000	0.156	0.421	0.000	
G20	3.000	3.000	x	3.000	3.000	0.897	3.000	x	3.000	0.342	3.000	0.255	3.000	0.967	0.814	0.485	0.421	0.000	0.000	

Table 3. Results of Ex 1 for Minimum Makespan Using Big-M and Convex Hull

	data-driven model with constraint 36				data-driven model with constraints (34 and 35)			
	big-M		convex hull		big-M		convex hull	
R	14	15	14	15	14	15	14	15
CPU (s)	64.50	380.90	40.23	305.73	337.0	2148.4	52.04	287.5
discrete variables	574	615	574	615	574	615	574	615
total variables	957	1021	957	1021	3095	3325	3095	3325
constraints	3978	4271	3978	4271	3958	4251	3958	4251
objective	289.48	289.48	289.48	289.48	289.48	289.48	289.48	289.48
LP-relaxation	1.00	1.00	286.68	286.68	1.00	1.00	286.68	286.68

Table 4. Computational Results for Ex 1

	R	CPU (s)	discrete variables	total variables	constraints	MILP	R <sub>gap</sub> <sup>a</sup> (%)
Makespan Minimization							
full-space model	14	198.2	574	5383	6753	289.48	0
	15	1018.6	615	5789	7261	289.48	0
data-driven model	14	64.9	574	3095	3958	289.48	0
	15	166.1	615	3325	4251	289.48	0
Cost Minimization (\$)							
full-space model	14	290.1	574	5382	6752	41,961.1	0
	15	7122.5	615	5788	7260	40,213.8	0
	16	8350.4	656	6194	7768	40,213.8	0
data-driven model	14	53.6	574	3094	3957	41,961.1	0
	15	231.2	615	3324	4250	40,213.8	0
	16	3200.15	656	3554	4543	40,213.8	0

<sup>a</sup>Relative gap.

Figure 9. Schedules for Ex 1 using data-driven and full-space models.

Table 5. Number of Jumbo Reels Needed during Each Period in Ex 2

time period	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	grades
T1																					
T2	40		120		104	25		80									58	45			

example, (i) no consecutive runs can process the same grade when applying the data-driven model, and (ii) eqs 11 and 12 are replaced by eq 13 in the full-space model and eqs 34 and 35 with 36 in the data-driven model.

Table 6 shows the model statistics and computational results of Ex 2 for the full-space and data-driven models. For makespan minimization, we again start by solving the models with 14 runs. As can be seen from Table 6, setting |R| = 14

Table 6. Computational Results for Ex 2

	RI	CPU (s)	discrete variables	total variables	constraints	MILP	$R_{gap}^a$ (%)
Makespan Minimization (h)							
full-space model	14	18,000	588	1291	7640	no solution	
	15	18,000	630	1376	8207	no solution	
	16	18,000	672	1461	8774	290.15	1.02
	17	18,000	714	1546	9341	289.93	1.09
Cost Minimization (\$)							
full-space model	16	18,000	672	1460	8773	58,536.1	76.08
	17 <sup>b</sup>	18,000	714	1545	9340	50,613.8	72.14
data-driven model	16	25.64	588	1291	4598	infeasible	
	15	316.2	630	1376	4931	infeasible	
	16	680.1	672	1461	5264	290.48	0
	17	1342.2	714	1546	5597	290.48	0
	18	9842.12	756	1630	5929	50,338.8	0

<sup>a</sup>Relative gap. <sup>b</sup>Run 17 is dummy at termination.

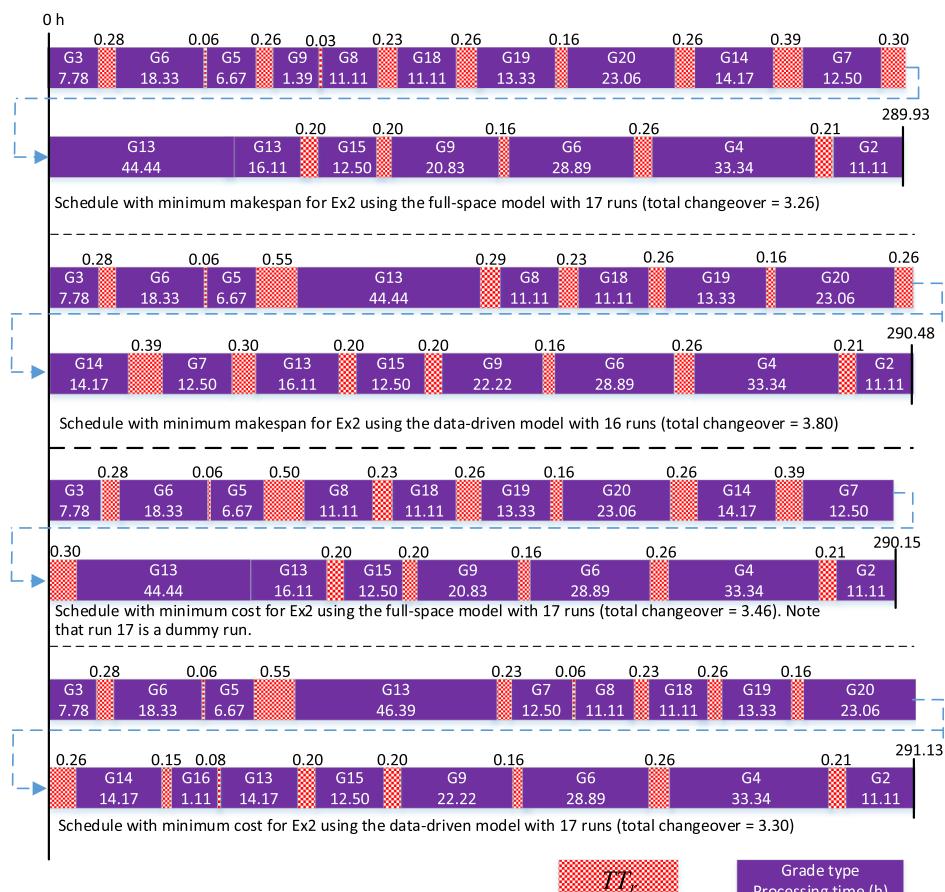


Figure 10. Schedules for Ex 2 using data-driven and full-space models.

leads to an infeasible MILP using data-driven model and returns no solution in 5 h using the full-space model. The same situations are observed when considering 15 process runs. Increasing the number of process runs to 16, both models yield roughly the same feasible solution. The data-driven model, however, performs very fast, enabling to find a makespan of 290.48 in 680 s, that is, one order of magnitude faster than the full-space model that cannot close the optimality gap within 5 h. With 17 runs, the data-driven model generates the same makespan encountered in 16 runs, but the CPU time increases

by a factor of two. However, the full-space model with 17 runs did find a slightly better solution (289.93 vs 290.48) up to maximum time limit. This is because unlike the data-driven model, the full-space model is allowed to process the same grade in two consecutive runs. This is apparent from Figure 10 where the full-space model for minimum makespan processes grade 13 in two consecutive runs 11 and 12 while the data-driven model processes the same grade in run 4 and run 11.

For the cost minimization, the full-space model generates a solution of \$58,536.1 with 16 runs but cannot close the

**Table 7.** Number of Jumbo Reels Needed during Each Period

time period	grades																			
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
T1																				
T2				4	160	72	71			72			100	18	51	21	50	69	32	25
T3				60	10	80		112	63	84	30	18	64							

**Table 8.** Computational Results for Ex 3

	R	CPU (s)	discrete variables	total variables	constraints	MILP	R <sub>gap</sub> (%) <sup>a</sup>
Makespan Minimization							
full-space model	19	36,000	817	8311	11,673	no solution	
	20	36,000	860	8699	12,241	no solution	
data-driven model	19	10,800	817	4723	7383	355.39	0.65
	20	10,800	860	4975	7776	355.39	0.65
Cost Minimization							
full-space model	19	36,000	817	8250	11,612	no solution	
	20	36,000	860	8698	12,240	no solution	
data-driven model	19	10,800	817	4722	7382	45,844.5	60.72
	20	10,800	860	4974	7775	46,844.5	61.56

<sup>a</sup>Relative gap.**Figure 11.** Best-found schedule for Ex 3 using the data-driven model with 24 runs.**Table 9.** Impact of the Number of Demand Points in Computational Effort of Data-Driven Model

T1	R  <sub>min</sub> <sup>b</sup>	CPU (s)	discrete variables	total variables	constraints	MILP (\$)	R <sub>gap</sub> (%) <sup>a</sup>
1	10	2.7	410	2174	2785	27,783.3	0
2	13	1939.7	546	3157	4464	38,058.3	0
3	14	10,800	602	3722	5677	42,677.7	65.87
4	14	10,800	616	4036	6537	46,375.0	36.46
5	12	2662.9	540	3722	6331	30,983.3	0
6	12	3312.8	552	3994	7071	33,161.1	0
7	12	10,800	564	4226	7811	34,275.0	52.34

<sup>a</sup>Relative gap. <sup>b</sup>|R|<sub>min</sub> is the minimum number of production runs needed to generate a feasible solution.

optimal gap in 5 h. On the other hand, the data-driven model generates almost 8% better solution with the same number of runs in 4143.4 s of CPU. Considering one more processing run (|R| = 17), the data-driven model improves the solution to \$50,338.8 and confirms it with 18 runs in 9842.12 s. The full-space model improves the solution by 13.5% but is again unable to close the optimality gap. Figure 10 shows the best

found schedules for minimum makespan and cost using the data-driven and the full-space models.

**5.3. Ex 3.** This example is hard to solve and considers three due dates, where  $ep_t = 168$  and  $ep_t = 336$  and  $ep_t = 450$ . The number of grades that should be produced in each period is given in Table 7. In this example, we set the computational time limit for the full-space model to 36,000 s and  $|R| \geq 19$ .

**Table 8** shows the model statistics and computational results of Ex 3. When choosing 19 runs, the solver is unable to return a feasible solution using the full-space model in the computational time limit of 36,000 CPUs. The data-driven model finds a makespan of 355.39 and a cost equals \$45,844.44 within 3 h of computational time. With 20 runs, the full-space model again fails to generate any feasible solutions for the problem. No improvement is observed using the data-driven model with 20 runs. The best-found schedules for Ex 3 using the data-driven model with 19 runs is depicted in **Figure 11**.

#### 5.4. Ex 4: Sensitivity Analysis on the Number of Due Dates.

In this example, we analyze the impact of the number of demand points (due dates) on the computational effort of the data-driven model. The aim is to produce 976 units of jumbo reels with minimum cost. We consider seven scenarios for the production of 976 jumbo reels during a time horizon of two weeks featuring multiple due dates ( $|T| = 1, 2, \dots, 7$ ). Data for each scenario is presented in Tables S1–S7 (see the Supporting Information). The initial inventory is set to zero for all grades, except for the grades 19 and 20, which is considered to be 70 and 60, respectively. **Table 9** shows the computational results for Ex 4. When there is only one due date ( $|T| = 1$ ), the data-driven model requires 10 production runs to generate the first feasible solution worth \$27,783.3. By adding an intermediate due date at time point 168 h, the minimum number of runs raises to 13 and the CPU time increases to 1939.7 s. When demands need to be considered in three due dates ( $|T| = 3$ ), the data-driven model cannot close the optimality gap within 3 h of CPU time. This also happens when the number of due dates is considered four and seven ( $|T| = 4, 7$ ). Overall, it can be seen from **Table 9**, both the solution quality and CPU time highly depend on how the required number of jumbo reels (967) is distributed among different due dates.

## 6. CONCLUSIONS

This paper has presented a data-driven decision support tool for the production scheduling of different paper grades in a paper machine. Some of problem constraints were first modeled using disjunctive programming and propositional logic and then translated into a MILP model that relies on continuous-time representation. We identified pairwise grade transition occurrences and their durations from a historical dataset. The former led to constraints appended to the full-space model, which reduce the domain of variables having a profound impact on the CPU time of the full-space model. The data-driven model was illustrated by four real-world test problems and was shown to be very efficient, allowing the solution of a large-scale problem where the full-space model fails to generate even a feasible solution. In future work, the proposed data-driven model will be generalized to tackle the long-term planning and scheduling problem arising at an integrated pulp and paper mill.

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.iecr.9b06907>.

Seven scenarios for the production of 976 jumbo reels during a time horizon of two weeks featuring multiple due dates ([PDF](#))

## ■ AUTHOR INFORMATION

### Corresponding Author

Iiro Harjunkoski – Department of Chemical and Metallurgical Engineering, School of Chemical Engineering, Aalto University, 02150 Espoo, Finland; ABB Power Grids Research, 68309 Mannheim, Germany; Email: [iiro.harjunkoski@aalto.fi](mailto:iiro.harjunkoski@aalto.fi)

### Authors

Hossein Mostafaei – Department of Chemical and Metallurgical Engineering, School of Chemical Engineering, Aalto University, 02150 Espoo, Finland; [orcid.org/0000-0002-4761-3037](https://orcid.org/0000-0002-4761-3037)

Teemu Ikonen – Department of Chemical and Metallurgical Engineering, School of Chemical Engineering, Aalto University, 02150 Espoo, Finland

Jason Kramb – UPM Communication Papers, 00100 Helsinki, Finland

Tewodros Deneke – Department of Computer Science, University of Helsinki, 00560 Helsinki, Finland

Keijo Heljanko – Department of Computer Science, University of Helsinki, 00560 Helsinki, Finland

Complete contact information is available at:

<https://pubs.acs.org/10.1021/acs.iecr.9b06907>

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

Financial support is gratefully acknowledged from the Academy of Finland project “SINGPRO”, Decision no. 313466 and No. 313469. Computational resources support used in process data mining is gratefully acknowledged from the CSC Finland, project no. 2000935.

## ■ NOMENCLATURE

### Sets/Indices

$I/i, i'$  paper grades

$R/r$  process runs

$T/t$  time periods

### Parameters

$h$  time horizon (h)

$ep_t$  upper extreme of period  $t$

$\tau_{i,i'}$  transition time between grades  $i, i'$

$pt_i$  the length of processing one jumbo reel of grade  $i$   
 $v_{\min,i}/v_{\max,i}$  min/max number of jumbo reels of grade  $i$  that can be produced during a run

$iv_i^0$  initial inventory of grade  $i$   
 $nw_{i,t}$  minimum number of jumbo reels of grade  $i$  that should be produced during period  $t$

$fs_{i,i'}$  1 if grade  $i'$  cannot be processed after grade  $i'$ ; 0 otherwise

$r_a$  anomaly threshold (signal processing)

$t_{\text{settle}}$  settling time (signal processing)

$t_{\text{batch}}$  minimum batch duration (signal processing)

$ci$  unit inventory cost (\$/unit)

$ct$  unit changeover cost (\$/h)

$cr$  cost for performing a production run (\$/run)

### Boolean variables

$X_{i,r}$  True if grade  $i$  is being produced during run  $r$

$X_{i,i',r}^{\text{Change}}$  True if grade  $i$  is changed to grade  $i'$  during run  $r$

$Y_{r,t}$  True if run  $r$  ends during period  $t$

**Binary variables**

- $x_{i,r}$  1 if grade  $i$  is being produced during run  $r$   
 $x_{i,i',r}^{\text{GChange}}$  1 if grade  $i$  is changed to grade  $i'$  during run  $r$  (can be treated as a continuous 0–1 variable, see Subsection 3.3)  
 $y_{r,t}$  1 if run  $r$  ends during period  $t$

**Discrete variable**

- $VY_{i,r,t}$  number of jumbo reels of grade  $i$  produced by run  $r$  during period  $t$

**Continuous variables**

- $LR_r$  duration of run  $r$  (h)  
 $SR_r$  start time of run  $r$  (h)  
 $TT_r$  transition time between run  $r - 1$  and  $r$   
 $V_r$  number of jumbo reels of grade  $i$  produced during run  $r$   
 $IV_{i,t}$  number of jumbo reels of grade  $i$  at the end of period  $t$   
 $MS$  makespan

**REFERENCES**

- (1) Papageorgiou, L. G. Supply Chain Optimisation for the Process Industries: Advances and Opportunities. *Comput. Chem. Eng.* **2009**, *33*, 1931–1938.
- (2) Maravelias, C. T.; Sung, C. Integration of Production Planning and Scheduling: Overview, Challenges and Opportunities. *Comput. Chem. Eng.* **2009**, *33*, 1919–1930.
- (3) Grossmann, I. Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. *AIChE J.* **2005**, *51*, 1846–1857.
- (4) Floudas, C. A.; Lin, X. Continuous-Time Versus Discrete-Time Approaches for Scheduling of Chemical Processes: A review. *Comput. Chem. Eng.* **2004**, *28*, 2109–2129.
- (5) Harjunkoski, I.; Maravelias, C. T.; Bongers, P.; Castro, P. M.; Engell, S.; Grossmann, I. E.; Hooker, J.; Méndez, C.; Sand, G.; Wassick, J. Scope For Industrial Applications of Production Scheduling Models and Solution Methods. *Comput. Chem. Eng.* **2014**, *62*, 161–193.
- (6) Castro, P. M.; Sun, L.; Harjunkoski, I. Resource–Task Network Formulations for Industrial Demand Side Management of a Steel Plant. *Ind. Eng. Chem. Res.* **2013**, *52*, 13046–13058.
- (7) Castro, P. M.; Rodrigues, D.; Matos, H. A. Cyclic scheduling of pulp digesters with integrated heating tasks. *Ind. Eng. Chem. Res.* **2014**, *53*, 17098–17111.
- (8) Mostafaei, H.; Castro, P. M.; Ghaffari-Hadigheh, A. A Novel Monolithic MILP Framework for Lot-sizing and Scheduling of Multiproduct Tree-Like Pipeline Networks. *Ind. Eng. Chem. Res.* **2015**, *54*, 9202–9221.
- (9) Song, Y.; Menezes, B. C.; Garcia-Herreros, P.; Grossmann, I. E. Scheduling and Feed Quality Optimization of Concentrate Raw Materials in the Copper Refining Industry. *Ind. Eng. Chem. Res.* **2019**, *57*, 11686–11701.
- (10) Castro, P. M.; Grossmann, I. E.; Zhang, Q. Expanding Scope and Computational Challenges in Process Scheduling. *Comput. Chem. Eng.* **2018**, *114*, 14–42.
- (11) Maravelias, C. T.; Grossmann, I. E. On the relation of continuous- and discrete-time state–task network formulations. *AIChE J.* **2006**, *52*, 843–849.
- (12) Oh, H.-C.; Karimi, I. A. Planning Production on a Single Processor with Sequence-Dependent Setups Part 1: Determination of Campaigns. *Comput. Chem. Eng.* **2001**, *25*, 1021–1030.
- (13) Liu, S.; Pinto, J. M.; Papageorgiou, L. G. A TSP-based MILP Model for Medium-Term Planning of Single-Stage Continuous Multiproduct Plants. *Ind. Eng. Chem. Res.* **2008**, *47*, 7733–7743.
- (14) Kopanos, G. M.; Laínez, J. M.; Puigjaner, L. An Efficient Mixed-Integer Linear Programming Scheduling Framework for Addressing Sequence-Dependent Setup Issues in Batch Plants. *Ind. Eng. Chem. Res.* **2009**, *48*, 6346–6357.
- (15) Erdirik-Dogan, M.; Grossmann, I. Planning Models for Parallel Batch Reactors with Sequence Dependent Changeovers. *AIChE J.* **2007**, *53*, 2284–2300.
- (16) Keskinocak, P.; Wu, F.; Goodwin, R.; Murthy, S.; Akkiraju, R.; Kumaran, S.; Derebail, A. Scheduling Solutions for the Paper Industry. *Oper. Res.* **2002**, *50*, 249–259.
- (17) Santos, M. O.; Almada-Lobo, B. Integrated Pulp and Paper Mill Planning and Scheduling. *Comput. Ind. Eng.* **2012**, *63*, 1–12.
- (18) Tang, L.; Liu, J.; Rong, A.; Yang, Z. A Multiple Traveling Salesman Problem Model for Hot Rolling Scheduling in Shanghai Baoshan Iron & Steel Complex. *Eur. J. Oper. Res.* **2000**, *124*, 267–282.
- (19) Balas, E. Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM J. Algebr. Discrete Methods* **1985**, *6*, 466–486.
- (20) Castro, P. M.; Grossmann, I. E. Generalized Disjunctive Programming as Asystematic Modeling Framework to Derive Scheduling Formulations. *Ind. Eng. Chem. Res.* **2012**, *51*, 5781–5792.
- (21) Grossmann, I. E.; Trespalacios, F. Systematic Modeling of Discrete-Continuous Optimization Models through Generalized Disjunctive Programming. *AIChE J.* **2013**, *59*, 3276–3295.
- (22) Mostafaei, H.; Castro, P. M. Continuous-Time Scheduling Formulation for Straight Pipelines. *AIChE J.* **2017**, *63*, 1923–1936.
- (23) Mostafaei, H.; Harjunkoski, I. Continuous-time Scheduling Formulation for Multipurpose Batch Plants. *AIChE J.* **2019**, *66*, No. e16804.
- (24) Keskinocak, P.; Wu, F.; Goodwin, R.; Murthy, S.; Akkiraju, R.; Kumaran, S.; Derebail, A. Scheduling Solutions for the Paper Industry. *Oper. Res.* **2002**, *50*, 249–259.
- (25) Correia, M. H.; Oliveira, J. F.; Ferreira, J. S. Reel and Sheet Cutting at a Paper Mill. *Comput. Oper. Res.* **2004**, *31*, 1223–1243.
- (26) Duckworth, D.; Lipták, G.; Haessig, P.; Werner, N. Pykalman, 2012 [Online] Available at: <https://github.com/pykalman/pykalman/>.
- (27) McKinney, W. Data Structures for Statistical Computing in Python. s.l. *Proceedings of the 9th Python in Science Conference*, 2010.
- (28) Lloyd, S. Least Squares Quantization in PCM. *IEEE Trans. Inf. Theor.* **1982**, *28*, 129–137.
- (29) Pedregosa, F.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.