

Scheduling of a pipeless multi-product batch plant using mixed-integer programming combined with heuristics

Sebastian Panek¹, Sebastian Engell¹, Cathrin Lessner²

¹ University of Dortmund

Process Control Lab (BCI-AST), University of Dortmund, Dortmund 44221, Germany

² Axxom Software AG

Axxom Software AG, Munich, Germany

Abstract

The work presented here deals with short-term scheduling in the chemical industry. It proposes an alternative model to STN and RTN in the form of a MILP formulation which describes the plant at a lower level of detail. The aim is to reduce the complexity of the model and thus to increase the solution efficiency. The main simplification compared to STN/RTN is the that product stocks and mass balance constraints are ignored because discrete fixed-size batches are assumed.

The proposed MILP formulation is used for the scheduling of a real-life example provided by a customer of Axxom, a lacquer production in a pipeless plant. Three types of recipes that involve cycles, rigid timing constraints between individual operations, and parallel allocations of stationary and mobile units are considered. Several dozens of production orders of different lacquer types in various colors with hundreds of operations have to be scheduled. A special solution procedure combines the solution of the MILP problem in two separate steps with three different heuristics which further reduce the model complexity. Several problem instances of the lacquer production are solved to demonstrate the successful application.

Keywords: supply chain management, scheduling, planning, multiproduct batch processing, pipeless plant

1. Introduction

Research efforts in the past 10 years in the area of batch plant scheduling (batch sizing, resource allocation, sequencing) have led to powerful general modelling frameworks as STN (Kondili et al., 1993; Shah et al., 1993) and RTN (Schilling et al., 1996). They enable the user to express a large variety of constraints that might arise in the processing industry, such as capacity constraints on units and products, timing constraints between operations, sequence-dependent changeover procedures, various storage policies, complex production recipes etc. Within both frameworks, the representation of time turned out to be crucial for the performance of the solvers of the mathematical models. This led to continuous time STN/RTN representations with uniform and non-uniform

grids (e.g. Ierapetritou et. al., 1998). Many recent publications focused on this problem by proposing improved modelling and solution approaches. For solver performance, it is usually beneficial to reduce the complexity of the model in terms of the number of equations and variables. This is often done by refining and adapting the constraints to exploit special properties of the plant and the production scheme. A discussion of various representations of time can be found in (Floudas et. al., 2004).

We refer to two recent continuous-time modelling approaches here. The first one is slot-based and suitable for network-based production schemes involving mass balances and product flows (Pinto et. al. 1994). The second one is based on order-oriented sequential processes without time slots (Manne, 1960; Cerda et. al. 1997).

The work presented here is strongly related to the latter approach. It aims at overcoming the difficulties mentioned above by an order-based modelling scheme. An alternative to STN and RTN in the form of a different MILP formulation which describes the plant at a lower level of detail is proposed. The aim is to simplify the representation of time and thus to increase the solution efficiency. An explicit representation is often avoided by establishing a minimal set of event points without predefined order. Sequencing decisions are encoded using dedicated binary variables which correspond to typical binary disjunctions arising in all sequencing problems. This leads to an efficient model formulation in which the resource assignment and the sequencing are decided in parallel. The main simplification compared to STN/RTN is that no product stocks and mass balance constraints are considered. Discrete fixed-size batches are assumed. This work also suggests a two step solution procedure. It combines the solution of the MILP problem in two separate steps with three different heuristics which further reduce the model complexity.

The remainder of this paper is structured as follows: after stating the problem the modelling approach is described in detail. In the following section the solution procedure is discussed and the results of numerical experiments are given.

2. Problem Statement

The proposed MILP formulation is used for the scheduling of a real-life example provided by a customer of Axxom, a lacquer production in a pipeless plant. The plant facilities consist of 5 mobile mixing vessels and 9 stationary processing units. The stationary units are: pre-dispersion line, main dispersion line, special pre-dispersion unit, dose distributor, two identical dose spinners, two identical filling stations and a laboratory for quality checks. Two mixing vessels have capacities of 19,000 litres, the other three vessels can hold 20,000 litres. The plant topology with traffic on limited routes, possible collisions and different distances is not considered here.

Three basic recipes for lacquers, each with 6-8 operations are given. The recipes involve cycles, rigid timing constraints between individual operations, and parallel allocations of stationary and mobile units. Three different types of constraints involve either 1) starting or 2) ending or 3) ending and starting dates of individual operations within the recipes. Each of these constraints establishes a link between two operations and forces them to either start or to end within a time window defined by the constraint. The cycles in the production comprise two steps: dosing and quality checks. The termination of a cycle depends on the result of the last quality check. During the processing steps, no

material flows from and to the vessels are considered such that the amount of product in the batch remains constant. Raw materials and the storage for end products are unlimited and available at any time. The market demand is represented by 29 production orders for different products and with irregular release and due dates spread over several months. The objective is to minimize the overall production cost which consists of penalties for missing the due dates. The recipe for a standard metallic lacquer is depicted in Fig. 1. The main processing steps are: assignment of a mobile mixing vessel, dosing, quality check, correction, second quality check and filling. The other recipes have a similar structure but involve two additional pre-processing steps. The mixing vessel holds the product during all steps and is thus allocated in parallel.

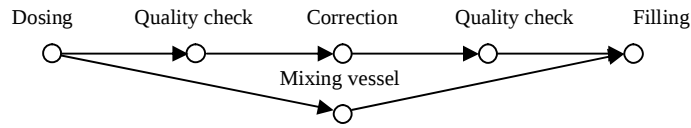


Figure 1. The recipe graph of metallic lacquer

3. Modelling Approach

3.1 Simplifications and Assumptions

Some simplifications with respect to the original problem are made. As the demanded product amounts and vessel capacities are quite similar and do not vary much, we assume equal mixing vessels with sufficient capacities to hold one batch of each product. Another simplification is related to the cycles. The results of the quality checks in the laboratory are a-priori unknown. Since the following steps depend on them, the overall number of processing steps is also unknown. We avoid stochastic modelling and assume that the first quality check always fails and the second one always succeeds. Thus, the loop in the recipe is always taken twice and we get 29 batches of different lacquer types in various colours with 202 operations in total that have to be scheduled. Since the lab is an unlimited resource quality checks may take place at any time arbitrarily often, but require a fixed amount of time. This allows us to remove all quality checks and the lab from the model. Appropriate timing constraints are established to make sure that the necessary minimal amount of time passes between the dosing, correction and filling operations. This reduces the total number of operations to 144 and the number of stationary units to 8.

3.2 The MILP Formulation

The key idea of the formulation discussed here was originally proposed by (Manne, 1960) and developed further in (Cerda et. al. 1997). The main contribution is the consideration of additional constraints, simplifications and more complex recipes. The approach uses continuous event points for the starting and the ending dates of the operations instead of time grids and slots. For each pair of operations o and o' that can be processed on the same unit m , a binary variable is introduced to ensure mutual exclusion in resource utilization. It is set to 1 if o precedes o' and set to 0 otherwise. Since resource assignment is also subject of optimization, we define that this binary

variable is 0 if either o or o' are assigned to an alternative unit (and no collision can occur). The resource assignment is modelled by additional binary variables. Starting and ending dates of operations are expressed by real variables. If an operation can be processed on alternative units, further copies of these variables are introduced for each possible unit.

3.2.1 Sets, Parameters and Constants

The following sets are used: jobs J , operations O , operations of job j : $O_j \subseteq O, j \in J$, units (machines) M , alternative units for operation o : $M_o \subseteq M$, operations to be executed on unit m : $O_m \subseteq O$, operations of job j that can be processed on unit m : $O_{jm} \subseteq O, j \in J, m \in M$.

Release and due dates of jobs: rel_j, due_j , recipes of jobs: $rec_j, j \in J$. Minimum and maximum operation durations on machine m : $dur_{om}^-, dur_{om}^+, m \in M, o \in O_m$.

Three types of timing constraints between starting and ending dates of operations are defined: ss (starting-starting constraints), es (ending-starting) and ee (ending-ending). Every constraint defines a time window $[t_{oo'm}^-, t_{oo'm}^+]$ which limits the difference of the corresponding event points (e.g. the end of the dispersion and the beginning of the dosing). The time windows for the three types of constraints are denoted by:

$t_{oo'm}^{ss-}, t_{oo'm}^{ss+}, t_{oo'm}^{es-}, t_{oo'm}^{es+}, t_{oo'm}^{ee-}, t_{oo'm}^{ee+} \in \mathbb{R}; o, o' \in O_m$. The order of subsequent operations of a job is established using es -constraints. A safe time horizon for Big-M constraints is $H \in \mathbb{R}$.

3.2.2 Variables

Machine allocation variables: $a_{om} \in \{0,1\}, m \in M, o \in O_m$. Variables representing starting and ending dates of operations: $s_{om}, e_{om} \in \mathbb{R}, m \in M, o \in O_m$. Variables to encode precedence relations of operations: $p_{oo'm} \in \{0,1\}, m \in M, o, o' \in O_m$.

3.2.3 Equations

Every operation must be processed on one machine and started after the release date of

$$\text{the job: } \forall o \in O: \sum_{m \in M_o} a_{om} = 1, \forall j \in J, \forall o \in O_j: \sum_{m \in M_o} s_{om} \geq rel_j,$$

Starting and ending dates of operation o on machine m are set to 0 if another machine is assigned to o : $\forall m \in M, \forall o \in O_m: s_{om} \leq H \cdot a_{om}, e_{om} \leq H \cdot a_{om}$. Minimum and maximum

operation durations: $\forall m \in M, \forall o \in O_m: s_{om} + dur_{om}^- \cdot a_{om} \leq e_{om}, s_{om} + dur_{om}^+ \cdot a_{om} \geq e_{om}$.

Timing constraints between operations: $\forall j \in J, \forall o, o' \in O_j$:

$$\begin{aligned} \sum_{m \in M_o} s_{om} + t_{oo'm}^{ss-} &\leq \sum_{m \in M_{o'}} s_{o'm}, \sum_{m \in M_o} s_{om} + t_{oo'm}^{ss+} \geq \sum_{m \in M_{o'}} s_{o'm}, \sum_{m \in M_o} e_{om} + t_{oo'm}^{es-} \leq \sum_{m \in M_{o'}} s_{o'm}, \\ \sum_{m \in M_o} e_{om} + t_{oo'm}^{es+} &\geq \sum_{m \in M_{o'}} s_{o'm}, \sum_{m \in M_o} e_{om} + t_{oo'm}^{ee-} \leq \sum_{m \in M_{o'}} e_{o'm}, \sum_{m \in M_o} e_{om} + t_{oo'm}^{ee+} \geq \sum_{m \in M_{o'}} e_{o'm}. \end{aligned}$$

If o and o' are processed on m and o precedes o' then o' must not precede o :

$$\forall m \in M, \forall o, o' \in O_m: p_{oo'm} + p_{o'om} \leq a_{om}, p_{oo'm} + p_{o'om} \leq a_{o'm}, p_{oo'm} + p_{o'om} \geq a_{om} + a_{o'm} - 1.$$

Operations on the same machine must not overlap: $\forall m \in M, \forall o, o' \in O_m$:

$$e_{om} - s_{o'm} \leq H(1 - p_{oo'm}) + H(2 - a_{om} - a_{o'm}), e_{om} - s_{o'm} \geq -H \cdot p_{oo'm} - H(2 - a_{om} - a_{o'm})$$

$$e_{o'm} - s_{om} \leq H \cdot p_{oo'm} + H(2 - a_{om} - a_{o'm}), e_{o'm} - s_{om} \geq -H(1 - p_{oo'm}) - H(2 - a_{om} - a_{o'm})$$

3.2.4 The objective function

The objective function penalizes the accumulated tardiness of the final operations of all jobs. These are the mixing vessel operations MV_j which finish later than the filling

operations because of the cleaning of the vessels: $\min \Omega = \sum_{j \in J} \max\{due_j - e_{MV_j}, 0\}$.

4. Solution Procedure

The commercial package GAMS/Cplex was used. Parameter studies of various Cplex parameters yielded that $dprind = 1$ clearly increased the solution performance. This setting was used for the procedure described below, the other parameters were set to their default values. The heuristics used here are similar to those in (Cerde et al., 1997).

- H1 - Non-overtaking of non-overlapping jobs:

if $due_j < rel_{j'}$, then $\forall m \in M, \forall o \in O_{jm}, o' \in O_{j'm} : e_{om} \leq s_{o'm}$

- H2 - Non-overtaking of jobs with equal recipes:

if $rec_j = rec_{j'} \wedge due_j < due_{j'}$, then $\forall m \in M, \forall o \in O_{jm}, o' \in O_{j'm} : e_{om} \leq s_{o'm}$

- H3 - Earliest due date: if $due_j < due_{j'}$, then $\forall m \in M, \forall o \in O_{jm}, o' \in O_{j'm} : e_{om} \leq s_{o'm}$.

Note that H1-H3 preserve enough degrees of freedom to keep the problem non-trivial. H1 and H2 do not exclude possibly optimal schedules. The 2-step solution procedure is:

1. Apply heuristics H3 to the problem by fixing the corresponding p-variables. Solve the problem and save the integer solution which represents a valid schedule.
2. Relax the variables fixed in step 1. Apply H1 and H2 by fixing the corresponding p variables. Solve the problem using the solution from step 1 as initial solution.

The solution obtained in step 1 is always a valid solution in step 2 and therefore can be used as an initial solution. This is because p-variables fixed to 1 in H1 and H2 are also fixed in H3. Being the most restrictive heuristics H3 fixes more p variables to 1 than H1 and H2 together. Both steps were implemented within the same GAMS model.

5. Optimization Results

The model and the solution procedure were implemented in the GAMS 21.3 language and solved with Cplex 9.0 on a 2.4 GHz Athlon machine with 1 GB of memory. Both steps of the solution procedure were limited to 20 min. computation time and a optimality gap of 5%. In order to investigate the scalability, several problem instances ranging from 10 to 29 jobs were solved. The job table was sorted according to due dates and for each instance, the first jobs from the table were taken. The results are shown in Table 1. The first column gives the problem size and the next 5 columns show the number of variables and equations of the problem as already reduced by Cplex. The final two columns show the solution times of both steps in seconds. All problem instances could be solved to optimality (accumulated tardiness and lower bound equal to 0) with reasonable computational effort (less than 700 seconds). While the number of real variables does not change between steps 1 and 2, the number of discrete variables increases considerably when the heuristics are switched from H3 to H1+H2. The first step of the solution procedure was always sufficient to compute optimal solutions. No

better solutions could be derived within the second step. The time spent in step 2 was needed to determine that the optimal solution had been found in step 1. Hence, the EDD heuristics (H3) led to optimal solutions in all test cases.

Table 1. Results of both steps 1 and 2 in the scalability experiments.

#Jobs	#Eqs.1+2	#Vars. 1	#Vars. 2	#disc. Vars 1	#disc. Vars 2	Time 1	Time 2
10	9807	1070	1376	762	1068	5.33	0.33
14	19187	1901	2585	1457	2141	20.12	0.77
16	25005	2406	3320	1898	2812	37.87	1.21
18	31553	2942	4016	2378	3452	57.07	1.64
20	39007	3559	4886	2923	4250	126.99	2.00
22	47273	4234	5800	3526	5092	243.71	2.61
29	81903	6971	9194	6043	8266	693.78	5.32

6. Summary, Conclusions and Remarks

This work demonstrates a successful application of a MILP formulation for short and mid-term scheduling problems to a real-life example from the chemical industry. An example problem with more than 80,000 equations and 9,000 variables could be solved to optimality in less than 700 seconds. Although less powerful (in terms of modelling strength) than the widely used STN/RTN approach, our model is sufficiently expressive to model sequencing constraints, resource assignment, various timing constraints and complex recipes with parallel operations. Sequence-dependant changeover procedures and more complex objective functions, involving storage costs and makespan, can also be accommodated in the proposed formulation. Unlike STN/RTN, the batch sizes must be planned in a separate pre-processing step. The solution efficiency was improved by introducing a two-step solution procedure involving heuristics. The authors gratefully acknowledge financial support from the EU AMETIST project (IST-2001-35304).

References

- Cerda, J., G. Henning and I. Grossmann, 1997, A Mixed-Integer Programming Model for Short-Term Scheduling of Single-Stage Multiproduct Batch Plants with Parallel Lines, *Ind. and Eng. Chem. Res.* 36 (1997), 1695-1707.
- Floudas, C. and X. Lin, 2004, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review, *Comp. and Chem. Eng.* 28 (2004), 2109-2129.
- Ierapetritou, M. and C. Floudas, 1998, Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes, *Ind. Eng. Chem. Res.* 37 (1998), 4341-4359.
- Kondili, E., C. Pantelides and R. Sargent, 1993, A general algorithm for short-term scheduling of batch operations – i. milp formulation, *Comp. and Chem. Eng.* 17 (1993), 211-227.
- Manne, A. S., 1960, On the job-shop scheduling problem, *Op. Res.* 8, Issue 2 (1960), 219-223.
- Pinto, J. M. and I. E. Grossmann, 1994, Optimal cyclic scheduling of multistage continuous multiproduct plants, *Comp. and Chem. Eng.* 18 (1994), 797-816.
- Schilling, G. and C. Pantelides, 1996, A simple continuous-time process scheduling formulation and a novel solution algorithm, *Comp. and Chem. Eng.* 20 (1996), 1221-1226.
- Shah, N., C. C. Pantelides and R. Sargent, 1993, A general algorithm for short-term scheduling of batch operations – ii. computational issues, *Comp. and Chem. Eng.* 17 (1993), no. 2, 229-244.