

A MINLP/RCPSP decomposition approach for the short-term planning of batch production

Norbert Trautmann^a and Christoph Schwindt^b

^aInstitute for Economic Theory and Operations Research
University of Karlsruhe, 76128 Karlsruhe, Germany

^bInstitute for Business Administration and Economics
Technical University of Clausthal, 38678 Clausthal-Zellerfeld, Germany

Abstract

We present a new solution approach for short-term planning of batch production, which decomposes the problem into batching and batch scheduling. Batching converts the primary requirements for products into individual batches. We formulate the batching problem as a mixed-integer nonlinear program, which can be solved by standard software. Batch scheduling allocates the batches to scarce resources such as processing units and intermediate storage facilities. The batch scheduling problem is modelled as a resource-constrained project scheduling problem, which is solved by a novel priority-rule based method.

Keywords: Batch production; Scheduling; Decomposition; Mixed-integer nonlinear programming; Resource-constrained project scheduling

1. Introduction

This paper deals with short-term planning of batch production in the process industries. In batch production mode, the total requirements of intermediate and final products are partitioned into batches. To produce a batch, at first the inputs are loaded into a processing unit. Then a transformation process is performed, and finally the batch is unloaded from the processing unit. We consider the case of multi-purpose processing units, which can operate different processes. The duration of a process depends on the processing unit used. The minimum and maximum filling levels of a processing unit give rise to lower and upper bounds on the respective batch size. Between consecutive executions of different processes in a processing unit, a changeover with sequence-dependent duration is necessary. Moreover, to avoid ongoing reactions of residuals, a processing unit needs to be cleaned before an idle time. In general, storage facilities of limited capacity are available for stocking raw materials, intermediates, and final products. Some products are perishable and must be consumed immediately after production. The product structure may be linear, divergent, convergent, or general. In the latter case, the product structure may also contain cycles. The input or output proportions are either fixed or can be chosen within prescribed bounds. For a practical example of a batch production, we refer to the case study presented in Kallrath (2002).

A plant is operated in batch production mode when a large number of different products are processed on multi-purpose equipment. In this case, the plant is configured according to (a subset of) the required final products. Before processing the next set of final products, the plant has to be reconfigured, which requires the completion of all operations. In order to ensure high resource utilization and short customer lead times, the objective of makespan minimization is particularly important. That is why we consider the short-term planning problem which for given primary requirements consists in computing a feasible schedule with minimum makespan.

Various solution methods for short-term planning of batch production are known from literature. Most of them follow a monolithic approach, which tackles the problem as a whole starting from a mixed-integer linear programming formulation of the problem. In those models, the period length is either fixed (time-indexed formulations, cf. e.g., Kondili et al., 1993) or variable (continuous-time formulations, see e.g., Ierapetritou and Floudas, 1998, or Castro et al., 2001). A disadvantage of the monolithic approaches is that the CPU time requirements for solving real-world problems tend to be prohibitively high (cf. Maravelias and Grossmann, 2004). To overcome this difficulty, heuristics reducing the number of variables have been developed (cf. e.g., Blömer and Günther, 1998).

A promising alternative approach is based on a decomposition of the problem into interdependent sub-problems, as it has been proposed e.g. by Brucker and Hurink (2000), Neumann et al. (2002), and Maravelias and Grossmann (2004). The solution approach developed in what follows decomposes the short-term production planning problem into a batching and a batch-scheduling problem. Batching provides a set of batches for the intermediate and final products needed to satisfy the primary requirements. Batch scheduling allocates the processing units, intermediates, and storage facilities over time to the processing of all batches. In this paper, we use a new formulation of the batching problem as a mixed-integer nonlinear program that, in contrast to the model discussed in Neumann et al. (2002), allows for taking into account alternative processing units of different size. Moreover, we present a novel priority-rule based method for batch scheduling which is able to cope with large problem instances.

The remainder of this paper is organized as follows. In Section 2 we formulate the batching problem as a mixed-integer nonlinear program. In Section 3 we show how to model the batch-scheduling problem as a resource-constrained project scheduling problem, and we sketch an appropriate priority-rule based solution method. Results of an experimental performance analysis of the new approach are discussed in Section 4. Section 5 is devoted to concluding remarks.

2. The batching problem

In what follows, the combination of a transformation process and a processing unit is referred to as a task. For example, if there are three alternative processing units for the execution of a transformation process, we define three tasks for this process. Let T be the set of all tasks and let β_τ and ε_τ be the batch size and number of batches for task $\tau \in T$. By Π_τ^- and Π_τ^+ we denote the set of input and output products, respectively, of task τ . $\Pi_\tau := \Pi_\tau^- \cup \Pi_\tau^+$ is the set of all input and output products of task τ , and

$\Pi := \bigcup_{\tau \in T} \Pi_{\tau}$ is the set of all products considered. In addition to β_{τ} and ε_{τ} , the proportions $\alpha_{\tau\pi} < 0$ of all input products $\pi \in \Pi_{\tau}^{-}$ and the proportions $\alpha_{\tau\pi} > 0$ of all output products $\pi \in \Pi_{\tau}^{+}$ have to be determined for all tasks $\tau \in T$ such that

$$\sum_{\pi \in \Pi_{\tau}^{+}} \alpha_{\tau\pi} = - \sum_{\pi \in \Pi_{\tau}^{-}} \alpha_{\tau\pi} = 1 \quad (\tau \in T) \quad (1)$$

Batch sizes β_{τ} and proportions $\alpha_{\tau\pi}$ have to be chosen within prescribed intervals $[\underline{\beta}_{\tau}, \bar{\beta}_{\tau}]$ and $[\underline{\alpha}_{\tau\pi}, \bar{\alpha}_{\tau\pi}]$, i.e.,

$$\underline{\alpha}_{\tau\pi} \leq \alpha_{\tau\pi} \leq \bar{\alpha}_{\tau\pi} \quad (\tau \in T, \pi \in \Pi_{\tau}) \quad (2)$$

$$\underline{\beta}_{\tau} \leq \beta_{\tau} \leq \bar{\beta}_{\tau} \quad (\tau \in T) \quad (3)$$

Let T_{π}^{-} and T_{π}^{+} be the sets of all tasks consuming and producing, respectively, product $\pi \in \Pi$ and let $\Pi^p \subseteq \Pi$ be the set of perishable products. Then equations

$$\alpha_{\tau\pi}\beta_{\tau} = -\alpha_{\tau'\pi}\beta_{\tau'} \quad (\pi \in \Pi^p, (\tau, \tau') \in T_{\pi}^{+} \times T_{\pi}^{-}) \quad (4)$$

ensure that the amount of a perishable product π produced by one batch of some task $\tau \in T_{\pi}^{+}$ can immediately be consumed by any task $\tau' \in T_{\pi}^{-}$ consuming π .

By ρ_{π} we denote the primary requirements less the initial stock of π . For recycled products $\pi \in \Pi$, ρ_{π} is augmented by an unavoidable residual stock after the completion of all batches. The final inventory of product π then equals $\sum_{\tau \in T} \alpha_{\tau\pi}\beta_{\tau}\varepsilon_{\tau}$. This amount

must be sufficiently large to match the requirements ρ_{π} for π . On the other hand, the final stock $\sum_{\tau \in T} \alpha_{\tau\pi}\beta_{\tau}\varepsilon_{\tau} - \rho_{\pi}$ of product π must not exceed the given storage capacity

σ_{π} for π . Both necessary conditions can be formulated as

$$\rho_{\pi} \leq \sum_{\tau \in T} \alpha_{\tau\pi}\beta_{\tau}\varepsilon_{\tau} \leq \rho_{\pi} + \sigma_{\pi} \quad (\pi \in \Pi) \quad (5)$$

In addition, the number of batches ε_{τ} must be integral, i.e.,

$$\varepsilon_{\tau} \in \mathbb{Z}_{\geq 0} \quad (\tau \in T) \quad (6)$$

To formulate the objective function, we divide the processing units into a set Γ of groups γ in such a way that first, each processing unit belongs to exactly one group and second, each transformation process can only be executed on processing units of one and the same group. Let Y_{γ} be the set of processing units in group $\gamma \in \Gamma$. By T_{υ} we denote the set of tasks that can be executed on processing unit υ . We refer to the processing unit $\upsilon \in Y_{\gamma}$ with maximum potential workload $\sum_{\tau \in T_{\upsilon}} p_{\tau}\varepsilon_{\tau}$ as the bottleneck of group γ , where p_{τ} stands for the processing time of task τ . The objective function to

be minimized is chosen to be the sum $\sum_{\gamma \in \Gamma} \max_{\tau \in T_\gamma} \sum_{\tau \in T_\gamma} p_\tau \varepsilon_\tau$ of all bottleneck workloads. In

this way we ensure that the total bottleneck workload is kept as small as possible while the actual workload to be processed is equally balanced among the alternative processing units of a group.

3. The batch-scheduling problem

3.1 Modelling as a resource-constrained project scheduling problem

Suppose that n batches numbered from 1 to n have to be scheduled. We model the processing of the batches as a project (cf. Brucker et al., 1999) that consists of a set $V = \{0, 1, \dots, n, n+1\}$ of activities, which require resources and time for their execution and which are linked by prescribed time lags between their starts. The processing of a batch is identified with exactly one activity $i \in \{1, \dots, n\}$ of the project. Dummy activity 0 represents the production start and dummy activity $n+1$ corresponds to the production end. Let $S_i \geq 0$ be the start time sought of activity i . Then S_{n+1} coincides with the production makespan. Vector $S = (S_i)_{i \in V}$ with $S_0 = 0$ is called a schedule.

Each processing unit can be viewed as a unit-capacity renewable resource. Let R^p be the set of all renewable resources and let $k_i \in R^p$ be the resource processing real activity i . By p_i and c_i we denote the processing and cleaning times of activity i , where we suppose that $p_i = c_i = 0$ for $i = 0, n+1$.

The need for cleaning a processing unit generally depends on the sequence in which the activities are executed on this unit. Let $P_k \in V \times V$ be the set of activity pairs (i, j) for which passing from i to j requires a cleaning of processing unit k . Given a schedule S , let $O_k(S)$ designate the set of all pairs (i, j) such that $i \neq j$, $S_i \leq S_j$, and $k_i = k_j$. $O_k(S)$ can be partitioned into the set $C_k(S)$ containing all pairs (i, j) for which k has to be cleaned between the completion of i and the start of j (because $(i, j) \in P_k$ or $S_j > S_i + p_i$) and the set $\bar{C}_k$ of pairs for which j must be started immediately after the completion of activity i . A schedule S is called process-feasible if

$$\left. \begin{array}{l} S_j \geq S_i + p_i + c_i, \text{ if } (i, j) \in C_k(S) \\ S_j = S_i + p_i, \text{ if } (i, j) \in \bar{C}_k(S) \end{array} \right\} \quad (k \in R^p) \quad (7)$$

Now we turn to the intermediates and storage facilities, which are both represented by so-called cumulative resources (cf. Neumann and Schwindt, 2002). For each non-perishable product, there is one cumulative resource keeping its inventory. Let R^y be the set of all cumulative resources. For each $k \in R^y$, a minimum inventory $\underline{R}_k$ (safety stock) and a maximum inventory $\bar{R}_k$ (storage capacity) are given. Each activity $i \in V$ has a demand r_{ik} for resource $k \in R^y$. If $r_{ik} > 0$, the inventory of resource k is replenished by r_{ik} units at time $S_i + p_i$. If $r_{ik} < 0$, the inventory is depleted by $-r_{ik}$ units at

time S_i . r_{0k} represents the initial stock level of resource k . Let $V_k^+ := \{i \in V \mid r_{ik} > 0\}$ and $V_k^- := \{i \in V \mid r_{ik} < 0\}$ be the sets of activities replenishing and depleting, respectively, the inventory of $k \in R^Y$. Schedule S is said to be storage-feasible if

$$\underline{R}_k \leq \sum_{i \in V_k^+ : S_i + p_i \leq t} r_{ik} + \sum_{i \in V_k^- : S_i \leq t} r_{ik} \leq \bar{R}_k \quad (k \in R^Y, t \geq 0) \quad (8)$$

To avoid waiting times between activities respectively producing and consuming perishable intermediates, temporal constraints of the type

$$S_j \geq S_i + \delta_{ij} \quad ((i, j) \in E) \quad (9)$$

with $E \subseteq V \times V$ have to be taken into account as well. δ_{ij} is a minimum time lag between the start of activities i and j . If $\delta_{ij} < 0$, then $-\delta_{ij}$ can be interpreted as a maximum time lag between the start of activities j and i . In case of $\delta_{ij} = p_i$, the corresponding temporal constraint is referred to as a precedence constraint. A schedule S satisfying $S_j \geq S_i + \delta_{ij}$ for all $(i, j) \in E$ is called time-feasible.

A schedule which is time-, process-, and storage-feasible is called feasible. The batch scheduling problem consists in finding a feasible schedule S with minimum makespan S_{n+1} .

3.2 Solution procedure

The priority-rule based scheduling method consists of two phases. During the first phase, we relax the storage-capacity constraints. Using a serial schedule-generation scheme, the activities are iteratively scheduled on the processing units in such a way that the inventory does not fall below the safety stock at any point in time. Deadlocks are avoided by means of a specific unscheduling technique. Based on the resulting schedule, precedence constraints between replenishing and depleting operations are introduced according to a FIFO strategy. Those precedence constraints ensure that the material-availability constraints are always satisfied. In the second phase, which again applies the serial schedule-generation scheme, the activities are scheduled subject to the storage-capacity and the precedence constraints introduced. Details of this procedure can be found in Schwindt and Trautmann (2004).

4. Computational results

We have compared our decomposition approach to the time-grid heuristic by Blömer and Günther (1998) and to the decomposition method by Maravelias and Grossmann (2004). As a test bed we have used the 22 instances introduced by Blömer and Günther, which have been constructed by varying the primary requirements for final products in the case study presented by Kallrath (2002). In addition, we have solved Example 2 discussed in Maravelias and Grossmann (2004).

For solving the batching problem, we have used the Solver package by Frontline Systems. For batch scheduling, we have implemented a randomized multi-pass version of the priority-rule based solution procedure in ANSI C. All computations have been performed on an 800 MHz Pentium III personal computer.

It turns out that for each of the 23 instances in the test set, the batching problem could be solved within less than 8 seconds. In each case, the optimality of the solution found could be verified by using an alternative MILP-formulation of the problem. The sizes of the resulting batch scheduling instances range from 24 to 100 operations. For all instances, within less than four minutes the priority-rule based method has either provided an optimal solution or the best solution known thus far could be improved. The results obtained for the large instances with more than 50 operations indicate that our decomposition approach scales quite well. The mean relative improvement achieved for those instances with respect to the time grid heuristic amounts to more than 35%. Moreover, the CPU time requirements have been decreased significantly compared to the time-grid heuristic and have been comparable to those reported by Maravelias and Grossmann.

5. Conclusions

In this paper we have presented an efficient heuristic method for the short-term planning of batch production, which is based on a decomposition of the problem into a batching and a batch scheduling problem. Whereas the batching problem is formulated as a MINLP of moderate size, the batch scheduling problem is solved by a novel two-phase priority-rule based method for resource-constrained project scheduling. The decomposition heuristic is able to approximately solve problem instances of practical size in the space of a few minutes. An important area of future research will be the development of efficient online-scheduling procedures that are based on the priority-rule based method. Such an online-scheduling algorithm could be used in the Available-to-Promise module of Advanced Planning Systems for Supply Chain Management.

References

- Blömer, F. and H.O. Günther, 1998, Scheduling of a multi-product batch process in the chemical industry, *Comp. Ind.* 36, 245.
- Brucker, P. and J. Hurink, 2000, Solving a chemical batch scheduling problem by local search, *Annals Oper. Res.* 96, 17.
- Brucker, P., A. Drexler, R. Möhring, K. Neumann, and E. Pesch, 1999, Resource-constrained project scheduling: notation, classification, models, and methods. *Eur. J. Oper. Res.* 112, 3.
- Castro, P., A.P. Barbosa-Póvoa, and H. Matos, 2001, An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants, *Ind. Eng. Chem. Res.* 40, 2059.
- Ierapetritou, M.G. and C.A. Floudas, 1998, Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes, *Ind. Eng. Chem. Res.* 37, 4341.
- Kallrath, J., 2002, Planning and scheduling in the process industry, *OR Spectrum* 24, 219.
- Kondili, E., C.C. Pantelides, and R.W.H. Sargent, 1993, A general algorithm for short-term scheduling of batch operations — I. MILP formulation, *Comput. Chem. Eng.* 17, 211.
- Maravelias, C.T. and I.E. Grossmann, 2004, A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants, *Comput. Chem. Eng.* 28, 1921.
- Neumann, K. and C. Schwindt, 2002, Project scheduling with inventory constraints, *Math. Meth. Oper. Res.* 56, 513.
- Neumann, K., C. Schwindt, and N. Trautmann, 2002, Advanced production scheduling for batch plants in process industries, *OR Spectrum* 24, 251.
- Schwindt, C. and N. Trautmann, 2004, A priority-rule based method for batch production scheduling in the process industries. In: Ahr, D., R. Fahrion, M. Oswald, and G. Reinelt, Eds., *Operations Research Proceedings 2003*. Springer, Berlin, 111.