Proceedings of the 14th International Symposium on Process Systems Engineering – PSE 2021+
June 19–23, 2022, Kyoto, Japan

An Improved Optimization Model for Scheduling of an Industrial Formulation Plant based on Integer Linear Programming

Vassilios Yfantis[a*], Alexander Babskiy[a], Christian Klanke[b], Martin Ruskowksi[a], Sebastian Engell[b]

[a]*Chair of Machine Tools and Control Systems, Technische Universität Kaiserslautern, Gottlieb-Daimler-Str. 42, 67663 Kaiserslautern, Germany*

[b]*Process Dynamics and Operations Group, TU Dortmund University, Emil-Figge-Str. 70, 44227 Dortmund, Germany*

vassilios.yfantis@mv.uni-kl.de

Abstract

This contribution deals with the development of an integer linear programming (ILP) model and a solution strategy for a two-stage industrial formulation plant with parallel production units for crop protection chemicals. Optimal scheduling of this plant is difficult, due to the number of units and operations that must be scheduled while at the same time a high degree of coupling between the operations is present. The problem is further complicated by the presence of optional intermediate storage that leads to alternative branches in the processing sequence of the products. The presented approach is compared to previous ones, namely a mixed-integer linear programming- and a constraint programming-based one. The ILP-based approach exhibits vastly superior computational performance, while still achieving the same solution quality.

Keywords: Batch Process Scheduling, Integer Linear Programming, Decomposition

1. Introduction

The increasing competition on the global market in addition to varying customer demands necessitates an increase in the efficiency and flexibility of production processes. Batch processes offer this kind of flexibility in the case of demand-driven production. A key component to the efficiency of such batch processes is optimal scheduling, i.e., the allocation of limited resources to manufacture several products over a given time horizon. Schedules should be generated in a fast and reliable manner to adapt to varying customer demands. Furthermore, schedules should try to optimize some criterion, e.g., minimizing production time or maximizing profit. These requirements for scheduling can be addressed by optimization methods, like integer and mixed-integer programming. Optimization models can include various constraints that describe the production process while simultaneously optimizing a scheduling objective. The main bottleneck of most optimization models in production scheduling is the computation time. This issue can be handled by applying decomposition techniques, where the scheduling problem is solved in an iterative manner (Elkamel et al., 1997). A straightforward decomposition approach is the iterative scheduling of batches or orders. The realization of the decomposition then mainly depends on the

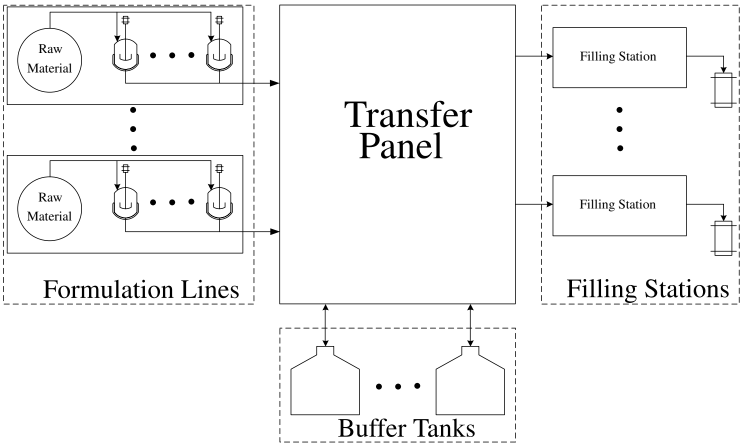

Figure 1: Schematic representation of the industrial formlation plant (Yfantis et al., 2019).

model structure, e.g., whether the model represents time through a time grid (Yfantis et al., 2019) or through precedence relations (Elekidis et al., 2019). In this contribution an efficient integer linear programming model for an industrial formulation plant is presented. A decomposition approach is employed, where orders are scheduled in an iterative fashion, while considering decisions from previous iterations. The solution approach is evaluated on an industrial-scale case study.

2. Indutrial Formulation Plant

The industrial formulation plant is schematically depicted in Figure 1. It can be divided into three parts, the formulation lines, the filling stations, and the buffer tanks. The plant operates in a sequential manner. Intermediate products are produced in the formulation lines and then filled into their final containers by the filling stations. The buffer tanks serve to decouple the two production stages. All sections of the plant are connected by a transfer panel. Each formulation line consists of a raw material pre-processing line, in which the preparation of active ingredients and solvents takes place, and several identical standardization tanks. After the pre-processing, a batch mixing operation takes place in one of the standardization tanks. The standardization tanks are always utilized to their full capacity, i.e., overproduction can occur. As a single pre-processing line feeds multiple standardization tanks, only one batch can start processing in each formulation line at each time point. Furthermore, each order can only be processed on a subset of available formulation lines. After a minimum standardization time, the intermediate product can be filled by a filling station. However, intermediate storage in the standardization tank or in an available buffer tank is also possible. The filling stations operate in a continuous manner, i.e., without an internal storage. A connected standardization or buffer tank is continuously drained by the filling station with an order and station dependent flowrate. Each filling station can only process a subset of available orders. After an operation finishes in any piece of equipment, a sequence dependent changeover time must elapse before the start of the next operation. The filling stations constitute a bottleneck of the process, as they cannot operate during the night shift, unlike the formulation lines, which operate continuously during the entire time horizon. The scheduling task consists of allocating the batches of the different orders to the standardization tanks and the subsequent filling operations to the filling stations while minimizing the total production time of the schedule. The buffer tanks can be used to decouple the two production stages, while accounting for constraints on the maximum capacity of the tanks. The case study has been investigated by Yfantis et al. (2019) and Klanke et al. (2021b). In the former work, mixed-integer linear programming (MILP) was employed together with a decomposition strategy, and a problem instance identical to the one examined in this paper was solved for a scheduling horizon of one week. In Klanke et al. (2021b) the same problem instance was solved by combining constraint programming (CP) and a moving-horizon strategy, outperforming the previous MILP formulation. Furthermore, different case studies for the same formulation plant were solved in Klanke et al. (2021a) using a heuristics-assisted genetic algorithm.

3. Solution Approach

3.1. Integer Linear Programming Model

In this section, the proposed integer linear programming model is presented. Since the model is very complex, this sections only focuses on some key constraints and variables, as well as on the objective function. The goal is to schedule the set of orders $\mathcal{I}$ on the available machines J. The machines are divided into the standardization tanks of the formulation lines $\mathcal{J}^{\text{FL}}$ and the filling stations $\mathcal{J}^{\text{FS}}$. The machines that can process order i are denoted by $\mathcal{J}_i$. The available buffer tanks are modeled by the set $\mathcal{B}$. The time horizon is discretized into equidistant time points $\mathcal{T}$. Some of the key constraints are shown in Eq. (1) - (6). The binary variable R_{ijt} indicates that a batch of order i is released from standardization tank j at time t. Eq. (1) guarantees the satisfaction of demand D_i, where cap_j is the batch size in tank j. The binary variable is set to one once the tank has been emptied. This is modeled by Eq. (2), where $E^{\text{fill}}_{ijj't}$ is a binary variable indicating the end of filling of a batch of order i from standardization tank j by filling station j' at time t and $Ref_{ijbj't}$ is a binary variable indicating a refilling of this batch into buffer tank b, in order to later be filled by filling station j'. A batch can be stored inside a standardization tank prior to its release. Intermediate storage of a batch of order i in standardization tank j at time t is indicated by the binary variable L_{ijt}. This variable is updated by Eq. (3), where E_{ijt} is a binary variable representing the end of a standardization operation, $S^{\text{fill}}_{ijj't}$ models the start of a filling operation from standardization tank j by filling station j'. When a standardization tank j is processing a batch of order i at time t the binary variable X_{ijt} is active. It is updated through the starting (S_{ijt}) and ending (E_{ijt}) binary variables in Eq. (4). Processing in the filling station is modeled by similar constraints. An important aspect of the scheduling problem is the modeling of the buffer balances. Instead of modeling stored quantities in the buffer tanks, Eq. (5) models the time intervals necessary to empty buffer tank b, containing order i by filling station j, if filling starts at time t through the integer variable I_{ibjt}. This variable is updated at every time step, using the parameter $p_{ijj'}$, which is equal to the number of time points needed to fill a batch of order i from standardization tank j' by filling station j and the binary variable Y_{ibjt}, indicating that an order i is filled from buffer b by filling station j at time t. Eq.

(6) ensures that the buffer level does not exceed its maximum capacity by considering an upper bound on the required filling time. Further constraints include the changeovers in the different machines, modeled in a similar fashion to Eq. (4), also using binary variables for their start, end, and processing. The processing times are modeled by linking the binary variables for the start and end of an operation through their time indices. The objective of the optimization problem is modeled by Eq. (7). In the first term the starting and end times of the filling operations are minimized. The remaining terms serve to minimize idle times, which occur in a makespan minimization due to multiple symmetric solutions. The second term penalizes the use of the buffer tanks. The scaling parameter w_{ij} is equal to the mean filling time of batches of order i by filling station j. The third term discourages intermediate storage in the standardization tanks if it is unnecessary.

$$\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}_i^{\text{FL}}} R_{ijt} \cdot cap_j \geq D_i, \ \forall i \in \mathcal{I} \tag{1}$$

$$R_{ij,t+1} = \sum_{j' \in \mathcal{J}_i^{\text{FS}}} E^{\text{fill}}_{ijj't} + \sum_{b \in \mathcal{B}} \sum_{j' \in \mathcal{J}_i^{\text{FS}}} Ref_{ijbj',t+1}, \ \forall i \in \mathcal{I}, j \in \mathcal{J}_i^{\text{FL}}, t \in \mathcal{T} \backslash \{|\mathcal{T}|\} \tag{2}$$

$$L_{ij,t+1} = L_{ijt} + E_{ijt} - \sum_{j' \in \mathcal{J}_i^{\text{FS}}} S^{\text{fill}}_{ijj',t+1} - \sum_{b \in \mathcal{B}} \sum_{j' \in \mathcal{J}_i^{\text{FS}}} Ref_{ijbj',t+1}, \\ \forall i \in \mathcal{I}, j \in \mathcal{J}_i^{\text{FL}}, t \in \mathcal{T} \backslash \{|\mathcal{T}|\} \tag{3}$$

$$X_{ij,t+1} = X_{ij,t} + S_{ij,t+1} - E_{ijt}, \ \forall i \in \mathcal{I}, j \in \mathcal{J}_i^{\text{FL}}, t \in \mathcal{T} \backslash \{|\mathcal{T}|\} \tag{4}$$

$$I_{ibj,t+1} = I_{ibjt} + \sum_{j' \in \mathcal{J}_i^{\text{FL}}} Ref_{ij'bj,t+1} \cdot p_{ijj'} - Y_{ibjt}, \ \forall i \in \mathcal{I}, b \in \mathcal{B}, j \in \mathcal{J}_i^{\text{FS}} \tag{5}$$

$$I_{ibjt} \leq cap_{ibj}, \ \forall i \in \mathcal{I}, b \in \mathcal{B}, j \in \mathcal{J}_i^{\text{FS}}, t \in \mathcal{T} \tag{6}$$

$$\min \ \frac{1}{2} \cdot \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i^{\text{FL}}} \sum_{j' \in \mathcal{J}_i^{\text{FS}}} \sum_{t \in \mathcal{T}} \left(E^{\text{fill}}_{ijj't} + S^{\text{fill}}_{ijj't} \right) + \sum_{i \in \mathcal{I}} \sum_{b \in \mathcal{B}} \sum_{j \in \mathcal{J}_i^{\text{FS}}} \sum_{t \in \mathcal{T}} \frac{1}{w_{ij}} \cdot Y_{ibjt} \cdot t \\ + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i^{\text{FL}}} \sum_{t \in \mathcal{T}} L_{ijt} \tag{7}$$

3.2. Decomposition

Due to its size and complexity the model cannot be solved in a monolithic fashion. To generate feasible schedules in a time efficient manner an order decomposition approach, similar to Yfantis et al. (2019), is employed. The orders are scheduled iteratively while preventing overlaps through constraints. These infeasible allocations can easily be identified since each machine possesses an active binary variable at each time point where an operation is being performed, instead of just using a single binary variable for the start of an operation. The night shifts of the filling stations are modeled in a similar way. In contrast to the approaches in Yfantis et al. (2019) and Klanke et al. (2021b) no batch decomposition is needed, as the model can schedule orders that consist of a large number of

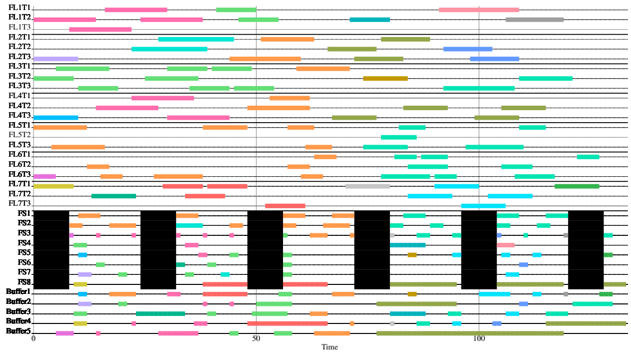

Figure 2: Gantt chart of the schedule generated with the proposed ILP-based approach.

batches efficiently. Furthermore, no two-step optimization approach is necessary, as the operations are already shifted to the left through the chosen objective function. Lastly, the time horizon is shifted to the end of the next day from the current makespan. If the subproblem is infeasible the time horizon is shifted by an additional day until a solution is found.

4. Results

The presented ILP-based solution approach was evaluated on the case study presented in Yfantis et al. (2019) and Klanke et al. (2021b). The setup consists of 7 formulation lines, each containing 3 standardization tanks, 8 filling stations and 5 buffer tanks. In total, 20 orders of different quantities, resulting in 78 batches are scheduled. A time horizon of one week, divided into 1-hour intervals, is considered. The solution approach was implemented in the programming language Julia (Bezanson et al., 2017). The ILP was solved using Gurobi on a Desktop PC (AMD Ryzen 5 3600 6-Core Processor @3.6 GHz). The subproblems were all solved to a 0 % optimality gap. The generated Gantt chart is depicted in Figure 2. It represents the batches on each standardization tank of the formulation lines, separated by the black solid lines, the filling stations, and the buffer tanks. Furthermore, the night shifts of the filling stations are illustrated as black regions. A makespan of 133 h is obtained, which is equal to the results obtained by the previous solution approaches. However, the benefit of the proposed ILP-based approach can be seen in the required computation time (cf. Table 1). The superior performance of the ILP model is further underlined by the fact, that no batch-based decomposition is needed. Instead, only an order decomposition is performed, so that a single subproblem can require scheduling a large number of batches, which would render it intractable for the previous approaches. The computation time is further enhanced by the lack of a two-step optimization approach, due to the chosen objective function, which results in fewer idle times than a makespan minimization. The superior performance can be attributed to the multiple active binary variables for a given schedule. In the MILP-based approach of Yfantis et al. (2019) binary variables only indicate the start of an operation, resulting in far less active binary variables. The tightly constrained active binary variables of the ILP aid the search procedure of the solver.

Table 1: Comparison between different solution approaches for scheduling of the industrial formulation plant.

Model	MILP (Yfantis et al., 2019)	CP (Klanke et al., 2021b)	ILP
Makespan	133 h	133 h	133 h
Computation Time	38 min	23 min	51 s

5. Conclusion and Outlook

This work presented a novel ILP-based formulation for the scheduling of an industrial formulation plant. In contrast to previous approaches, the model only employs integer variables, which greatly enhances its computational performance. Instead of minimizing the makespan, an objective function that discourages idle times is formulated, eliminating the need for a two-step optimization approach. The structure of the model enables a monolithic optimization without running into memory limitation issues. However, then the solution times are prohibitive for a real application. Nevertheless, in future work a monolithic optimization can be performed on specialized hardware to provide a reference for the decomposition approaches and other solution methods.

Acknowledgements

This work was partially funded by the European Regional Development Fund (ERDF) in the context of the project OptiProd.NRW (https://www.optiprod.nrw/en).

References

J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, 2017. Julia: A fresh approach to numerical computing. SIAM Review 59 (1), pp. 65–98.

A.P. Elekidis, F. Corominas, M.C. Georgiadis, 2019. Production Scheduling of Consumer Goods Industries. Industrial & Engineering Chemistry Research 58, pp. 23261-23275.

A. Elkamel, M. Zentner, J.F. Pekny, G.V. Reklaitis, 1997. A Decomposition Heuristic for Scheduling the General Batch Chemical Plant. Engineering Optimization 28 (4), pp. 299-330.

C. Klanke, D. Bleidorn, C. Koslowski, C. Sonntag, S. Engell, 2021a. Simulation-based scheduling of a large-scale industrial formulation plant using a heuristics-assisted genetic algorithm. GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1587–1595.

C. Klanke, D. Bleidorn, V. Yfantis, S. Engell, 2021b. Combining Constraint Programming and Temporal Decomposition Approaches - Scheduling of an Industrial Formulation Plant. Lecture Notes in Computer Science, Vol. 12735, pp. 133-148.

V. Yfantis, T. Siwczyk, M. Lampe, N. Kloye, M. Remelhe, S. Engell, 2019. Iterative Medium-Term Production Scheduling of an Industrial Formulation Plant. In: Computer Aided Chemical Engineering, Vol. 46, pp. 19–24.