

This item is the archived peer-reviewed author-version of:

Efficient multi-product multi-BOM batch scheduling for a petrochemical blending plant with a shared pipeline network

Reference:

Hill Alessandro, Cornelissens Trijntje, Sørensen Kenneth.- Efficient multi-product multi-BOM batch scheduling for a petrochemical blending plant with a shared pipeline network

Computers and chemical engineering - ISSN 0098-1354 - 84(2016), p. 493-506

Full text (Publisher's DOI): <http://dx.doi.org/doi:10.1016/J.COMPCHEMENG.2015.09.015>

To cite this reference: <http://hdl.handle.net/10067/1289810151162165141>

Efficient multi-product multi-BOM batch scheduling for a petrochemical blending plant with a shared pipeline network

Alessandro Hill* Trijntje Cornelissens
Kenneth Sörensen

Department of Engineering Management
University of Antwerp
Prinsstraat 13, 2000 Antwerp, Belgium

November 23, 2015

Abstract

We present an effective scheduling heuristic for realistic production planning in a petrochemical blending plant. The considered model takes into account orders spanning a multi-product portfolio with multiple bills of materials per product, that need to be scheduled on shared production facilities including a complex pipeline network. Capacity constraints, intermediate storage restrictions, due dates, and the dedication of resources to specific product families have to be respected. The primary objective of the heuristic is to minimize the total order tardiness. Secondary objectives include the minimization of pipeline cleaning operations, the minimization of lead times, and the balanced utilization of filling units.

The developed algorithm is based on a dynamic prioritization-based greedy search that schedules the orders sequentially. The proposed method can schedule short to mid-term operations and evaluate different plant configurations or production policies on a tactical level. We demonstrate its performance on various real-world inspired scenarios for different scheduling strategies.

Our heuristic was used during the construction phase of a new blending plant and was instrumental in the optimal design of the plant.

Key words: batch scheduling, chemical blending plant, heuristic.

*corresponding author

Email addresses: alessandro.hill@uantwerpen.be, trijntje.cornelissens@uantwerpen.be, kenneth.sorensen@uantwerpen.be

1 Introduction

In this paper we present a heuristic to solve a batch scheduling problem in a real(istic) multi-product petrochemical blending plant. In this competitive semi-process industry, expected customer service level is high and delivery lead-times are short. Non-performance is penalized with high fines. The number of different products produced, as well as the high cost of liquid bulk storage forces the plant to focus on a make-to-order (MTO) production strategy. This in turn renders the On-Time In-Full (OTIF) delivery of customer orders a challenging task. For these reasons, the efficient planning and operating of such a high-performing blending plant needs decision support systems based upon powerful planning and scheduling algorithms.

This research was motivated by a research project executed in collaboration with a petrochemical company. The focus of this project was not on the development of an operational scheduling tool to handle the day-to-day planning of the plant; for this purpose, the company used a commercial software package. Rather, the aim was to develop a planning heuristic that could be used as the scheduling component of tactical analysis tools. For this reason, the heuristic needed to be both very fast and very flexible, being able to plan, e.g., the expected order set of an entire year in a reasonable amount of time, while taking into account multiple objectives.

As a result, the generic heuristic we propose is appropriate for short-term scheduling, but its design allows it to be integrated in tools for strategic or tactical decision support on plant layout (number, capacity and location of production facilities), and plant configuration (dedication of production facilities to product families, availability of personnel, opening or closing pipeline connections, etc.). Our heuristic was used during the startup phase of a new petrochemical blending plant to simulate the efficiency and determine the expected capacity of the plant design. To this end, it was expected that the heuristic should be able to schedule a realistic set of customer orders for a whole year in a reasonable amount of computing time. The performance of our heuristic made it possible to test new scenarios for both plant layout and plant configuration, which was important to determine the best possible configuration of the new plant. A detailed discussion of this validation phase follows in Section 5.

The heuristic considers raw material availability, production capacities, intermediate storage limitations, customer order due dates, as well as the several characteristics proper to the production process. Additionally, it takes into account the complex pipeline network that regulates the material flows between all production facilities. In fact, the layout of the pipeline system is designed to limit multiple direct end-to-end connections, but the shared pipelines generate supplementary resource availability constraints and a higher risk for contamination between production batches.

1.1 Industrial application

Although the scheduling problem described in this paper applies to semi-process production plants of different size, capacity and industry (e.g., specialty chemicals, pharmaceutical, food sector), the target in this paper is an industrial petrochemical blending plant with a yearly capacity of 100 000 m³. The plant completes circa 80 000 customer orders annually, of which 80% are delivered in bulk and 20% in drums. It

produces about 500 different products, independent of further labeling or packaging.

45 Regarding production facilities, the target plant counts about 60 component tanks for base oils and additives, 20 blending tanks, 50 intermediate storage tanks, 12 loading arms, and 3 packaging lines. The plant's pipeline system uses 4 manifolds that control the access to 400 pipelines. To avoid contamination in the pipelines, a state-of-the-art cleaning system pushes a pressure-driven PIG (Pipeline Intervention Gadget) through
50 the pipes in order to clean or empty them when needed. The use of the PIG should be minimised.

Characteristically for this industry each blending product has its preferred formula, but on top of that it can be produced according to multiple Bills-of-Material (BOM), using different base oils and additives depending on their price and availability.
55 As a result, the total number of different BOMs used over a year amounts to 5000. Since most of the BOM-components can be stored in multiple component tanks, the accessibility of the component pipelines must be considered when choosing the BOM and the related component tanks for each production order.

As mentioned earlier the majority of products are produced to order (MTO). Even
60 though a limited number of fast-moving products are produced to stock (MTS), the heuristic does not treat these products differently, as this would require it to take into account the full complexity of MTS replenishment policies, which in turn would force it to consider continuous or periodic review, safety stock levels, target stock levels, optimal replenishment periods, and economic order quantities. In this paper,
65 we therefore assume that all products are MTO and that every production order corresponds to a customer order. The only distinction made between products is that, before loading or packaging, customer orders can always be stored one by one in a *single-order* tank dedicated to the corresponding product family, but that for a few fast moving products the customer orders are preferably stored in a dedicated *single-product*
70 tank, possibly simultaneously with other customer orders for that product.

Finally, production and logistic personnel work in a one-, two- or three-shift pattern depending on the seasonal demand. As a consequence the use of component and blending tanks, pipelines and finishing lines is limited to these working hours.

1.2 Production process

75 The stages of the production process needed to blend a product are illustrated in Figure 1. The white rectangles represent transfer operations through pipelines, during which the source and destination tanks are occupied too, but no other value-adding activity occurs. In Stage 1 (Component Pumping) all necessary BOM-components such as base oils and additives, are pumped sequentially out of their individual component tanks into a blending tank. Scarce or low-volume additives are injected via a
80 decanting unit that can be considered as a special type of component tank. During the pumping of a component, the component tank, connecting pipelines and the blending tank are occupied. The component tanks are replenished when needed.

Throughout Stage 2 (Blending) the product components are mixed in a blending
85 tank. These tanks have different sizes and are dedicated to one or more product families. A blending tank is occupied during component pumping (Phase-In Blending), heating and blending (Product Blending), and during the transfer of the product into a

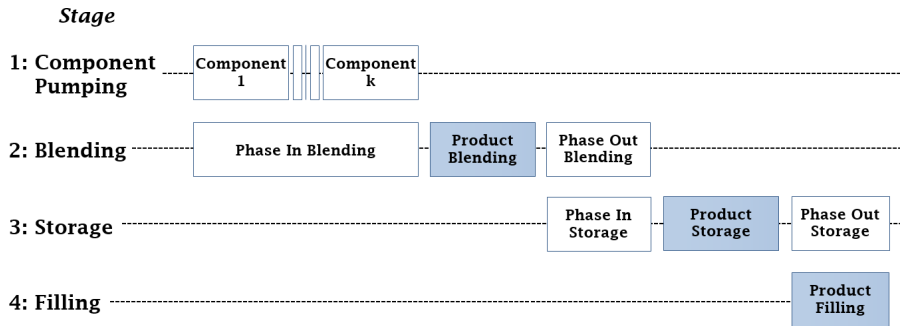


Figure 1: The stages of a typical blending production process and their operations.

storage tank (Phase-Out Blending). The duration of the heating and blending process is independent of the product volume but can depend on the product type and blending tank. The duration of Phase-In and Phase-Out Blending depends on the pumping capacity and product volume transferred.

During Stage 3 (Storage) the product is stored into a storage tank. Like blending tanks, storage tanks have different sizes. Moreover, a *single-order* tank is dedicated to a product family, a *single-product* tank is dedicated to one specific product. During the transfer from the blending tank, both the storage tank and connecting pipelines are occupied (Phase-In Storage). The product remains in the storage tank (Product Storage) until the pumping towards a filling unit starts. A minimum dwell time is necessary in the storage tank for quality control. During the filling (Phase-Out Storage) the storage tank and the transfer pipelines to the filling unit remain occupied.

Finally in Stage 4 (Filling) the product is bulk-loaded or drummed via a filling unit (Product Filling). Filling can start immediately after the minimum dwell time in the storage tank, but ideally it should be finished immediately before the customer order due-date, i.e., at the date it has to leave the factory. In case of (too) early production a maximum dwell time is imposed, after which the production order is filled in a costly additional container to free up space in the storage tank.

Using the description of the blending process above, the operations needed to schedule a production order while taking into account the finite capacity of the main production facilities can be summarized as follows (see also Figure 2).

- For every BOM-component, there is one *component pumping* operation at an appropriate component tank. The consecutive pumping operations do not overlap in time.
- In parallel with each of these pumping operations, there are multiple *component transfer* operations expressing the use of the necessary component pipelines to the blending tank. Depending on the path chosen, the number of pipelines differs.
- There is one *blending* operation on the blending tank. It overlaps at the start with the sequence of component pumping operations, lasts the blending and heating time, and finally overlaps with the product transfer to the storage tank.

- For the product transfer, multiple parallel *product transfer* operations express the use of the product pipelines needed to pump the product to the storage tank.
- On the storage tank, the *storage* operation includes the loading of the tank during the product transfer, the intermediate storage duration with a minimum and maximum dwell time, and the unloading of the tank via the filling pipelines.
- Finally, one *filling* operation represents the use of a filling unit and overlaps with the unloading of the storage tanks.

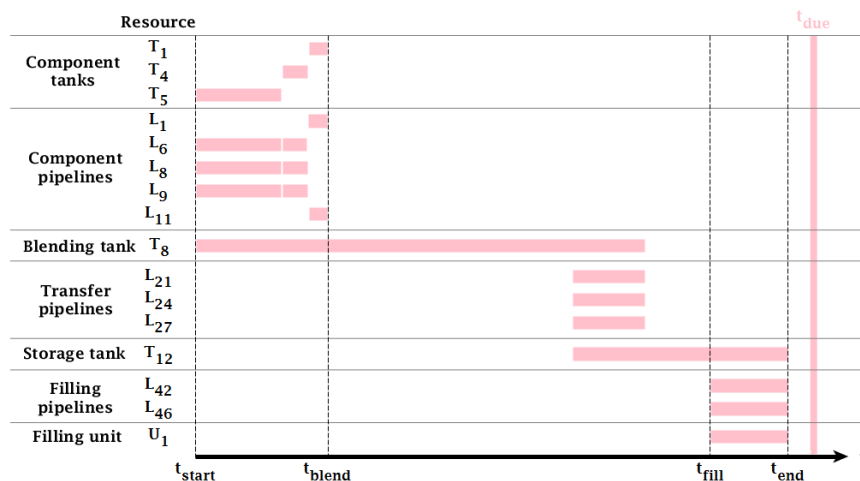


Figure 2: The scheduled operations of a production order on the different resources.

1.3 Related work

Computational scheduling models for batch production and the process industries have been proposed since the early 80's (Egli and Rippin, 1986), with pioneering work on integer linear models for production scheduling appearing as early as the late 50's (Wagner, 1959). The literature on scheduling methods for process industries has been thoroughly reviewed several times in the last decade (Floudas and Lin, 2004; Méndez et al., 2006; Henning, 2009; Harjunkski et al., 2014). There have also been some attempts to develop systematic frameworks, classifications, and unified notations for chemical production scheduling (Burkard and Hatzl, 2005; Méndez et al., 2006; Maravelias, 2012). Even though there is general agreement in the literature that a good optimization solution can result in significant savings through better capacity utilization, most reviews agree that a lot of work needs to be done especially in bridging the gap between theoretical models and practical applications of these models. Harjunkski et al. (2014), e.g., state that “most [...] methods are presented in the literature from a purely modeling point of view and they are mainly tested only on small-scale examples”.

145 A large fraction of the research on batch scheduling has focused on the develop-
ment of mixed-integer programming models, using either continuous-time or discrete-
time formulations (Floudas and Lin, 2004). Such models are then solved using either
problem-specific solution methods or commercial solvers such as CPLEX or Gurobi.
Problem-specific solution methods use strategies such as decomposition (e.g., Kopanos
et al., 2010; Kelly and Zyngier, 2008; Janak et al., 2006a), reformulation (e.g., Janak and
Floudas, 2008), and tightening (e.g., Velez et al., 2013), generally within a branch-and-
150 bound framework. Some approaches aim at *reactive* scheduling, i.e., updating the
production schedule when an unforeseen event occurs (e.g., Janak et al., 2006b).

An important aspect in the scheduling of a blending plant is the determination of
the best formula (also called recipe, or bill of material) to produce each product with.
Using the best BOM for each production order has an impact on inventory levels, as
155 well as on the number of switches that need to be made in blenders and other tanks.
This aspect has been tackled in the literature (e.g., Li et al., 2010; Li and Karimi, 2011;
Castillo and Mahalec, 2014a,b) and is also treated in this paper.

Most scheduling algorithms in the literature are implicitly designed to be used
in a Manufacturing Execution System (MES), to support the *operational* short-term
160 scheduling of the semi-process plant (Harjunkoski et al., 2014). The primary purpose
of the heuristic developed in this research, however, is to be integrated within a *tac-*
tical decision-making tool, allowing the chemical company to make decisions on the
optimal number of blending and/or storage tanks, the optimal dedication of resources
to product families, the optimal size of the workforce, etc. Such a tactical decision sup-
165 port tool requires an underlying scheduling engine that is able to quickly establish the
feasibility and performance of a suggested alternative configuration (Schittekat and
Sørensen, 2009), an approach which has been called *hierarchical optimization* (Van Es-
sen et al., 2011). In many cases, this will require the heuristic to schedule the orders of
a considerably longer period than necessary for short-term scheduling, to incorporate
170 the effects of seasonality, the fact that some products are only ordered infrequently,
etc. It also requires the heuristic to be able to assess (and optimize for) a set of objec-
tives other than the minimization of the total cost.

The algorithm described in this paper uses an approach called *rule-based* schedul-
ing (Haupt, 1989), in which the orders are first sorted according to some criterion
175 (called the *dispatching rule*) and then scheduled one by one in this order. Simple
scheduling rules such as the shortest processing time (SPT) heuristic or the earliest
due date (EDD) heuristic fall in this category. Capacity constraints on the resources,
as well as other constraints are generally taken into account during this constructive
phase. An obvious drawback of this family of approaches is their myopic behavior in
180 which early decisions may lead to suboptimal solutions. On the other hand, we will
demonstrate in this paper that the clever use of dispatching rule can lead to a heuristic
that is both fast, flexible, and produces high-quality solutions.

The algorithm is based on an iterative insertion technique that is guided by *dy-*
namic resource preferences. In this approach, the sequence in which the orders are
185 scheduled, is established first. Our approach follows that of Kudva et al. (1994) in that
it schedules the operations for an order based on various priorities. Therefore, our
algorithm operates in an hierarchical greedy fashion. Contrary to Egli and Rippin
(1986), our algorithm does not use an a priori enumeration of all potential routings,

since this is computationally intractable as the plant size increases or the scheduling horizon is extended.

Following this introduction we give a detailed explanation of the considered scheduling problem in Section 2. The heuristic algorithm is developed in Section 3. In Section 4 we describe the effect of different parameterizations of our heuristic for various scenarios, before we close with our conclusions in Section 6. The nomenclature used in this article is provided in Appendix 7.

2 Scheduling a petrochemical blending plant

In this section we define the scheduling model and describe its inputs and outputs. After outlining the model in Section 2.1 we describe the model input in Section 2.2. The characteristics of a schedule are defined in Section 2.3. In sections 2.4 and 2.5 we present the model objectives and the model constraints in a more formal way before giving an illustrative example in Section 2.6.

2.1 Outline

Let $\mathcal{H} = [t, \bar{t}] \subseteq \mathbb{N}_0$ be the *scheduling horizon* in which we want to place production orders. We assume an underlying discrete time representation with minute as the atomic unit.

Each order represents a batch that has to run through the production process that consists of the operations component pumping, component transfer, blending, product transfer and filling (see Section 1.2). In the model we consider tanks, pipelines and production units as *resources* that can be utilized by the different operations. Each resource has specific attributes (e.g., capacity, pumping speed) and an initial resource calendar with planned down times and shift breaks. The remaining time periods can be used to schedule the operations.

Resources are eligible for selected operations according to their type, e.g., blending tanks can only serve for blending operations. Furthermore, every blending tank can only process orders for products that are in its dedicated product family. This restriction applies to all tanks and filling units such that the product families, and therefore the products, have different potential routings (see Figure 3). However, these paths can overlap, potentially resulting in contaminations in the pipeline system. We do not explicitly model the corresponding cleaning measures but focus on their minimization through appropriate secondary objectives in the path selection (Section 2.4).

Products can often be produced using different formulas and, for this reason, have multiple BOMs. The choice of BOM to produce an order determines the tanks and pipelines that can be used for component pumping. Special decanting units can be used as a source for arbitrary components instead of the single-component tanks. These flexible facilities are generally used for low-volume components, because they provide component material at increased cost and reduced pumping speed compared to the regular component tanks.

The considered scheduling problem incorporates multiple objectives (e.g., component cost minimization, pipeline occupation minimization) originating from the dif-

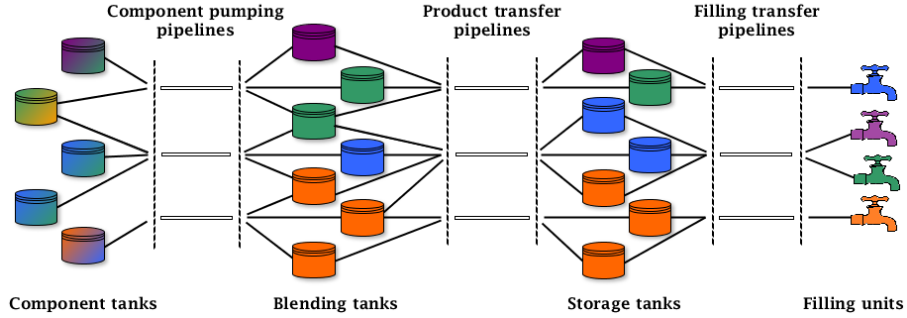


Figure 3: The dedicated resources in the production process.

230 ferent production stages. However, we focus on the primary objective of total order tardiness minimization and consider the others as secondary.

2.2 Scheduling scenarios

235 A *scheduling scenario* comprises all relevant information pertaining to the production environment used in the scheduling model. This includes the plant layout, the plant configuration and the production orders, all of which are explained in this section.

2.2.1 Plant layout

240 The *plant layout* summarizes all physical production-relevant resources. It includes the component tanks \mathcal{T}_{comp} , the blending tanks \mathcal{T}_{blend} , storage tanks \mathcal{T}_{store} , filling units \mathcal{U}_{fill} , as well as the pipeline network \mathcal{N} consisting of pipelines \mathcal{L} . The storage tanks can be classified in two different types: the single-order tanks \mathcal{T}_{SOT} and the single-product tanks \mathcal{T}_{SPT} . Each tank T has a capacity $vol(T)$. These tank-type resources are explained in the remainder of this subsection. An example of a plant layout for which we present a schedule in Section 2.6 is given in Figure 4.

245 **Component tanks** A component tank is used to store a specific component that needs to be pumped into a blending tank if it appears as an ingredient in the chosen BOM. We assume a replenishment policy that ensures the component's availability. Each component can also be added via one of the decanting units at a higher price and slower speed since the corresponding process is more laborious. We model these decanting units as component tanks that hold multiple components. Some low-volume components have to be pumped out of such a decanting unit since they are not assigned to their own component tank.

250

Blending tanks Blending tanks are used to produce the different products from their constituting components. They do not differ in terms of obtained product quality

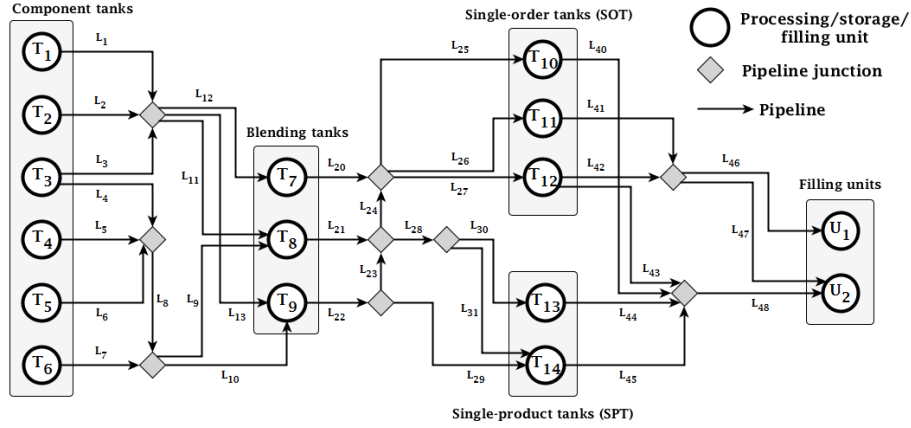


Figure 4: An example plant layout

or processing speed. In practice, stirring and air blending technology is used interchangeably. Nevertheless, the sizes of these tanks differ and in general the pipeline system connects them only to a subset of component tanks and storage tanks.

Single-product tanks A *single-product tank* (SPT) is a storage tank that is used to store one specific product, that can be used for different production orders at the same time. At each point in time $t \in \mathcal{H}$ each SPT $T \in \mathcal{T}_{SPT}$ has a certain material level $level^S(T, t)$ with respect to the schedule S . These levels increase or decrease after pumping material into or out of the tank respecting the tank's capacity. During such an injection or extraction no other order can access the tank resource. The product dwell time is bounded by a minimum dwell time $\underline{t}_{dwell}(T)$ (e.g., due to quality control measures) and a maximum dwell time $\bar{t}_{dwell}(T)$.

Single-order tanks A *single-order tank* (SOT) is a storage tank that can hold material for one production order at a time. The product that is stored in such a SOT $T \in \mathcal{T}_{SOT}$ has to belong to the product family to which T is dedicated. As for the SPTs, the dwell time is restricted for the SOTs.

Filling units A filling unit models a facility used to transfer the product into a packaging format. Typically, drums of different sizes are used. Alternatively, the material can be loaded into trucks by loading arms. Drum filling is slower than truck loading since it involves more manual interaction. An order is either a packaging order or a loading order, and therefore forced to use a suitable filling unit.

Pipeline network In contrast to many other scheduling models we do not assume a full connection scheme between the tanks of two consecutive process steps (e.g., not every storage tank can be reached from every blending tank). The pipeline network model contains *network nodes* that represent tanks, filling units and pipeline junctions.

The latter are not considered to be resources but just the points of connection for two pipelines. We assume that materials can be pumped through a pipeline in only one direction, and that material flows follow paths that do not intersect. Thus we represent the pipelines connecting nodes as *network arcs* from a start node to an end node. We basically allow an arbitrary pipeline network that follows the basic process scheme in Figure 3.

An important feature of model are the so-called *manifolds*, that route the material through the network. A manifold M can be seen as a black box that manages a subset of connections between its incoming pipelines and its outgoing pipelines through a complex system of valves. To model a manifold in our network we apply the following splitting technique. Each pipeline $L \in \mathcal{L}$ that enters M is represented by an arc A with an artificial end node N . Similarly, for a pipeline L' originating from M we obtain an arc A' with an artificial start node N' . If M can accommodate a material flow from L to L' then we add an arc from N to N' . Such a transformation is illustrated in Figure 5.

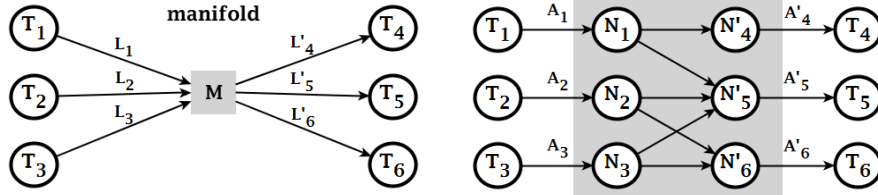


Figure 5: Modelling a manifold M in the pipeline network. The manifold M is replaced by artificial end nodes N_1, N_2 and N_3 , and start nodes N'_4, N'_5 and N'_6 . The arcs connecting these nodes represent the connections that M allows.

Pipeline cleaning system In-place pipeline cleaning is performed in between the pumping operations of incompatible products. We do not explicitly model the PIG (Pipeline Intervention Gadget) usage. However, the corresponding wear and maintenance costs play a role in the scheduling objectives presented in Section 2.4.

2.2.2 Production orders

We consider an *order set* O that contains all production orders that are due within the horizon \mathcal{H} . Each order $O \in O$ has the following sequence-relevant attributes.

- Ordered product $prod(O)$, order volume $vol(O)$, and expected profit $profit(O)$.
- Order date $t_{order}(O)$: the point in time when the order is placed, i.e., its earliest production start time.
- Due date $t_{due}(O)$: the point in time when the filling of the material should be completed in order to deliver the order on time.

305 Furthermore, we denote the length of a shortest period in which a production order can be fulfilled from the start of the pumping until the end of the filling as the *minimal order lead time* $t_{lead}(O)$. This duration may vary due to tank-dependent pumping speeds, filling speeds, and BOM components. Analogously, we define the *maximal order lead time* $\bar{t}_{lead}(O)$. In both cases we respect the dwell interval $[\underline{t}_{dwell}(T), \bar{t}_{dwell}(T)]$ 310 for each storage tank T . In the algorithm explained in Section 3 we will also use an *estimated order lead time* $\tilde{t}(O)$ which is the arithmetic mean of $t_{lead}(O)$ and $\bar{t}_{lead}(O)$.

2.2.3 Products and BOMs

Each product P in the product *portfolio* \mathcal{P} belongs to a *product family* $F \in \mathcal{F}$. This classification defines the product groups that can consecutively use resources without 315 causing contamination. A product BOM B specifies a set of components, their proportions in the recipe and the sequence in which they have to be added in the blending operation. The material cost per unit of product produced according to a BOM B is given by $c(B)$. This value can be deduced from the component costs, and is a significant factor in determining an optimal BOM for a production order. The same product 320 may be produced using different BOMs, here designated as $boms(P)$. It is assumed that all BOMs for a product P result in the same product quality.

2.2.4 Plant configuration

The *plant configuration* comprises plant properties relevant for production that can be typically modified on short or medium term. These are

- 325 • dedication of resources to specific materials,
- pumping and filling speeds,
- manifold connections, and
- production policies (e.g., working shifts, maintenance down times).

Component tanks and SPTs are dedicated to single components or products whereas 330 blending tanks, SOTs and filling units are dedicated to product families. This means that the resources can exclusively process or hold this one dedicated material, or all products of the dedicated family, respectively. The dedication of pipelines to specific materials is not needed in practice due to the presence of efficient in-place pipeline cleaning systems. However, such an extension could be integrated in our model and 335 our algorithm in a straightforward way.

The pumping and filling speeds are tank- and unit-dependent parameters that require empirical evidence-based information. The reconfiguration of manifolds is costly and only considered in conjunction with major changes in the production process or the product portfolio.

2.3 Scheduling decisions

340 In this section we introduce the decision variables of the scheduling problem. Even though we develop a heuristic algorithm, this formal representation allows us to unambiguously define the schedule properties (objectives and constraints) used in the

algorithm. The resulting formulation could also be used as a starting point for a mathematical programming approach. Because such an approach is not likely to result in an algorithm capable of solving the large real-life instances that we target, we will not pursue this research direction in this work.

Let $O \in \mathcal{O}$ be an order in a schedule S . Then the complete routing of the order through the production plant is determined by the decisions listed in Table 1.

Table 1: Base decisions that determine the routing for order O in schedule S .

Decision	Description
$bom^S(O)$	The selected BOM to produce $prod(O)$
$\mathcal{T}_{comp}^S(O)$	The sequence of component tanks
$\mathcal{L}_{pump}^S(O, C)$	The pipeline sequence used for pumping material C
$\mathcal{T}_{blend}^S(O)$	The blending tank
$t_{blend}^S(O)$	The blending start time
$\mathcal{L}_{transfer}^S(O)$	The pipeline sequence used for transferring the product
$\mathcal{T}_{store}^S(O)$	The product storage tank
$\mathcal{L}_{fill}^S(O)$	The pipeline sequence used for filling the product
$\mathcal{U}_{fill}^S(O)$	The filling unit
$t_{fill}^S(O)$	The filling start time

We assume that the remaining relevant points in time for O can be computed using the blending and filling start times. In particular, the *start time* $t_{start}^S(O)$ (i.e., the start of the component pumping) and the *end time* $t_{end}^S(O)$ (i.e., the end time of the product filling) can be calculated. For a summary of abbreviations and general notation we also refer to Table 4. Figure 2 illustrates these decisions for a single order.

2.4 Model objectives

The presented scheduling problem is of multi-objective nature. As it is typical for production environments, cost savings must be weighted against economy of time. To deal with this, we order the decisions that need to be taken according to their practical importance. For each of them we introduce a priority rule. The resulting sequence of prioritization steps characterizes our algorithm.

Let the *tardiness* of an order O in schedule S be $t_{tardi}^S(O) = \max\{0, t_{end}^S(O) - t_{due}(O)\}$ and its *earliness* $t_{earli}^S(O) = \max\{0, t_{due}(O) - t_{end}^S(O)\}$. Then the objectives are as listed below, in the mentioned order.

- i. *Minimization of the total tardiness*: To adhere to customer contracts and reduce shipping delays we prefer a low total tardiness, computed as the sum of the tardiness-es of the orders that could not be scheduled in time.
- ii. *Maximization of the total blending tank volume utilization*: The blender utilization ratio for an order is the corresponding volume compared to the vessel capacity. Since the duration of the actual mixing is volume independent an in-

- crease of this ratio increases the plant performance. Therefore, incrementing the sum of all these ratios is beneficial.
- iii. *Minimization of the total component costs*: The material (or component) cost for an order is induced by the selected BOM and the order volume. Shifting BOM can result in a significant cost reduction.
 - iv. *Minimization of the pumping pipeline cleaning*: Cleaning the pumping pipelines between incompatible batches by the in-place system is needed to avoid contamination. This procedure takes time and produces extra costs for maintenance.
 - v. *Minimization of the lead time*: The lead time for a production order, from the start of the pumping to the end of the filling, is affected by all the stages in the production process. Its reduction means an increased resource availability and less manufacturing costs. Moreover, it supports the finishing of ordered products on time.
 - vi. *Balancing workload among the filling units*: Except for low-volume periods we assume that the personnel needed to perform the filling on each unit is on duty during the work shifts. As preferred in most manufacturing environments we favor to evenly distribute the workload among the different resources.

2.5 Constraints

In the following we describe the (hard) constraints emanating from the production environment in the scheduling model.

Order constraints The production for an order O cannot start earlier than when it was created, whereas it can end after its due date. Thus, for a schedule S , we require

$$t_{start}^S(O) \geq t_{order}(O). \quad (1)$$

A packaging order has to be filled by a filling unit of type ‘package’ whereas a loading order has to use a filling unit of type ‘loading’.

Configuration constraints In our model we do not allow the re-dedication of resources to different materials over a scheduling horizon. During shift breaks all the resources except the storage tanks have to be empty or unused.

Process constraints The structure of the production process follows the sequence of operations as described in Section 1.2. Each order is produced in exactly one blending tank, stored in one storage tank and filled via one filling unit. Additionally, we assume that any order grouping or batching is done prior to importing the order set O and that orders can therefore not be combined in batches. Furthermore, material cannot be withdrawn from a tank while material is being added and, in general, a non-SPT resource can only process one order at a time.

Machinery constraints Orders can be filled early to avoid the violation of the maximal dwell time $\bar{t}_{dwell}(T)$ in a storage tank T . For SPTs the limitation is rather loose whereas it is tight (e.g., 24 hs) for SOTs due to inspection regulations. Neither the blending tanks nor the storage tanks are allowed to be filled with more material than their capacity admits. We neglect the different length of the pipeline connections and deduce the transfer time from the corresponding pumping speed of the source tank.

2.6 An illustrative example

Consider the plant shown in Figure 4. The plant consists of component tanks $\mathcal{T}_{comp} = \{T_1, \dots, T_6\}$, blending tanks $\mathcal{T}_{blend} = \{T_7, T_8, T_9\}$, SOTs $\mathcal{T}_{SOT} = \{T_{10}, T_{11}, T_{12}\}$, SPTs $\mathcal{T}_{SPT} = \{T_{13}, T_{14}\}$, filling units $\mathcal{U}_{fill} = \{U_1, U_2\}$ and 34 pipelines in \mathcal{L} . The filling unit U_1 operates as a loading facility and unit U_2 fills finished product into smaller packages. S contains orders $\mathcal{O}_S = \{1, \dots, 5\}$. The orders 1, 2 and 3 are completed in time whereas 4 and 5 are late. Note that component tank T_6 is not used in S as some of the pipelines. Figure 6 represents a valid schedule for this illustrative example. We use this example to describe the developed heuristic in Section 3.

3 The scheduling heuristic

In this section we describe the scheduling algorithm for the model presented in Section 2 using the introduced terminology.

As pointed out in Section 1.3, the heuristic in this paper belongs to the category of *rule-based* scheduling approaches (Haupt, 1989). It is essentially an iterative insertion technique guided by dynamic resource preferences, similar to the one developed in Kudva et al. (1994). First the sequence in which the orders are inserted is defined. Then the operations for an order are scheduled based on various priorities. The algorithm performs a gradual exploration of the possible production paths, guided by these priorities, in a greedy fashion. For each resource it dynamically maintains its available time windows.

3.1 Priority measures

In this section we explain the prioritization mechanisms used in the algorithm. For each asset x , the preference given to x is expressed by a *priority value* $\sigma(x)$.

3.1.1 Order prioritization

The order prioritization rule determines the sort order in which the production orders are scheduled. The algorithm re-prioritizes the unscheduled orders \mathcal{O}' in each main iteration of the algorithm. The prioritization rule used is the *earliest estimated latest start time*.

$$\sigma_{order}(O) = -(t_{due}(O) - \tilde{t}_{lead}(O)) \quad (2)$$

To evaluate the effect of the order prioritization strategy (2) we consider an alternative rule which prefers orders with large volume. Finding suitable resources for

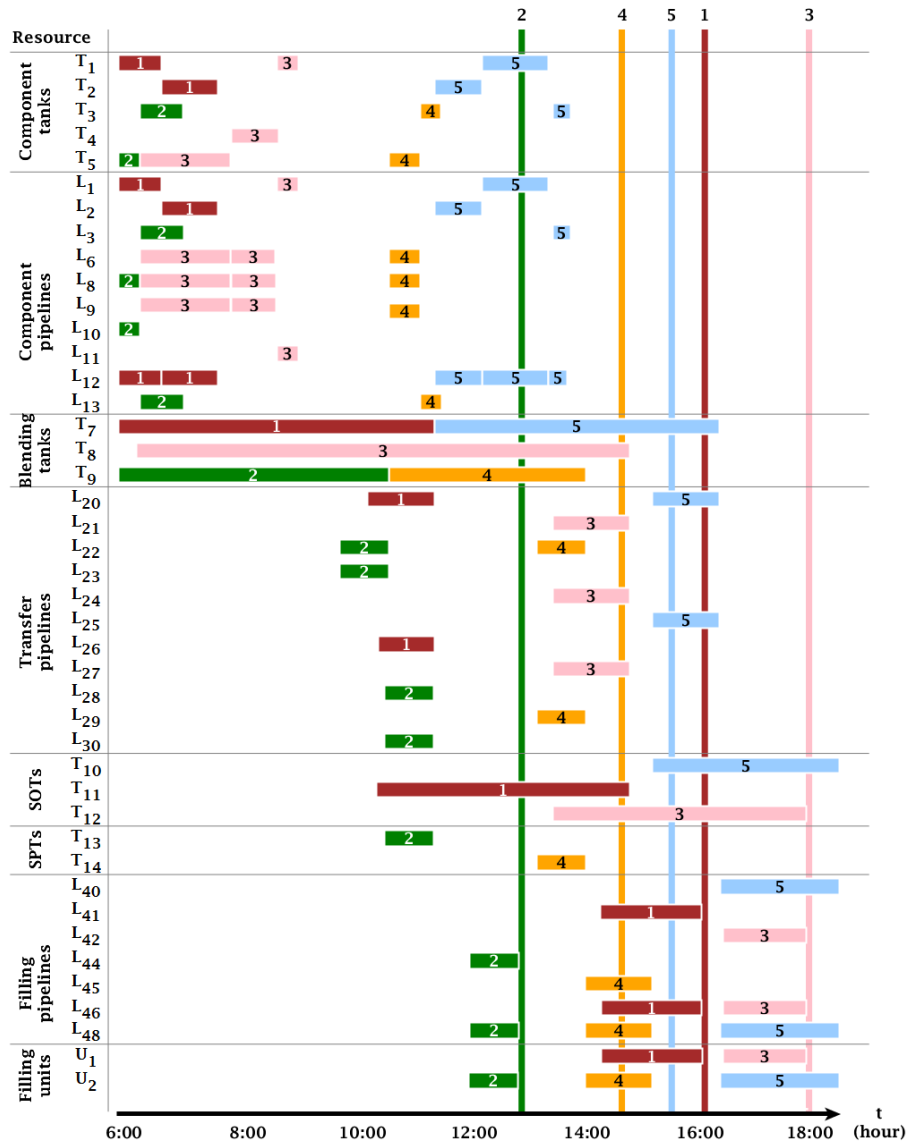


Figure 6: A Gantt chart representation of a schedule with 5 orders for the plant in Figure 4.

these high-volume orders tends to get harder in a busy schedule, and this strategy therefore aims to allocate resources to them as soon as possible.

$$\sigma'_{order}(O) = vol(O) \quad (3)$$

The result of this step is that the 5 production orders in Figure 6 are sorted: 1, 2, 3, 4, 5.

435 In reality, it is the responsibility of planning personnel to decide on the final production schedule. The actual operations can differ from the computed schedule because of short-term scenario changes, price volatilities or technical irregularities. Similarly, some orders might require ‘manual’ intervention of the planner to be scheduled (on time). To avoid an unrealistic delay for such an order in the schedule, we incorporate a maximum lateness parameter \tilde{t}_{late} in our algorithm. This allows that an order
440 remains unscheduled if it cannot be inserted less than e.g., 3 days late.

3.1.2 Blending prioritization

Instead of sequentially deciding on the blending tank and the blending time these decisions are linked in the algorithm. A *blending task* is defined by a pair (T_b, t_b) consisting of a blending tank T_b and a blending start time t_b . The use of blending tasks allows the algorithm to favor a tank T_b even when using this tank results in a lower utilization rate compared to another tank T'_b that is only available at inconvenient times regarding the order’s due date. The preference value of a blending task for an order $O \in \mathcal{O}$ is calculated by the following formula.

$$\sigma_{blend}(T, t, O) = \begin{cases} -\frac{vol(O)}{vol(T)} \left[\tilde{t}_{end} - t_{due}(O) \right]^{-1}, & \text{if } \tilde{t}_{end} < t_{due}(O) \\ -\frac{vol(T)}{vol(O)} \left[\tilde{t}_{end} - t_{due}(O) \right], & \text{if } \tilde{t}_{end} > t_{due}(O) \\ \frac{vol(O)}{vol(T)}, & \text{if } \tilde{t}_{end} = t_{due}(O) \end{cases} \quad (4)$$

Here \tilde{t}_{end} represents the estimated production end time for order O when blending in tank T at time t . This rule generally gives the highest priority to processes that are
445 expected to finish on time. Secondly, it favors finishing close to the due date. Finally, it takes into account the tank utilization to avoid wasting blending capacity.

Although a blending task can take place at any moment within the time horizon, it is computationally intractable to evaluate more than a limited set of start times t for each blending task. The algorithm therefore derives potential start times from the
450 utilization of suitable blending tanks in the current schedule instead. More specifically, it tries to use each compatible blending tank immediately before and immediately after each other job that occupies it. Potential start times between two consecutive time windows during which the blending tank is unavailable, are determined using a time step parameter $t_{blendStep}$. Additionally, we allow blending at the earliest possible point in time $t_{blend}(O) = t_{order}(O) + \tilde{t}_{pump}(O)$, as well as the latest occasion
455 to be in time $t_{due} - \tilde{t}_{blend2filled}$, or as late as possible: $t_{due} + \tilde{t}_{late} - \tilde{t}_{blend2filled}$, with $\tilde{t}_{blend2filled} = \tilde{t}_{blend} + \tilde{t}_{transfer} + \tilde{t}_{fill}$. Here we do not expect to know the actual duration of

the remaining operations and, therefore, use the estimated order pumping, blending, transfer, and filling times.

460 Using this strategy we maximize the blending tank capacity utilization and minimize the total tardiness $\sum_{O \in \mathcal{O}_S} t_{tardi}^S(O)$ in a schedule S (objectives i. and ii.).

An alternative blending task prioritization is the following rule which focuses on the maximization of the scheduled volume.

$$\sigma'_{blend}(T, t, O) = -t \quad (5)$$

Using this rule, the blending process of order O is scheduled as early as possible, but not before $t_{blend}(O)$. This prioritization strategy yields a sequence of blending tasks for each of the 5 production orders in Figure 6. E.g., the sequence for order 1 contains 465 the blending task $(T_7, 6:00)$ which is finally chosen, as depicted in Figure 6.

3.1.3 Storage tank prioritization

Analogous to the blending task we say that a *storage task* consists of a storage tank and a start time for the actual product storage. We generally prefer the storage of a finished product P in an SPT than in an SOT for two reasons. Since the SOTs can hold various products we rather keep them open for products that do not have a dedicated SPT. Secondly, we are interested in a high SPT utilization since they can hold more volume on average and typically allow longer product dwell times. Moreover, the following preferences with respect to storage tanks are taken into account. If the chosen tank T is an SOT we try to maximize the capacity utilization, i.e., choose the tank that fits the order volume best. We say that the *free capacity* of a tank $T \in \mathcal{T}_{SPT}$ at time t is $vol_{free}^S(T, t) = vol(T) - level^S(T, t)$. In case T is an SPT we prefer to keep the maximal free capacity among the feasible SPTs as high as possible to be prepared for the insertion of large volume orders.

$$\sigma_{store}(T, t) = \begin{cases} \frac{1}{vol_{free}^S(T, t)}, & \text{if } T \in \mathcal{T}_{SPT} \\ \frac{vol(O)}{vol(T)}, & \text{if } T \in \mathcal{T}_{SOT} \end{cases} \quad (6)$$

Figure 6 shows that the SOT T_{11} is selected for order 1 based on this prioritization of the storage tanks.

3.1.4 Filling prioritization

Orders have to be either packaged, or loaded into trucks, via the available filling units. In order to balance the workload (objective vi.) we prefer a suitable unit $U \in \mathcal{U}_{fill}$ with low utilization in the current schedule S in terms of time. We set the priority of each available unit U to

$$\sigma_{fill}(U) = \frac{1}{\mu^S(U)}, \quad (7)$$

470 where $\mu^S(U)$ is the total time that U is occupied in S . Note that we are restricted to units that are available and compatible with the considered order's product family.

A *filling task* specifies a filling unit, the used pipeline path and the pumping start time. In the schedule in Figure 6, filling unit U_1 is used together with pipelines L_{41} and L_{46} for order 1 at about 14:00. Order 4 is assigned unit U_2 instead, since its filling operation intersects with the one for order 1.

3.1.5 Product BOM prioritization

For each order, a BOM needs to be selected, indicating a component sequence needed to feed the blending operation. There are two reasons for alternative BOMs. On the one hand multiple components C_1, \dots, C_k can be acquired pre-mixed as a single aggregating component C_Σ instead of being added separately during the component pumping. Even though C_Σ might be more expensive to purchase, the shortage of a single component C_i ($i \in \{1, \dots, k\}$) or the unavailability of pumping pipelines might make such a decision either desirable or necessary. On the other hand the availability of a comparable material at a lower price can also impact the BOM selection. We incorporate this monetary objective by giving higher priority $\sigma_{BOM}(B)$ to a BOM B that has a lower unit material costs as follows.

$$\sigma_{BOM}(B) = \frac{1}{c(B)} \quad (8)$$

This strategy aims at the minimization of the material costs (objective iii.). The BOM for order 1 is implicitly given in Figure 6 by the component tanks T_1 and T_2 which are used to pump the necessary ingredients. The BOM that is chosen for order 5 requires 3 components which are pumped out of tanks T_1 , T_2 and T_3 . As can be seen in Figure 4, these component tanks are the only ones from which ingredients can be pumped into the used blender T_7 . Note that the pumping for order 3 cannot start earlier because blending tank T_7 is occupied by order 1.

3.1.6 Component pumping prioritization

To determine the pumping operation for an order the algorithm needs to choose suitable component tanks for the components of the selected BOM B . We recall that a component C can be retrieved from multiple component tanks. It can also be added just in time via the slower decanting units, which are also modeled as component tanks. This gives rise to a variety of pumping paths to the selected blending tank. We prefer to use components in tanks with higher pumping rates to minimize the total pumping duration.

Furthermore, consider the pumping of C through a pipeline L . This pipeline might need to be cleaned if the subsequent material for an incompatible production order is contaminated by the residue of C . The minimization of these cleaning operations is objective iv., and the minimization of the pumping duration contributes to objective v., i.e., the lead time minimization. We prioritize feasible component tanks T_C and the used pumping pipelines L_P by first minimizing the number of pipelines $|L_P|$ used and, secondly, by minimizing the *pumping duration* $t_{pump}(O, B, T_C)$.

$$\sigma_{pump}(T_C, L_P, O) = -|L_P| + \frac{1}{t_{pump}(O, B, T_C)} \quad (9)$$

We say that a *pumping task* consists of the sequence of component tanks and the pumping start time. The components for order 2 in Figure 6 are pumped from tanks T_5 and T_3 in the plant given in Figure 4. The component from tank T_3 is pumped via pipelines L_3 and L_{13} to blending tank T_9 . The number of utilized pipelines is 2 which is favorable, according to the above priority, compared to the usage of pipelines L_4 , L_8 and L_{10} , alternatively.

3.2 Schedule polishing

To further refine the scheduling decisions we apply a post-processing procedure. This is necessary due to the fact that several scheduling decisions are taken based on an estimation of the duration of a set of operations. For instance, when considering the start of a blending process the algorithm does not yet know the exact dwell and filling times. For this reason, the following polishing techniques are performed on the scheduled orders.

Routing compacting The lead time is reduced by shortening the dwell time in the storage tank without changing the order's resources.

Routing shifting The whole routing is shifted forward in time in order to reduce the order's tardiness.

Both strategies are applied until no more improvement can be found. A time step parameter $t_{polishStep}$ is used to check dwell time reductions and routing shifts.

3.3 Overall algorithm

The scheduling algorithm iteratively tries to assign resources of highest priority value, given by the σ -functions, to the different tasks that need to be executed. The order in which this search is carried out is based on the knowledge that the blending stage plays a central role in the production and is assumed to be susceptible to bottleneck situations in many plants. Thus we implemented the following strategy.

Initially, the orders are ranked according to their priority. This list is worked off until all the orders were either scheduled or skipped. The latter can happen in situations where we are not able to schedule the order within the feasible time window (earliest start date, maximal earliness/lateness). In each iteration we first attempt to schedule the order's blending operation on the most preferred blending tank at the preferred time. The transfer pipeline path and the storage tank is selected afterwards. To enable a product dwell of variable length in the storage tanks we incorporate a dwell prolongation step-size $t_{dwellStep}$. Typically, the step size is set to a value between 10 and 30 minutes. This is for instance needed to take into account overnight tank dwells and to wait for filling unit availability. We recall that each storage tank has a lower and an upper bound on the dwell time for the product it contains.

To meet the due date we try to schedule the order's filling operation on either a packaging unit or a loading unit (depending on the order type). Finally, we consider the component pumping phase for the order. The evaluation of BOMs in order of their preference is followed by the pumping sequence selection, the allocation component

tanks and the used pumping paths. Note that the start time of a component pumping can be calculated by subtracting the subsequent component pumping times from the blending start time. This completes the production process for the order.

535 Whenever we fail to reserve the resources of highest priority in one of these steps the algorithm backtracks and considers the next best option in the cascading scheme. Figure 7 illustrates the complete algorithm.

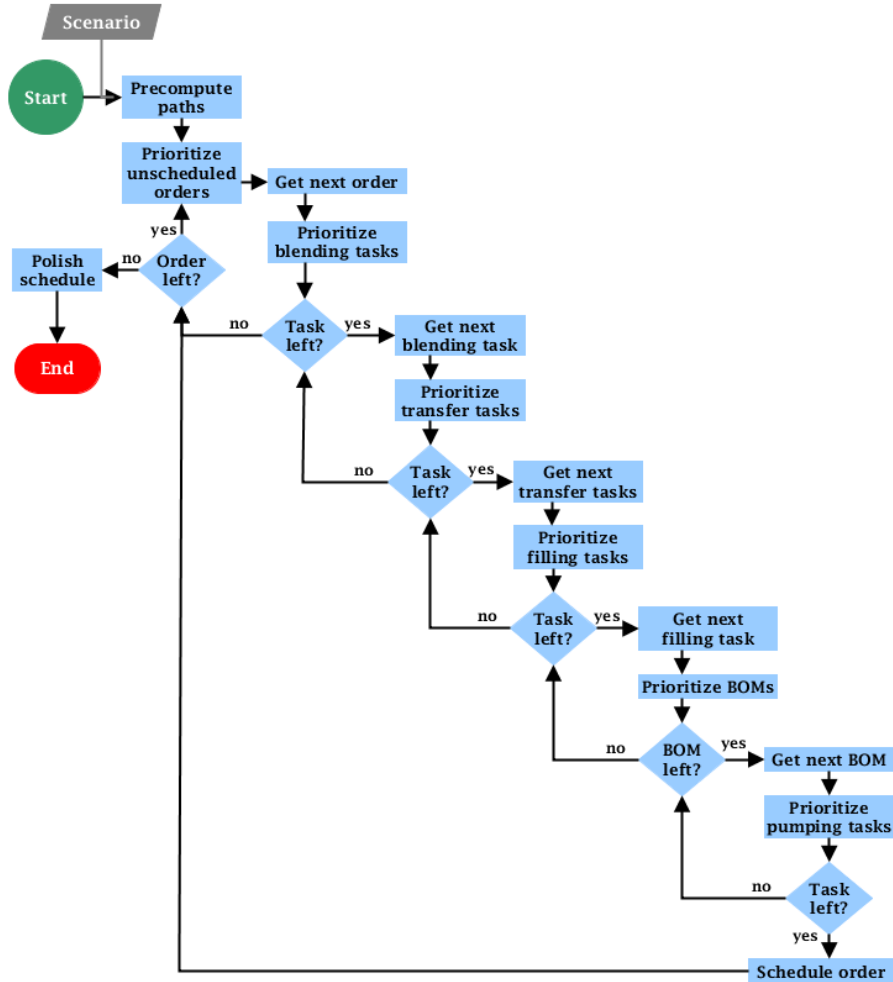


Figure 7: The overall algorithm.

3.4 Complexity and efficiency

540 Both exact and heuristic scheduling algorithms generally suffer from the very large number of possible solutions that characterize most scheduling problems. However,

a careful implementation can increase algorithm performance and reduce the effects of problem complexity. We therefore incorporated some tuning measures, which are described below.

545 The number of considered blending tasks can be reduced by requiring the blending tank to be available during the shortest possible preceding pumping phase and the shortest product transfer after the actual blending. Similarly, the storage tank selection can be restricted by looking ahead for the minimum storage dwell time or the dwell time until the filling should start to meet the due date.

550 We precompute the feasible pipeline paths between component tanks and blending tanks, between blending tanks and storage tanks, and between storage tanks and filling units respecting the tank dedications. All possible connecting paths are enumerated using depth-first search in the pipeline network in a preprocessing phase and stored for efficient access. Note that there can be multiple paths between a start and an end node.

555 4 Computational study

In this section we study the algorithm on a scenario of realistic size. We investigate two different aspects. First, we analyze the computational effort needed to take decisions in the various stages. Secondly, we compare the impact of our prioritization rules on the quality of the obtained schedule.

560 The base scenario used for the experiments has about 140 tanks and production units, and more than 300 pipelines. The orders have an average volume of 15 m^3 whereas the blending facilities can hold 50 m^3 on average. More than 100 products are grouped in 6 product families. For confidentiality reasons we are not able to give a more detailed analysis of the order data and the plant specifications.

565 We set the parameters $t_{blendStep}$ and $t_{dwellStep}$ to 20 minutes. For the polishing phase, $t_{polishStep}$ was set to 10 minutes.

The described algorithm was implemented in C++ and compiled with the Visual C++ Express compiler. The computations were carried out on a Microsoft Windows system equipped with an Intel Core I7-3630QM (2.4GHz) processor and 8 GB RAM.

570 4.1 Search analysis

We are interested in the distribution of the numbers of iterations needed in the different stages of the scheduling procedure. More precisely, we would like to get an idea where the algorithm makes the most effort to determine suitable resources. For instance, choosing the blending kettle for an order might happen after neglecting $k - 1$ blenders. If l is the total number of blending tasks that were prioritized for this order then the ratio k/l indicates a relative effort needed to take the decision. Figure 8 depicts the corresponding distribution for all the scheduled orders in each stage of the production using notched box plots. The data was derived from a scheduling run on an underlying scenario with 100 orders.

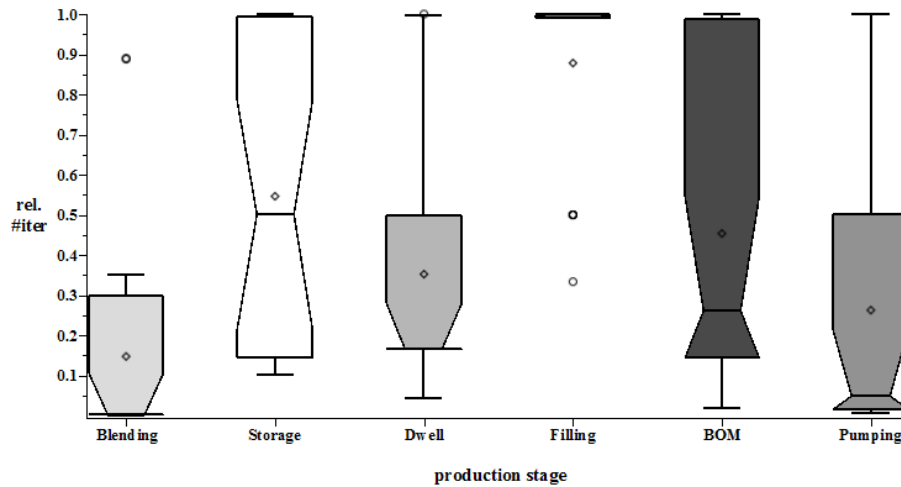


Figure 8: Relative number of iterations needed to find suitable resources in each stage of the production process for all the scheduled orders.

580 We observe that it is rather costly to find a preferred transfer path (including the storage tank) and an appropriate BOM. Suitable blending tanks could mostly be found rather efficiently by the heuristic. The reason for the high ratios in the filling stage is that each storage tank is uniquely connected to a filling unit in the plant and for each product family there exists exactly one such dedicated unit.

585 4.2 Strategy analysis

In this section we study the performance of the algorithm on different scenarios. These scenarios span 1, 7, 30, 90, 180 and 360 days and contain about 15, 70, 400, 1200, 2600 and 6000 orders. Moreover, we consider three working shift regimes: 1 shift, 2 shifts, and 3 shifts.

590 We compare the two order prioritization rules presented in Section 3.1.1. Table 2 shows the results for priority rule (2) whereas Table 3 lists results for rule (3). In both cases we experiment with the two blending prioritization ideas in Section 3.1.2: on time (4) and earliest (5). The first column (**shifts**) contains the number of working shifts, followed by the column indicating the horizon days (**days**). The rule to prioritize the blending tasks is given in column **blend.strat.**. Column **t(s)** contains the run time of the heuristic in seconds and **iter**·10⁻³ the number of iterations divided by 1000. The average order lead time (minutes) is **t_{lead}(θ)** and **late(%)** is the percentage of orders that could not be finished on time. The average tardiness of such late orders is given in column **tardi(θ)** in minutes. Column **#pig** shows the number of needed
600 PIG-cleaning operations.

Table 2: Computational results for order prioritization rule (2): earliest latest start time.

Scenario		Heuristic	Schedule					
shifts	days	blend.strat.	t(s)	iter·10 ⁻³	$t_{lead}(o)$	late(%)	tardi(o)	#pig
1	1	Earliest	2	4	1489	21,4	545	24
1	1	OnTime	4	1	728	21,4	495	23
1	7	Earliest	10	60	998	10	79	87
1	7	OnTime	6	49	578	18,5	132	77
1	30	Earliest	60	926	974	28,9	442	413
1	30	OnTime	46	784	608	45,4	998	392
1	90	Earliest	287	2455	1052	21,2	265	1262
1	90	OnTime	158	2334	632	45,7	853	1208
1	180	Earliest	1048	6231	1054	24	293	2779
1	180	OnTime	691	6420	628	50,6	967	2664
1	360	Earliest	5727	13751	1051	26	335	5621
1	360	OnTime	3569	13343	640	50,1	923	5414
2	1	Earliest	3	4	1514	7,1	191	21
2	1	OnTime	1	2	597	14,2	197	23
2	7	Earliest	12	31	1153	0	0	83
2	7	OnTime	7	11	472	23,2	43	75
2	30	Earliest	116	483	1065	4,6	12	368
2	30	OnTime	52	510	506	29,4	81	353
2	90	Earliest	547	1480	1102	3,1	11	1181
2	90	OnTime	216	1318	520	28,5	73	1102
2	180	Earliest	3000	3669	1140	2	6	2494
2	180	OnTime	950	3006	514	31	82	2324
2	360	Earliest	12040	7505	1160	1,8	7	5073
2	360	OnTime	5637	6426	512	32,7	79	4734
3	1	Earliest	2	2	649	0	0	25
3	1	OnTime	4	3	508	21,4	29	22
3	7	Earliest	25	14	867	0	0	83
3	7	OnTime	8	10	374	27	60	73
3	30	Earliest	89	240	1066	0	0	371
3	30	OnTime	58	315	337	18,7	35	354
3	90	Earliest	266	768	1066	0	0	1164
3	90	OnTime	202	901	340	17,8	34	1113
3	180	Earliest	626	1534	1062	0	0	2421
3	180	OnTime	521	1737	342	20,1	36	2346
3	360	Earliest	1488	3061	1063	0	0	4923
3	360	OnTime	1134	3497	341	19,8	37	4717

Table 3: Computational results for order prioritization rule (3): biggest volume.

Scenario		Heuristic	Schedule					
shifts	days	blend.strat.	t(s)	iter10* ⁻³	t _{lead} (θ)	late(%)	tardi(θ)	#pig
1	1	Earliest	2	4	1400	21,4	551	22
1	1	OnTime	4	1	649	21,4	497	22
1	7	Earliest	9	82	850	12,8	88	86
1	7	OnTime	7	70	536	15,7	116	83
1	30	Earliest	82	1080	945	24,3	450	414
1	30	OnTime	56	888	699	33,1	573	378
1	90	Earliest	350	2828	1040	21,9	371	1275
1	90	OnTime	245	2516	708	31,3	498	1173
1	180	Earliest	1793	7068	1038	23,6	384	2749
1	180	OnTime	1210	6700	697	32,5	505	2639
1	360	Earliest	10203	15726	1050	23,8	400	5562
1	360	OnTime	6503	13970	703	32,3	518	5413
2	1	Earliest	4	4	1514	7,1	191	21
2	1	OnTime	1	2	615	21,4	201	21
2	7	Earliest	21	31	1087	0	0	81
2	7	OnTime	5	9	467	26	47	74
2	30	Earliest	129	525	992	3,3	25	359
2	30	OnTime	71	434	515	27,1	149	350
2	90	Earliest	543	1757	1024	4	43	1159
2	90	OnTime	492	1467	539	27,9	149	1103
2	180	Earliest	2611	3941	1061	3,3	35	2445
2	180	OnTime	3570	3366	524	27,8	135	2351
2	360	Earliest	10439	7901	1066	3,5	30	4943
2	360	OnTime	12539	6944	536	28,6	152	4801
3	1	Earliest	3	3	654	0	0	25
3	1	OnTime	3	4	452	0	4	24
3	7	Earliest	23	17	737	0	0	85
3	7	OnTime	7	10	370	22,9	55	78
3	30	Earliest	108	299	935	0	0	381
3	30	OnTime	47	257	349	19,4	36	353
3	90	Earliest	428	1144	936	0,9	10	1148
3	90	OnTime	413	899	351	16,9	34	1108
3	180	Earliest	1548	2388	959	0,5	5	2410
3	180	OnTime	2438	1775	348	18,1	34	2373
3	360	Earliest	5980	4341	956	0,4	3	4901
3	360	OnTime	7411	3356	348	18	38	4790

As expected, we observe that the computational effort (**t(s)**) increases more than linearly when increasing the scheduling horizon and the number of orders, respectively. This is mainly due to the fact that the insertion of orders in later iterations (more orders scheduled) requires dealing with resources with a large number of non-availabilities. This is supported by the observation that the run time increases significantly faster than the number of iterations. Still the algorithm is able to schedule all orders of an entire year in an acceptable time (< 3 hours) in the tested case studies.

Order prioritization rule (3), which attempts to insert orders of larger volume first, outperforms rule (2). Mainly due to the cases in which the plant operates in 2 shifts we observe an average increase of 26% in tardiness and an increase of the total number of late orders by 22% when using (2). However, the run times are 28% less on average when using (2).

Independent from the order prioritization rule we observe that selecting the earliest blending task is superior to the on-time strategy regarding the total tardiness. However, this yields extended lead times due to larger dwell times in the storage tanks.

As an example for the usefulness of approach to provide decision support for tactical planning we consider the different shift scenarios in Table 3. For the listed scenarios we see that working in a 1-shift mode results in a significant number of late orders. An augmentation to 3 shifts yields notable improvement regarding the num-

ber of late orders as well as the order lead time and tardiness. However, such a change should be weighted against the increased variable costs.

5 Real-life validation and use of the heuristic

As indicated in the introduction, the main purpose of our scheduling heuristic was to support tactical decisions on the design of a real-size chemical production plant. In particular, the aim was to investigate the efficiency of the plant layout (number, capacity and location of production facilities) and plant configuration (dedication of production families, shift patterns and the use of pipeline connections). Therefore, a prerequisite for our solution method was to tackle, in a reasonable computation time, a planning problem with an horizon of several months to a year and with the huge dimensions and high complexity of a real petrochemical blending plant.

Consequently, it was not possible to compare the quality of our solutions with existing solutions based on exact or heuristic solution methods. On the one hand, solution methods proposed in the literature do not tackle or report about problems of this size and complexity, and testing our heuristic on smaller and/or simpler examples was not relevant. On the other hand, the professional scheduling system used in the plant, is meant for (very) detailed scheduling of production orders for the next five working days. Even this system, configured for the actual layout and configuration of our blending plant, had to relax some detailed constraints (constraints which are also not considered in our tactical schedule) in order to avoid initial performance issues.

The real-life validation of our scheduling heuristic, as well as its further use by the operations management team, took place during the start-up phase of the plant. During this phase, which lasted over several months, our heuristic was used to determine an estimate of the production capacity of the plant, by — starting from a given configuration of the plant — estimating its expected production capacity. To this end, the orders of the previous year were scheduled on several potential plant configurations. The capacity originally designed by the engineering team was then compared to the actual capacity observed by the operational team, and to the design capacity predicted by our scheduling heuristic. In general the original design capacity matched the capacity predicted by our heuristic (based on the customer orders of the previous year), except when temporary or definitive changes to the plant design took place. When the actual capacity deviated from the design capacity, this was generally due to the introduction of new products, new production strategies for packaging, and typical startup issues. As common in this type of industrial projects, several issues concerning data quality had to be solved first.

In the last phase of this research project, the scheduling heuristic was used to test new scenarios for the dedication of product families to blending tanks and for the dimensioning of the single-order loading tanks. The outcome of the scheduling heuristic and related (confidential) capacity views proved to be useful to support the decisions for improvement. Discussions on further use of the heuristic and integration in the professional ICT environment of the company have been started, but were out of the scope of this research project.

In conclusion, the quality of the solutions produced by our heuristic was success-

665 fully demonstrated “on site”, and was instrumental in the optimal design of a new
 blending plant. Testing our heuristic on other plant designs, as well as the compar-
 670 ison with other techniques that can tackle these kind of real-size design problems, is
 left for future research.

6 Conclusions

670 In this paper we have presented a batch scheduling heuristic that we developed in the
 context of a research project in the chemical industry. We integrated various practical
 aspects emanating from the underlying state-of-the-art blending plant into a formal
 model. We explicitly modeled the complex pipeline system that is used to transfer the
 materials between the different stages as well as the wide product portfolio in which
 most products can be produced according to different formulations.

675 The scheduling heuristic incorporates several prioritization mechanisms that guide
 the search in a greedy fashion. We compared different priority rules for the order and
 blender selection their individual effect on the minimization of the total tardiness.

680 The two main strengths of the presented method are the following. On the one
 hand the heuristic is able to efficiently compute schedules by respecting various highly
 relevant constraints. The exchangeable prioritization rules guide the search and can
 be adapted to follow multiple objectives. On the other hand the presented method can
 be used to provide decision support when integrated in a tactical planning tool. By
 appropriately setting parameters such as production policies, plant layout, and plant
 configuration, the impact on mid to long-term horizons can be studied.

685 7 Appendix

Table 4: Nomenclature.

Scenario data	
O	Order set
\mathcal{P}	Set of all products
\mathcal{F}	Set of all product families
\mathcal{T}	Set of all tanks
\mathcal{T}_{comp}	Set of component tanks
\mathcal{T}_{blend}	Set of blending tanks
\mathcal{T}_{SOT}	Set of single order tanks (SOTs)
\mathcal{T}_{SPT}	Set of single product tanks (SPTs)
\mathcal{U}_{fill}	Set of filling units
\mathcal{L}	Set of all pipelines
\mathcal{L}_{comp}	Set of component pumping pipelines
$\mathcal{L}_{transfer}$	Set of product transfer pipelines
\mathcal{L}_{fill}	Set of filling pipelines
\mathcal{H}	Scenario horizon interval $[t, \bar{t}]$
\bar{t}	Scenario horizon end
$\bar{t}_{dwell}(T)$	Maximum product dwell time in storage tank T

$\underline{t}_{dwell}(T)$	Minimum product dwell time in storage tank T
\underline{t}	Scenario horizon start
$\underline{vol}(T)$	Volume capacity of tank T
$t_{blend}(O, T)$	Blending time needed for order O in tank T
<hr/>	
Attributes of order O	
$prod(O)$	Order product
$t_{due}(O)$	Order due date
$t_{order}(O)$	Order date
$vol(O)$	Order volume
$profit(O)$	Order profit
$type(O)$	Order filling type: packaging or loading
$\tilde{t}_{lead}(O)$	Estimated order lead time
$\bar{t}_{lead}(O)$	Maximal order lead time
$\underline{t}_{lead}(O)$	Minimal order lead time
$t_{pump}(O, B, T_C)$	Pumping duration using BOM B component tanks T_C
$\underline{t}_{blend}(O)$	Earliest blending start time
<hr/>	
Product attributes	
$c(B)$	Per unit cost of a BOM B
$\mathcal{F}(P)$	The product family of product P
$boms(P)$	Set of available BOMs for product P
<hr/>	
Attributes of schedule S	
O^S	Set of scheduled orders
$level^S(T, t)$	Level of SPT T at time t
$vol_{free}^S(T)$	Free capacity of SPT T at time t
$t_{start}^S(O)$	Start time of order O (start of pumping)
$t_{end}^S(O)$	End time of order O (end of loading)
$t_{earli}^S(O)$	Earliness of order O with respect to its due date ($\max(0, t_{due}(O) - t_{end}^S(O))$)
$t_{tardi}^S(O)$	Tardiness of order O with respect to its due date ($\max(0, t_{end}^S(O) - t_{due}(O))$)
$bom^S(O)$	BOM used for order O
$\mathcal{T}_{comp}^S(O)$	Component tank sequence for order O
$\mathcal{L}_{pump}^S(O, C)$	Pipeline sequence used to pump component C for order O
$t_{blend}^S(O)$	Actual blending start time for order O
$\mathcal{T}_{blend}^S(O)$	Blending tank for order O
$\mathcal{L}_{transfer}^S(O)$	Pipeline sequence to transfer order O to the storage tank
$\mathcal{T}_{store}^S(O)$	Storage tank for order O
$\mathcal{L}_{fill}^S(O)$	Pipeline sequence to conduct order O to the filling unit
$t_{fill}^S(O)$	Filling start time for order O
$\mathcal{U}_{fill}^S(O)$	Filling unit for order O
<hr/>	
Heuristic parameters	
$\sigma(x)$	Priority value of a task, tank, filling unit or BOM x
\bar{t}_{late}	Maximal lateness for a scheduled order
$t_{blendStep}$	Time step between two potential blending starts
$t_{dwellStep}$	Time step between two potential storage tank dwell

$t_{polishStep}$	times Time step used to compact or shift scheduled orders during polishing
------------------	--

References

- R.E. Burkard and J. Hatzl. Review, extensions and computational comparison of MILP formulations for scheduling of batch processes. *Computers & Chemical Engineering*, 29(8):1752–1769, 2005.
- 690 P.A. Castillo and V. Mahalec. Inventory pinch based, multiscale models for integrated planning and scheduling-part i: Gasoline blend planning. *AIChE Journal*, 60(6): 2158–2178, 2014a.
- P.A. Castillo and V. Mahalec. Inventory pinch based, multiscale models for integrated planning and scheduling-part ii: Gasoline blend scheduling. *AIChE Journal*, 60(7):
695 2475–2497, 2014b.
- U.M. Egli and D.W.T. Rippin. Short-term scheduling for multiproduct batch chemical plants. *Computers & Chemical Engineering*, 10(4):303–325, 1986.
- C.A. Floudas and X. Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*,
700 28(11):2109–2129, 2004.
- I. Harjunkoski, C.T. Maravelias, P. Bongers, P.M. Castro, S. Engell, I.E. Grossmann, J. Hooker, C. Méndez, G. Sand, and J. Wassick. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62:161–193, 2014.
- 705 R. Haupt. A survey of priority rule-based scheduling. *Operations-Research-Spektrum*, 11(1):3–16, 1989.
- G.P. Henning. Production scheduling in the process industries: current trends, emerging challenges and opportunities. *Computer Aided Chemical Engineering*, 27:23–28, 2009.
- 710 S.L. Janak and C.A. Floudas. Improving unit-specific event based continuous-time approaches for batch processes: Integrality gap and task splitting. *Computers & Chemical Engineering*, 32(4):913–955, 2008.
- S.L. Janak, C.A. Floudas, J. Kallrath, and N. Vormbrock. Production scheduling of a large-scale industrial batch plant. i. short-term and medium-term scheduling. *Industrial & engineering chemistry research*, 45(25):8234–8252, 2006a.
715
- S.L. Janak, C.A. Floudas, J. Kallrath, and N. Vormbrock. Production scheduling of a large-scale industrial batch plant. ii. reactive scheduling. *Industrial & engineering chemistry research*, 45(25):8253–8269, 2006b.

- 720 J.D. Kelly and D. Zyngier. Hierarchical decomposition heuristic for scheduling: Co-ordinated reasoning for decentralized and distributed decision-making problems. *Computers & Chemical Engineering*, 32(11):2684–2705, 2008.
- G.M. Kopanos, C.A. Méndez, and L. Puigjaner. Mip-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. *European Journal of Operational Research*, 207(2):644–655, 2010.
- 725 G. Kudva, A. Elkamel, J.F. Pekny, and G.V. Reklaitis. Heuristic algorithm for scheduling batch and semi-continuous plants with production deadlines, intermediate storage limitations and equipment changeover costs. *Computers & Chemical Engineering*, 18(9):859–875, 1994. An International Journal of Computer Applications in Chemical Engineering.
- 730 J. Li and I.A. Karimi. Scheduling gasoline blending operations from recipe determination to shipping using unit slots. *Industrial & Engineering Chemistry Research*, 50(15):9156–9174, 2011.
- J. Li, I.A. Karimi, and R. Srinivasan. Recipe determination and scheduling of gasoline blending operations. *AIChE Journal*, 56(2):441–465, 2010.
- 735 C.T. Maravelias. General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, 58(6):1812–1828, 2012.
- C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski, and M. Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30(6):913–946, 2006.
- 740 P. Schittekat and K. Sörensen. Or practice-supporting 3pl decisions in the automotive industry by generating diverse solutions to a large-scale location-routing problem. *Operations Research*, 57(5):1058–1067, 2009.
- H. Stadtler and C. Kilger. *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. Springer, 4th edition, 2010.
- 745 G. Van Essen, P. Van den Hof, and J.-D. Jansen. Hierarchical long-term and short-term production optimization. *SPE Journal*, 16(01):191–199, 2011.
- S. Velez, A. Sundaramoorthy, and C.T. Maravelias. Valid inequalities based on demand propagation for chemical production scheduling mip models. *AIChE Journal*, 59(3):872–887, 2013.
- 750 H.M. Wagner. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2):131–140, 1959.