Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale, Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Novel Detailed Representation of Batch Processes for Production Scheduling

Alexandros Koulouris[a]*, Georgios P. Georgiadis[a,b]

[a] International Hellenic University, Department of Food Science and Technology, Thessaloniki, Greece
[b] Intelligen Inc., Freehold, NJ, USA
* Corresponding Author: akoul@ihu.gr.

ABSTRACT

Traditional scheduling approaches often rely on simplified process representations to reduce computational complexity, failing to capture the real-world dynamics where tasks often overlap, and their timing depends on finer operational steps. To address these limitations, this paper proposes a novel process representation that breaks down production tasks into smaller, more primitive steps called operations. Unlike traditional methods, this approach provides a more granular depiction of task timing and resource dependencies. Operations can define the start or end of other tasks, utilize shared resources, and incorporate recipe constraints that mandate task sequencing. The proposed representation is utilized to develop two MILP models to address the makespan and the cycle time minimization problems. Finally, the efficiency and practical applicability of the developed models are showcased with a help of a case study from the pharmaceutical industry.

Keywords: production scheduling, process representation, mixed integer programming, makespan, cycle time

INTRODUCTION

The highly competitive industrial landscape enforces companies to strive for profit maximization. In that line, optimal production scheduling plays a significant role in ensuring the efficient utilization of the available resources and the increase of productivity. Effective schedules allow companies to improve important Key Performance Indicators (KPIs) like, customer satisfaction, mitigation of overproduction and overall profitability boost. However, the optimal production scheduling problem remains a significant challenge due to its high combinatorial complexity, arising from the dynamic and interconnected nature of modern manufacturing. More specifically, in chemical process production steps often overlap and their exact timing and duration can be influenced by dependencies between operations and shared resources usage. These production details are often overlooked in traditional process representations for production scheduling, resulting in suboptimal solutions that may not reflect real-world conditions.

A plethora of mathematical optimization models, such as mixed-integer linear programming (MILP) has been proposed to solve the optimal production scheduling problem [1]. Formulating the production scheduling problem as an MILP model requires a coherent representation of the production process. The most used representations are the State-Task-Network (STN) [2] and the Resource-Task-Network (RTN) [3]. These were initially introduced for network production environments and were later extended to include sequential and hybrid environments. Numerous extensions of the STN and the RTN have been proposed by the scientific community, that e.g. include the structural aspects of the plan [4] or consider routing limitations [5]. Recently, the General Material Task System (GMTS) representation has been proposed for systems with multiple production environments and diverse types of processes [6]. However, applying such methods in industrial practice is difficult. Simplified process representations are often used to make scheduling problems computationally feasible, but they fail to capture the complexity and flexibility inherent in production systems. Traditional approaches treat batch processing stages as rigid blocks, with fixed precedence constraints determining the order in which tasks are executed. In a real production environment, the tasks rarely follow strict sequential execution. On the contrary, they often overlap, and their exact timing is driven by operations that are highly dependent on resource availability and inter-task dependencies. This mismatch between traditional scheduling models and real-world production highlights the need for a more flexible and accurate approach.

To address these limitations, this paper introduces a novel process representation that captures the complex interactions between tasks, their resource requirements, and timing uncertainties inherent in production environments. Tasks are broken down into smaller processing steps called procedures and operations, capturing the interdependencies between tasks and their resource requirements. Operations are tied to recipe constraints that define strict sequencing rules, but timing flexibility can be introduced using dummy operations with variable durations. This approach allows the model to handle timing uncertainties and resource-sharing scenarios more effectively, making it possible to generate optimized schedules that closely reflect real-world production realities. The proposed representation is incorporated into a MILP-based scheduling framework, improving the accuracy and reliability of scheduling decisions without significantly increasing computational complexity. Two MILP models for makespan minimization and cycle time minimization are developed with their applicability and efficiency demonstrated through case study applications.

MATHEMATICAL FORMULATION

Process Representation

A batch process consists of processing tasks (*procedures*) each executed in some equipment unit. Within a procedure, more primitive processing steps (*operations*) can be recognized. For example, a fermentation procedure consists of a series of operations such as SIP, loading of materials, heating, fermenting, transferring-out the broth, CIP etc. With the exception of the main equipment unit defined at the procedure level, all other resources (materials, utilities, labor etc.) are consumed at the operation level. Relative timing of all processing tasks is also done at the operation level. This is an important aspect of the proposed formulation which differentiates it from established process representations. Moreover, it offers an intuitive representation for process engineers, as it aligns perfectly with how they conceptualize and structure their process.

In batch processing, the set of all procedures comprises the process *recipe* for the production of a stable product or intermediate. Unlike discrete manufacturing industries, recipes in processing industries are, in general, rigid structures with operations executed in sequence or in parallel with fixed timings. However, in many cases, the execution of certain operations could be delayed if the required resources are not available. For example, a CIP operation can wait up to a certain time (the dirty-hold time) if the required cleaning skid is not available. This kind of flexibility in the execution of operations is introduced in the formulation in the form of flexible shifts which denote the maximum possible delay (positive for forward scheduling, negative for backward scheduling). For consistency with the rest of the representation, flexible shifts are modelled as “dummy” operations with variable duration.

Let *M* be the set of all operations in a recipe. Each operation, *m*, in this set is given a triplet of variables for its start (s_m), duration (d_m) and end (e_m) which we call “timing elements”. Because of the way recipes are executed, all these timing elements are interrelated (e.g. the start of one operation coincides with the end of the previous). Figure 1 shows an example of the proposed representation for a fermentation procedure. A dummy “Flexible Delay” operation has been added to the list of actual operations to represent the possible delay in the execution of the CIP operation if needed. The duration of this dummy operation is considered variable.

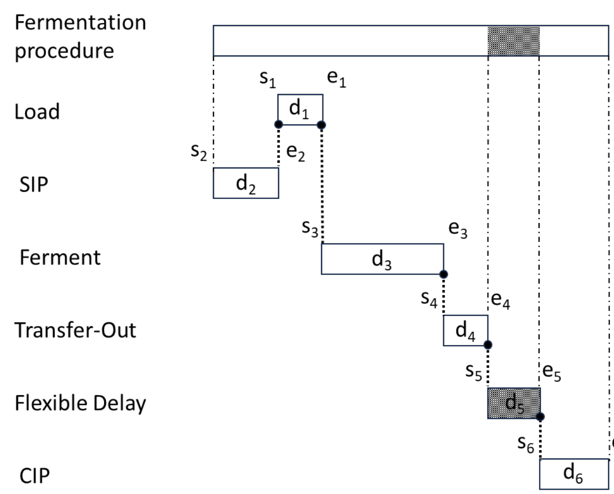

Figure 1. Representation of a fermentation procedure.

Let *V* be the set of all timing elements in the recipe ordered by dependency. Then, it is possible to construct a lower-triangular matrix, σ, whose entries represent the coefficients in the dependency equations. In other words:

$$V = \sigma \cdot V + \varphi + F \qquad (1)$$

In this equation, φ is a vector of constant values (e.g. the fixed duration of certain operations) and *F* is a vector of variable values (e.g. the duration of the dummy flexible shift operations).

The actual execution of a recipe constitutes a *batch*. When scheduling a batch, the objective is to find the appropriate timing values for the execution of all processing tasks and assign all needed resources. The set of processing tasks does not necessarily coincide with the set of operations which give rise to the vector of timing elements, V, since, in scheduling, only tasks (procedures and operations) that consume resources are of interest. However, the timing of these tasks can be determined with the help of the timing elements.

Let I be the set of all processing tasks that need to be scheduled. If a task i represents a procedure, then its start (L_i) is the earliest start time of all contained operations, its end (C_i) is the latest end time of all contained operations and its duration (Dur_i) is the difference between the two:

$$L_i \leq V_n \quad \forall i \in I, n \in S_{i,n}$$
$$C_i \geq V_n \quad \forall i \in I, n \in E_{i,n} \qquad (2)$$
$$Dur_i = C_i - L_i \quad \forall i \in I$$

where $S_{i,n}$ are the indices n of vector V that the start L_i of a task i depends upon and $E_{i,n}$ are the indices n of vector V that define the end C_i of a task i.

Let J be the set of available equipment to be used for the execution of all tasks. Each task i (procedure or operation) has a specific pool of candidate equipment to be used for its execution. For this reason, a matrix $IJ_{i,j}$ of binary values is constructed with unary values in the permissible task-equipment pairs.

This framework can be extended to introduce equipment-related durations of operations and also compatibility/connectivity constraints between all equipment. For reasons of brevity, these extensions are not discussed in this paper.

MILP formulation

The proposed novel production process representation is integrated within a MILP model to address the production scheduling optimization problem. The developed MILP model is based on the general precedence framework, which utilizes binary sequencing variables to indicate optimal task ordering and allows for a flexible and efficient approach to manage complex scheduling constraints. = It is important to note that the proposed process representation is versatile enough to be integrated with other MILP frameworks for optimal production scheduling.

Given a number of batches B of processing tasks to be processed in the available equipment J, the MILP model generates an optimal production schedule. It should be underlined that the set of processing tasks include all tasks (procedures and operations) that require the use of a resource. Two binary variable sets are employed to describe the allocation $Y_{i,b,j}$ and sequencing decisions $X_{i,b,i',b'}$. Moreover, continuous variables are introduced to model all time-related decisions like start ($L_{i,b}$) and completion ($C_{i,b}$) of tasks, value of timing elements ($V_{n,b}$) and duration of flexible operation shifts ($F_{n,b}$). All constraints of the model are presented below.

Allocation constraints

According to constraint set (3), exactly one equipment j will process each batch b of task i. The subset $IJ_{i,j}$ represents the equipment capable of processing task i.

$$\sum_{j \in IJ_{i,j}} Y_{i,b,j} = 1 \; \forall i \in I, b \in B \qquad (3)$$

Timing constraints

In constraint sets (4), the value of timing elements related to all tasks, that represent a procedure or an operation ($n \notin Flex_n$) are calculated based on their relative timings ($\sigma_{n.n'}$). More specifically, the value of an entity n in batch b equals the summation of all values of all other entities n' of batch b that entity n depends on. Respectively, constraints (5) calculate the value of timing items related to flexible operation shifts. The difference here is that the duration is not known but it is rather an unknown variable to be optimized. Constraints (6) ensure that the start time of a task will be less than the value of all relative start elements plus the start time of the batch. Similarly, the completion times are calculated in the next constraint set (7). Notice that the inequalities are required since in the case of procedures a task may be comprised by numerous operations and flexible shifts ($n \in Flex_n$). Finally, constraints (8) and (9) denote that the batch b of a task i will start and finish after the start and completion of the previous batch b-1.

$$V_{n,b} = \sum_{n'<n} \left(\sigma_{n,n'} \cdot V_{n',b}\right) \quad \forall n \notin Flex_n, b \in B \qquad (4)$$

$$V_{n,b} = \sum_{n'<n} \sigma_{n,n'} \cdot V_{n',b} + F_{n,b} \quad \forall n \in Flex_n, b \in B \qquad (5)$$

$$L_{i,b} \leq V_{n,b} + L_{i1,b} \quad \forall i \in I, b \in B, n \in S_{i,n} \qquad (6)$$

$$C_{i,b} \geq V_{n,b} + L_{i1,b} \quad \forall i \in I, b \in B, n \in E_{i.n} \qquad (7)$$

$$L_{i,b} \geq L_{i,b-1} \quad \forall i \in I, b \in B \qquad (8)$$

$$C_{i,b} \geq C_{i,b-1} \quad \forall i \in I, b \in B \qquad (9)$$

Sequencing constraints

Constraints (10) and (11) are complementary sequencing constraints that make use of a big-M parameter. In particular, constraint set (10) ensures that if a batch b of task i is processed before another batch b' of task i' ($X_{i,b,i',b'}=1$) and both of them are processed in the same equipment ($Y_{i,b,j}=Y_{i',b',j}=1$) then batch b' of task i' must start after the completion of batch b of task i. Respectively, constraint set (11) considers the case where batch b of task i is processed after another batch b' of task i' ($X_{i,b,i',b'}=0$) in the same equipment. Finally, constraints (12) guarantee that batch b' of task i will start after the completion of a batch $b<b'$ of the same task when both are processed in the same equipment.

$$L_{i',b'} \geq C_{i,b} - M \cdot (1 - X_{i,b,i',b'}) - M \cdot (2 - Y_{i,b,j} - Y_{i',b',j})$$
$$\forall i \in I, i' \in I, b \in B, b' \in B, j \in (IJ_{i,j} \cap IJ_{i',j}) : i < i' \qquad (10)$$

$$L_{i,b} \geq C_{i',b'} - M \cdot X_{i,b,i',b'} - M \cdot (2 - Y_{i,b,j} - Y_{i',b',j})$$

$\forall i \in I, i' \in I, b \in B, b' \in B, j \in (IJ_{i,j} \cap IJ_{i',j}): i < i'$ (11)

$$L_{i,b'} \geq C_{i,b} - M \cdot (2 - Y_{i,b,j} - Y_{i,b',j})$$
$\forall i \in I, b \in B, b' \in B, j \in IJ_{i,j}: b < b'$ (12)

Objective function

The goal of the mathematical model is to optimize a given objective that is meaningful to the problem under study and can improve significant production KPIs. Some typical examples are the minimization of production makespan, changeover minimization, order earliness/tardiness minimization, profit maximization and changeover minimization. In the context of this model the objective of makespan minimization C_{max} is considered.

$$C_{max} \geq C_{i,b} \qquad \forall i \in I, b = |B| \quad (13)$$

Extension for Periodic Production Scheduling

The proposed process representation is also utilized to address the cycle time minimization problem in production environments that follow a periodic scheduling approach. Cyclic, and more specifically periodic, scheduling is a very popular production scheme in many industries due to its reduced shop floor nervousness and simple implementation. However, it poses some clear disadvantages mainly related to the tight timing constraints it imposes, which typically negatively affect the production makespan.

A cyclic schedule is characterized as a schedule that is repeated following a specific timing and resource allocation pattern. Timewise this pattern is specified by the cycle time H. This means that if the first task i in batch b starts at time $L_{i,b}$, then the same task will start in the subsequent batch at $L_{i,b}+H$. In traditional cyclic scheduling, the resource allocation pattern is very limited, since each task is processed by exactly one of the available equipment throughout the schedule. As expected, this significantly reduces flexibility and resource utilization, potentially lower productivity. To address this issue the concept of periodicity has been proposed [7]. For each task i, the periodicity π_i denotes how often resource allocation decisions are repeated and is equal to the size of the task's equipment pool. For example, if the equipment pool consists of $J1$ and $J2$, then all even batches of task i {b_1, b_3, b_5, ...} are processed by J1 and all odd ones {b_2, b_4, b_6, ...} by $J2$.

The mathematical formulation developed to tackle this optimization problem is based on the presented detailed process representation and the MILP model introduced in the previous section. However, this needs to be modified to properly address the specifics of the cycle time minimization problem. This also showcases the general applicability of the proposed framework and its ability to optimally deal with a wide range of production scheduling problems.

The following constraints (3') – (7') need to be adapted to the current optimization problem:

$$\sum_{j \in IJ_{i,j}} Y_{i,b,j} = 1 \qquad \forall i \in I, b \in B: b < \pi_i \quad (3')$$

$$V_{n,b} = \sum_{n' < n} (\sigma_{n,n'} \cdot V_{n',b}) \ \forall n \notin Flex_n, b \in B: b < \pi_n \quad (4')$$

$$V_{n,b} = \sum_{n' < n} (\sigma_{n,n'} \cdot V_{n',b}) + F_{n,b} \qquad \forall n \in Flex_n, b \in B: b < \pi_n \quad (5')$$

$$L_{i,b} \leq V_{n,b} + (b-1) \cdot H \quad \forall i \in I, b \in B, n \in S_{i,n} \quad (6')$$

$$C_{i,b,n} \geq V_{n,b} + (b-1) \cdot H \quad \forall i \in I, b \in B, n \in E_{i.n} \quad (7')$$

Notice that in this case only the batches specified by the periodicity parameter are considered since the allocation decisions are repeated in the cyclic schedule. Since a cyclic scheduling approach is considered, constraints (4) – (7) need to be updated so that the required timing pattern is imposed. More specifically, constraint sets (4') and (5') are only considered for a subset of the total number of batches, which is defined by the periodicity parameter. Notice that the periodicity of a timing element π_n is equal to the periodicity of the associated task π_i. Constraints (6') ensure that the start time of a task will be less than the value of all relative start elements plus the required cycle time. Similarly, the completion times are calculated in the constraint set (7').

Cyclic heuristic constraints

The constraints described in this subsection are responsible for generating a cyclic rather than a typical production schedule. They ensure a periodic pattern both regarding the timing and the allocation decisions, that is specified by the cycle time and the periodicity parameter.

$$L_{i,b} + H = L_{i,b+1} \forall i \in I: i = 1, b \in B: b < |B| \quad (14)$$

$$Y_{i,b+\pi_i,j} = Y_{i,b,j} \forall i \in I, b \in B, j \in J: b \leq (|B| - \pi_i) \quad (15)$$

$$V_{n,b+\pi_n} = V_{n,b} \forall n \in N, b \in B: b \leq (|B| - \pi_n) \quad (16)$$

$$F_{n,b+\pi_n} = F_{n,b} \forall n \in N, b \in B: b \leq (|B| - \pi_n) \quad (17)$$

$$L_{i,b+\pi_i} \geq C_{i,b} \forall i \in I, b \in B: b \leq (|B| - \pi_i) \quad (18)$$

Constraint set (14) guarantees that the first task i of batch $b+1$ must start at exactly H time units after the start of the same task in the previous batch b. It's important to emphasize that this constraint is only valid for the first task of each batch, thus increasing the timing flexibility for the rest of the tasks. The following constraint sets (15) – (18) are tightening constraints that take advantage of the schedule's cyclic nature to improve the model's computational efficiency. In particular, constraints (15) impose that the equipment allocation decisions are repeated for every π_i batches. For example, let us assume that task $I1$ of batch $B1$ is processed in equipment $J1$, and the periodicity of $I1$ is 3, then constraint (15) will guarantee that batch $B4$ of $I1$ will be processed in equipment $J1$. Similarly, constraints (16) and (17) impose the timing constraints of the schedule's cyclic nature utilizing the periodicity parameter associated with each entity π_n. Finally, constraint set (18) denotes that task *i* of batch $b+\pi_i$ will start after task *i* of batch *b*.

Cyclic scheduling objective

The goal of the mathematical model is to minimize the production cycle time *H*, which is beneficial for key production metrics, like decreased dead-times and increased throughput.

$$\min H \tag{19}$$

In summary the mathematical model for the minimization of the cycle time consists of constraints (3') – (7'), (10) – (12), (14) – (18) and the objective function (19).

CASE STUDY

One case study from the pharmaceutical industry is presented here to illustrate the applicability of the proposed process representation. The case study is solved both for the makespan minimization problem as well as for cyclic time minimization. The case study uses a recipe representing a single step from a multi-step organic synthesis of a small API molecule. The recipe consists of eleven procedures, each requiring several operations. The synthesis steps are executed in two reactors (R-101 and R-102) which utilize two holding tanks (T-101 and T-102) for solution preparation and material holding after each reactor process. A filtration and a drying task complete the recipe. Figure 2 shows a graphical depiction of the reaction and holding procedures and operations with their scheduling links. Blue arrows indicate internal (within a procedure) links while red arrows indicate links across procedures. As it can be seen, there exist complex operation interrelations with various scheduling links.

The reactors execute dedicated tasks; the tanks, however, can be used alternately for any holding task depending on their availability. The transfer of material into a tank from a reactor is considered inflexible but the transferring of material out of the tank to some reactor has a flexible start of 12 hours maximum. In other words, the material can be held in the tank until the destination reactor for the material becomes available.

Figure 3a shows, in the form of a Gantt chart, the optimal schedule of six batches of this process using the makespan minimization objective. For brevity, only the reactor and tank procedures are shown. The batches are identified by different colors in the chart. As seen, the optimal solution makes full utilization of reactor R-101 by inserting tasks of subsequent batch within the gaps between the tasks of each executed batch. This is made possible by the use of flexible shifts in the tank operations that delay the transfer of material to the reactors

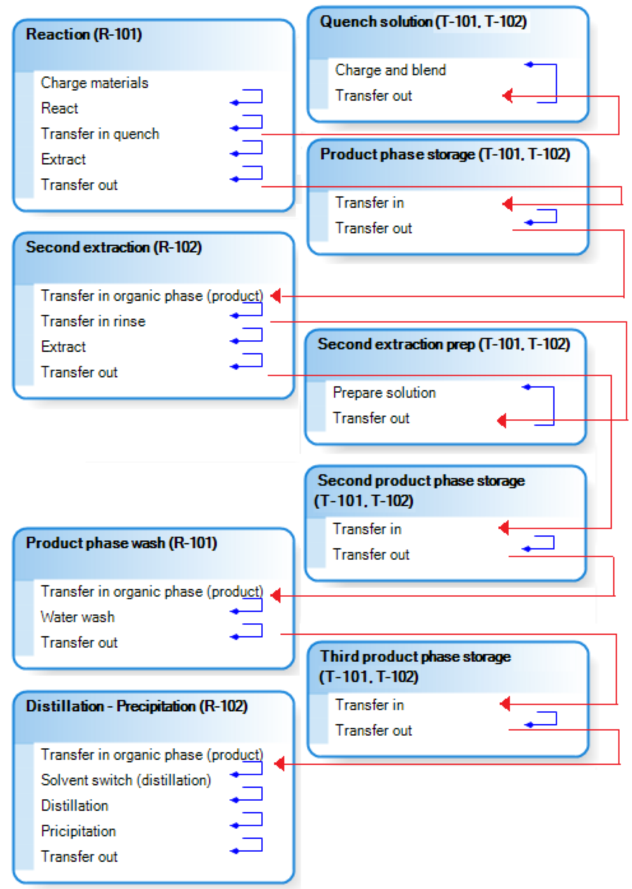

Figure 2. Graphical depiction of recipe.

until the latter become available. These flexible shifts are indicated in the charts by the gaps of idle time seen in the operations executed in T-101 and T-102. The optimal make span is calculated at 159 hours.

In the second case shown in Figure 3b, the obtained schedule minimizes the cycle time. In this case, all tasks are executed periodically and all batches are identical. The optimization still makes use of flexible shifts to minimize the cycle time but the shifts have the same value in all batches in order to preserve periodicity. The calculated minimum cycle time is 20.5 hours, but at the expense of increased makespan which is now 171 hours. In an industrial setting, however, a periodic schedule is sometimes favored due to its simplicity in execution.

The SCIP open-source solver [8] was used for the solution of the optimization problems. The recipe representation, formulation of the mathematical optimization problem and visualization of the results were executed in the scheduling software SchedulePro (by Intelligen, Inc.).

CONCLUSION

In this paper, a novel detailed representation capable of capturing all timing aspects of a production

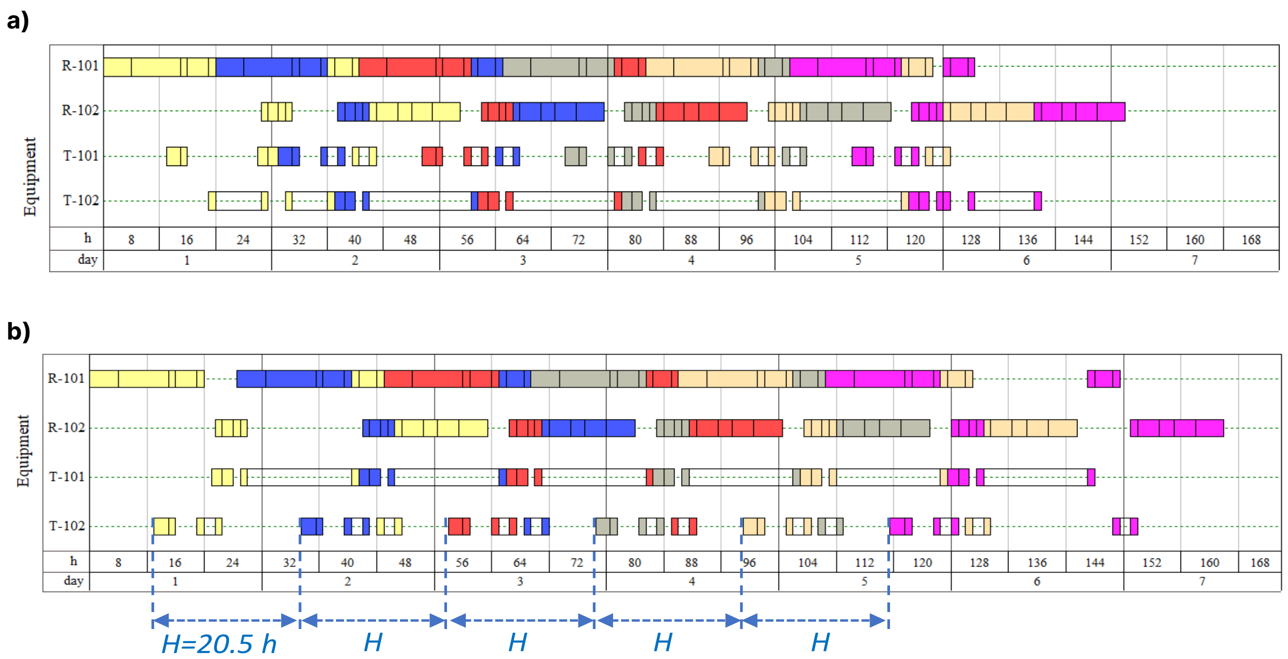

Figure 3: Gantt charts for a) makespan minimization and b) cycle time minimization

process is presented. The proposed approach ensures that optimal scheduling decisions even for very detailed recipes can be generated. Based on this representation, two MILP models have been developed - one focusing on makespan minimization and the other on cycle time minimization. The applicability and efficiency of these models are demonstrated through a realistic and complex case study, where optimal schedules are promptly generated. The results highlight the ability of the representation in handling complex production processes facilitating the implementation of optimization-based solutions into real industrial environments, thus reducing the gap between academic research and industrial practice.

REFERENCES

1. Georgiadis GP, Elekidis AP, Georgiadis MC. Optimization-based scheduling for the process industries: from theory to real-life industrial applications. *Processes* 7:438 (2019) https://doi.org/10.3390/pr7070438
2. Kondili E, Pantelides CC, Sargent RWH. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. *Comput. Chem. Eng.* 17 (2):211-277 (1993) https://doi.org/10.1016/0098-1354(93)80015-F
3. Pantelides, CC. Unified frameworks for optimal process planning and scheduling. *In: Proceedings of the Second Conference on Foundations of Computer Aided Operations. CACHE Publications* 253-274 (1994)
4. Barbosa-Povoa AP, Macchietto S. Detailed design of multipurpose batch plants. *Comput. Chem. Eng.* 18 (11-12):1013-1042 (1994) https://doi.org/10.1016/0098-1354(94)E0015-F
5. Zyngier D, Kelly JD. UOPSS: a new paradigm for modeling production planning & scheduling systems. *In: Proceedings of the Symposium on Computer Aided Process Engineering:* 20 (2012)
6. Samadi A, Maravelias CT. A comprehensive chemical production scheduling representation. *Comput. Chem. Eng.* 181: 108552 (2024) https://doi.org/10.1016/j.compchemeng.2023.108552.
7. Georgiadis GP, Koulouris A. A MILP Model for the Minimization of Cycle Time in Periodic Production Scheduling using Flexible Operation Shifts. *Comput. Aided Chem. Eng.* 53:1651-1656 (2024) https://doi.org/10.1016/B978-0-443-28824-1.50276-3
8. Bestuzheva K, Besançon M, et al. The SCIP Optimization Suite 8.0. arXiv preprint arXiv:2112.08872 (2021)