CHEMICAL ENGINEERING TRANSACTIONS

VOL. 74, 2019

A publication of AIDIC

The Italian Association of Chemical Engineering
Online at www.cetjournal.it

Guest Editors: Sauro Pierucci, Jiří Jaromír Klemeš, Laura Piazza
Copyright © 2019, AIDIC Servizi S.r.l.
ISBN 978-88-95608-71-6; **ISSN** 2283-9216

DOI: 10.3303/CET1974103

An Effective MILP-Based Decomposition Algorithm for the Scheduling and Redesign of Flexible Job-Shop Plants

Natalia P. Basán[a], Mariana E. Cóccola[a], Alejandro García del Valle[b], Carlos A. Méndez[a,*]

[a]INTEC (UNL –CONICET), Güemes 3450, Santa Fe, 3000, Argentina
[b]University of A Coruña, C/ Mendizábal s/n, Ferrol, 15403, Spain
cmendez@intec.unl.edu.ar

This paper presents a decomposition algorithm for the integrated scheduling and redesign problem of a multi-stage batch plant dealing with multipurpose units and heterogeneous recipes. First, the procedure solves the scheduling problem considering the existing plant configuration with the main goal of minimizing the makespan. Then, a second objective of minimizing the number of units utilized without worsen the makespan achieved in the first stage is considered. The units released can be reallocated to other compatible processing stages in order to minimize the initial makespan value. In order to tackle large industrial examples, both scheduling and redesign problems are solved through a decomposition algorithm, which has a MILP model as its core. The procedure is tested on several realistic instances, demonstrating its robustness and applicability.

1. Introduction

In the context of a complex decision-making process in industry, scheduling is defined as the problem of planning the manufacturing processes by allocating a set of operations to the available resources over a given time horizon taking into account some production targets (Castro et al., 2018; Pinedo, 2016). In a company, a good production program allows increasing the productivity of the whole manufacturing process. The main goal is to maximize the plan efficiency, minimizing costs or makespan (Harjunkoski et al., 2014; Méndez et al., 2006). In the last two decades, optimization techniques based on exact mathematical programming methods have played an important role in the decision-making processes. From the literature, it follows that two types of scheduling problems may be defined according to the shop environment, the number of processing stages, the operational constraints, and the complexity of the production route: (i) flow shop and (ii) job shop, besides of their respective variants. All these scheduling problems turn out to be NP-hard (Garey et al., 1976; Guo and You, 2018; Hegyháti, 2018).

Particularly, the job shop environment is related to multipurpose plants. According to Kopanos and Puigjaner (2019), in multipurpose plants, the products are manufactured via different processing networks and some units may be used to perform non-consecutive operations for the same product. Multipurpose plants result in more flexible operation, which can be optimized to decrease equipment idle time and to more efficiently utilize critical equipment units. The scheduling problem in flexible manufacturing plants, commonly known as Flexible Job Shop scheduling Problem (FJSP), considers that an operation (the processing of a job at one stage) may be carried out on a set of compatible production units. According to Shen et al. (2018) the availability of alternative units performing similar operations may increase the system performance and help to manage preventive maintenance or tackle breakdown and other unforeseen events.

Generally, the flexible manufacturing systems involve the processing of a variety of products, using different processing sequences. Moreover, mixing operations (or assembly operations for assembly lines) of intermediate products may also be required to obtain the final products. Hence, this paper proposes a MILP model based on the general precedence notion to solve to optimality the integrated scheduling and redesign problem of multi-stage plants involving multipurpose units and mixing operations. Moreover, the MILP model is then embedded within a decomposition algorithm in order to efficiently address large-scale scheduling

Paper Received: 1 July 2018; Revised: 3 October 2018; Accepted: 7 March 2019

problems arising in flexible manufacturing plants. The main goal is to minimize the makespan while satisfying all process constraints. First, the procedure solves the scheduling problem considering the full set of units. Then, the algorithm runs again but in this case for determining if the plant is oversized through the minimization of the number of units to be utilized without exceeding the makespan found in the first step. In case of unoccupied units, reassignments to other groups of cells or for other operations can be analysed.
The robustness and applicability of the proposed MILP-decomposition algorithm is demonstrated by solving several instances derived from two complex examples.

2. Problem statement

In the FSJP problem, a multipurpose unit may handle operations of several processing stages. Therefore, units performing similar functions are grouped together in a work cell or workstation u. Figure 1 illustrates an assembly scheduling problem with multipurpose units, in which a set of nine products should be processed on six stages.

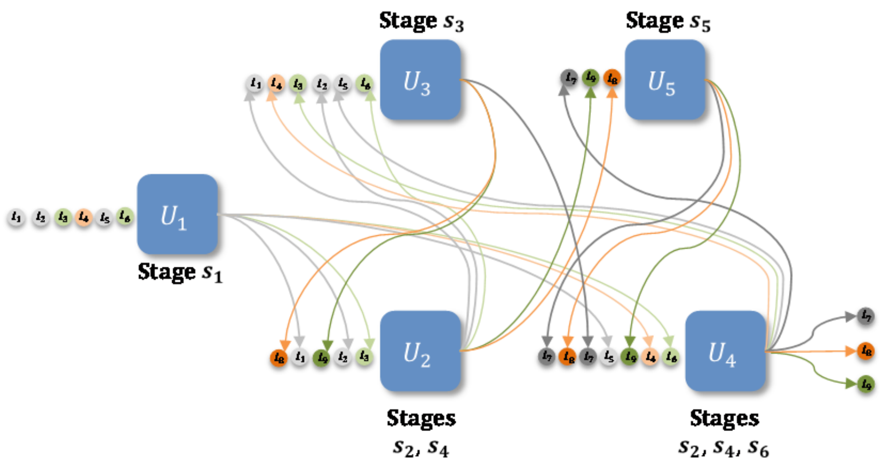

Figure 1: Job shop scheduling problem with multipurpose units

The FSJP problem is mathematically represented as a set of products $i \in I$, which should be processed through a predefined sequence of processing stages $s \in S_i$, not necessarily using all stages. The available units $K = \{k_1, k_2, .., |K|\}$ are allocated in cell groups or workstations $u \in U$ according to their similar or identical functions: $k \in K_u$. Additionally, the problem representation defines that a subset of processing stages $s \in S_u$ is performed in each u. In case that a final product $i \in I^{fin}$ is composed by intermediate products $i' \in I^{int}$, the subset SA_i determines which products integrate i. The mixing (or assembly) operations are executed at processing stages $s \in S^{\#}$.
A specific unit $k \in K$ is able to process just one product at a time and can be categorized as either single purpose or multipurpose. A single-purpose unit may be assigned to a single processing stage while multipurpose units can handle operations belonging to several stages. For treating with multipurpose units, the problem definition incorporates a set of tasks $T = \{t_1, t_2, \dots, t_t\}$, where each task $t \in T$ references the processing of one product *i* at one stage *s*. Consequently, the processing of each product $i \in I$ on the manufacturing line is decomposed in a set of tasks $t \in JT_i$, where $JT_i \subset T$. In addition, the subset $ST_s \subset T$ includes all operations to be performed at stage s. Note that for each product i to be processed at any stage $s \in S_i$, there will be a task $t \in (JT_i \cap ST_s)$.
The problem assumptions determines that: (i) the processing time of product i at stage s, pt_{is}, is known in all cases; (ii) transportation times between stages are considered negligible or included in the processing times; (iii) all parameters are deterministic; (iv) there are no setup or changeover times; (v) either UIS (unlimited intermediate storage) or NIS (non-intermediate storage) transfer policy between stages can be adopted.
The first problem goal is to find the optimal plant operation that minimizes the makespan (MK). Once the optimal *MK* is determined and fixed, the problem incorporates a second objective function for minimizing the number of units to be utilized at each workstation $u \in U$. Thus, the units resulting unoccupied may be assigned to other feasible workstations to further minimize the makespan value.

3. Mathematical formulation

The FSJP problem is formulated as a Mixed Integer-Linear Programming (MILP) model based on the general precedence concept presented by Méndez et al. (2000). The proposal of these authors is based on continuous time and was originally applied to batch scheduling problems with dedicated units. Hence, this formulation is extended in order to incorporate multipurpose units and assembly operations. The main objective is to determine the optimal operation of the plant in order to minimize the makespan and improve the use of the processing units considering all the operational constraints of the process. Therefore, the mathematical model allows to determine: (i) allocation and sequencing of the tasks in each machine or unit according to the processing recipes of each product, (ii) the start and end times of each operation, (iii) the minimum number of processing units needed to satisfy the best makespan, and (iv) relocation of feasible processing units to increase the efficiency of the process. The constraints composing the mathematical model follow.

$$minimize\ MK \tag{1}$$

$$\sum_{k\in K_s} Y_{tk} = 1 \qquad \forall\ s \in S, t \in ST_s \tag{2}$$

$$Tf_t \geq Ts_t + pt_t \qquad \forall\ t\ \in T \tag{3}$$

$$Ts_t \geq Tf_{t'} \qquad \forall\ i\ \in I, s \in S, t\ \in\ JT_i, t' \in\ JT_i, t\ \in\ ST_s\ t' \in\ JT_{s-1}: s > 1 \tag{4}$$

$$Ts_{t'} \geq Tf_t - M(1 - W_{tt'}) - M(2 - Y_{tk} - Y_{t'k}) \quad \forall\ s\ \in S, s' \in S, t\ \in ST_s, t' \in\ ST_{s'},\ k\ \in\ (K_s \cap\ K_{s'}): t < t \tag{5}$$

$$Ts_t \geq Tf_{t'} - MW_{tt'} - M(2 - Y_{tk} - Y_{t'k}) \qquad \forall\ s\ \in S, s' \in S, t\ \in ST_s, t' \in ST_{s'}, k\ \in\ (K_s \cap\ K_{s'}):\ t < t' \tag{6}$$

$$Ts_t \geq Tf_{t'} \qquad \forall s \in S^{\#}, i\ \in I, i' \in SA_i, t\ \in\ JT_i, t' \in\ JT_{i'}\ t\ \in\ ST_s\ t' \in\ JT_{s-1} \tag{7}$$

$$MK \geq Tf_t \qquad \forall\ t\ \in T \tag{8}$$

The objective function is represented by Eq. (1). The main goal is the minimization of the makespan. Eq. (2) is the assignment constraint defined for every processing stage s and task $t \in ST_s$. Binary variable Y_{tk} values 1 if operation t is performed in unit k; otherwise, it is set to zero. Eq. (3) computes the ending time of task t, $Tf_t,$ as its begin time Ts_t plus the processing time of $t,$ which is known a priori through parameter pt_t. Note that $pt_t = pt_{is}$ when $t\ \in (JT_i \cap\ ST_s)$. Variables Tf_t and Ts_t are defined as continuous positive in the mathematical model. Eq. (4) establishes that the processing of product i on stage s, denoted by task $t \in (JT_i \cap ST_s)$, should not start until its processing in the previous stage $(s-1)$, denoted with task $t' \in (JT_i \cap ST_{(s-1)})$, has been completed. The constraint (4) should be modified according to the storage policy adopted; for UIS policy, Eq. (4) should be expressed as inequality while for NIS policy, the constraint becomes in equality. On the other hand, Eqs. (5) and (6) are used to sequence any pair of tasks assigned to a same processing unit. Remember that a multipurpose unit k may process tasks belonging to different stages s, always that $k \in K_s$. Following the general precedence concept, the sequencing constraints on a unit k are defined for any pair of tasks (t, t') where $t < t',\ t\ \in ST_s,\ t' \in ST_{s'}$ and $k\ \in\ (K_s \cap\ K_{s'})$. The binary sequencing variable $W_{tt'}$ takes 1 as value when t is processed before than t'; otherwise, it is set to zero. If two tasks (t, t') are assigned to a unit k $(Y_{tk} + Y_{t'k} = 2)$ and task t is chosen to be processed before task $t',$ i.e. $W_{tt'} = 1,$ then Eq. (5) forces that the begin time of t', $Ts_{t'}$, will be greater than the ending time of task t, Tf_t. But, if task t' is processing earlier than $t,$ i.e. $W_{tt'} = 0,$ the reverse statement holds true and Eq. (6) becomes active. The parameter M is an upper bound for timing variables. Eq. (7) forces that the processing of product i in a mixing stage $s \in S^{\#}$ cannot begin until its intermediate products $i' \in SA_i$ have completed their processing in the previous stage $(s-1)$. Finally, the minimum completion time of all tasks, given by continues positive variable MK, is computed by Eq. (8).

Once founding the optimal solution for the scheduling problem, the minimization of the number of units that being used is chosen as a new objective. After solving the MILP model (1)-(8), the variable MK is bounded by its optimal value (fixing it as upper bound) and a new model is generated considering two additional constraints (9)-(10). In this instance, the objective function (1) is replaced for Eq. (9). V_k is a continuous positive variable that, according to Eq. (10), is set to zero only when unit k is not utilized in any stage. In this case, we say that unit k is unoccupied.

$$minimize\ \sum_{k \in K} V_k \qquad (9)$$

$$Y_{tk} \leq V_k \qquad \forall\, t \in T, k \in K \qquad (10)$$

4. The MILP-based decomposition strategy

Due to the integrated scheduling and redesign problem of flexible manufacturing plants is strongly combinatorial and NP-hard (Pinedo, 2016), the full space approach presented above may not be suitable for solving real-world industrial problems. To take advantage of the robustness offered by the MILP model, it was embedded within a decomposition algorithm, which iteratively solves the mathematical model but considering a reduced space search, maintaining the number of assignment and sequencing decisions at a reasonable level at each solver execution. Although the algorithm does not guarantee the convergence to the optimal solution of the problem, it is capable to provide high quality solutions, sometimes the optimal one, with relative computational effort even for large-size instances. The procedure is based on the general decomposition technique presented by Kopanos et al. (2010), which is extended in order to deal with multipurpose units, mixing operations and redesign constraints. As shown Figure 2, the algorithm is composed of three stages: (i) construction stage, (ii) improvement stage, and (iii) redesign stage.

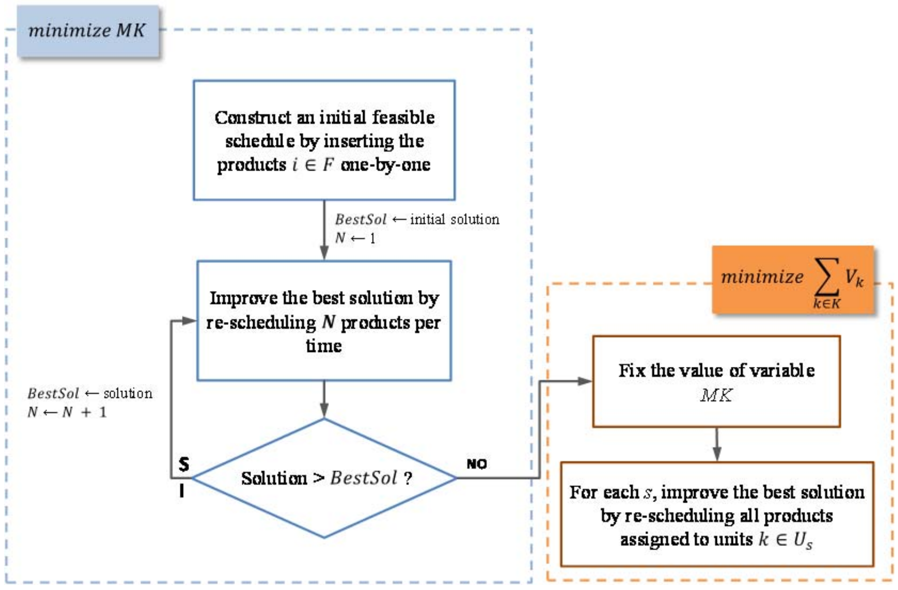
$minimize\ MK$
Construct an initial feasible schedule by inserting the products $i \in F$ one-by-one
$BestSol \leftarrow$ initial solution
$N \leftarrow 1$
Improve the best solution by re-scheduling N products per time
$BestSol \leftarrow$ solution
$N \leftarrow N + 1$
SI
Solution > $BestSol$?
NO
$minimize \sum_{k \in K} V_k$
Fix the value of variable MK
For each s, improve the best solution by re-scheduling all products assigned to units $k \in U_s$

Figure 2: General structure of the MILP-based algorithm

4.1 Constructing an initial feasible solution

The first stage of the iterative procedure aims at finding an initial feasible schedule by considering the full set of processing units. In order to minimize the MIP solver search space, this paper proposes to insert the products $i \in I^{fin}$ one-by-one and to select them following the lexicographic order. When a product $i \in I^{fin}$ is selected to be scheduled, its subassemblies $i' \in SA_i$ are also scheduled. In the constructive stage, the MILP model (1)-(8) should assign and sequence all tasks $t \in (JT_i \cap JT_{i'})$ for reaching a feasible solution for the problem. A boolean parameter called $active_t$ is used by the algorithm to determine if task t is scheduled at iteration $iter$. After each resolution of the MILP model, binary variables Y_{tk} for tasks t with $active_t = true$ are fixed. At end, the constructive step solution is saved in parameters sMK, sY_{tk}, and sW_{tk}.

4.2 Improving the initial solution

In this second algorithmic step, the initial scheduling solution given by the constructive stage is improved by performing several rescheduling iterations. A rescheduling action consists of releasing a subset of products $i \in I^{fin}$ from the current schedule in order to find better unit assignments or sequencing for them. Note that when a product *i* is selected, it means that all tasks $t \in JT_i$ may be rescheduled. Moreover, tasks $t \in JT_{i'}$, where $i' \in SA_i$ may be also reassignment or reordered. Once all products $i \in I^{fin}$ and their sub-products have been rescheduled, the procedure checks if the makespan has been improved. When true, the algorithm starts the rescheduled iterations again from the beginning. Otherwise, the procedure terminates and reports the current makespan as the best solution found. A key point to take into account in the improvement stage is the number of products $i \in I^{fin}$ to be rescheduled at each model execution, given by parameter N. The fewer the value of N, the higher the number of iterations and smaller the solver search space, resulting in manageable model sizes. However, as the value of N increases, the number of iterations is reduced but the feasible region becomes larger, and hence, the resulting mathematical models become more complex and intractable.

4.3 Incorporating the redesign decisions

The last stage of the algorithm evaluates the redesign of the plant, determining the units that can be released if the production plant is oversized. Hence, this step aims at minimizing the number of processing units required to process all products within the makespan given by the improvement stage. Once founding the best solution for the scheduling problem, the procedure fixes the value of variable MK and several rescheduling actions are iteratively applied on the current scheduling, solving the MILP model (2)-(11) for each workstation $u \in U$. The procedure sets the value of parameter $active_t$ in true for all tasks $t \in ST_s$, where $s \in S_u$, with the goal of reallocating such operations in the smallest number of units $k \in K_u$. The procedure ends when all workstations $u \in U$ have been iterated.

5. Computational results

The MILP-based decomposition algorithm developed in this paper may be utilized for solving any multi-stage batch scheduling problem arising in flexible job shop environments, such as pharmaceutical industries, aerospace industries, automotive industries, printing industries, between others. In this work, the performance of the proposed solution strategy is tested by solving two complex instances.

5.1 Example 1

The first example deals with the scheduling problem of a company that manufactures a product (mould) composing of 5 different sub-products. The production line has 9 processing stages and 16 units (2 multipurpose). Three examples involving the production of 4, 6, and 8 final products, respectively, are considered. The computational statistics and objective values for each instance are summarized in Table 5.

Table 1: Computational statistics for the three instances of case study 1

Final product	MILP model						Iterative algorithm		
	Cont. var.	Bin. var.	Eqs.	Obj. func.	CPU (s)	Gap %	Initial solution	Best solution	Total CPU (s)
4	194	2332	4685	**979**	127.7	0	1266	**979**	28.8
6	290	5166	10363	**1355**	3600	30.5	1633	**1355**	455.7
8	386	9112	18265	**1772**	3600	55.4	2030	**1764**	1145

Note that for the first problem instance, the decomposition strategy reports the optimal solution in less time than the full space approach. For the case of producing 8 products, a quantity of 192 tasks must be assigned and sequenced on 16 processing units. In spite of the complexity of the third problem instance, the decomposition approach shows a good computational performance, reporting a high quality solution with a modest CPU time.

5.2 Example 2

This example involves the manufacture of 25 products, each one formed by 2 sub-products. The production process comprises 42 equipment units and 7 processing stages, where 13 units are multipurpose and 2 stages perform mixing operations. Note that a total of 225 tasks have to be assigned and sequenced in 42 processing units in order to find a feasible schedule. Such example results into a huge MILP model of 147951 equations, 2550 binary variables and 7202 continuous variables. As consequence, the full space approach reaches a makespan of 238 days, with a gap of 24.2%, after the predefined CPU time limit of an hour. Instead, the decomposition algorithm improves such solution by 3.5%, featuring a makespan of 230.3 days in just 652.3 seconds of CPU time.
The best solution for the scheduling problem is depicted in Figure 3a. Taking this schedule as basis, the redesign stage reduces the number of units utilized from 42 to 35 without changing the makespan (see Figure 3a). If the company decides to relocate the unoccupied units to the workstation representing the bottleneck, the makespan can be reduced up to 196.1 days, as shown Figure 3b.

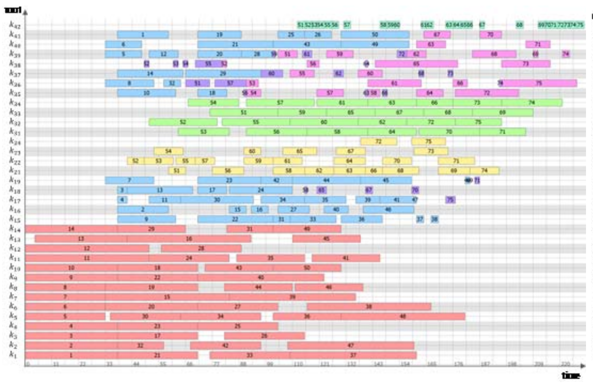
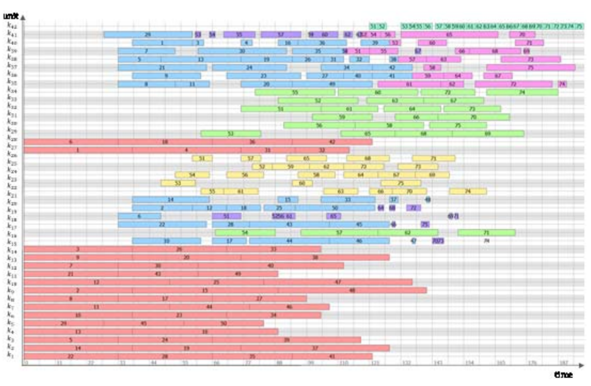

Figure 3a: Best solution reported by the redesign stage - $MK = 230.3$ *days*
Figure 3b: Best solution reported after relocating some units - $MK = 196.1$ *days*

6. Conclusions

This paper has addressed the integrated scheduling and redesign problem of flexible manufacturing plants dealing with multipurpose units. First, a rigorous mathematical model relied on the general precedence concept was developed for solving the problem under study. In order to extent the use of the MILP model to real-world applications, it was embedded within a three-stage decomposition algorithm, which solves repeatedly the rigorous model but considering a reduced search space at each solver execution. The computational results demonstrated that the algorithm converges efficiently to the optimal solution with low computational efforts when small to medium sized instances are considered. For large-scale problems, the iterative procedure reports better solutions than the rigorous mathematical model, also outperforming the exact optimization approach in terms of CPU time.

Acknowledgments

The authors gratefully acknowledge the financial support from CONICET under Grant PIP 112 20150100641 and from “Universidad Nacional del Litoral” under Grant CAI+D 2016-UNL / PIC 50420150100101LI.

References

Castro, P.M., Grossmann, I.E., Zhang, Q., 2018. Expanding scope and computational challenges in process scheduling. Comput. Chem. Eng. 114, 14–42. https://doi.org/10.1016/j.compchemeng.2018.01.020

Garey, M.R., Johnson, D.S., Sethi, R., 1976. The Complexity of Flowshop and Jobshop Scheduling. Math. Oper. Res. 1, 117–129.

Guo, C., You, F., 2018. Robust optimization for batch process scheduling under uncertainty using piecewise linear decision rule. Chem. Eng. Trans. 70, 1711–1716.

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. 62, 161–193.

Hegyháti, M., 2018. Batch process scheduling with eS-graph: A case study. Chem. Eng. Trans. 70, 115–120.

Kopanos, G.M., Méndez, C.A., Puigjaner, L., 2010. MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. Eur. J. Oper. Res. 207, 644–655.

Kopanos G.M., Puigjaner L., 2019, Solving Large-Scale Production Scheduling and Planning in the Process Industries, Springer, Switzerland.

Méndez, C.., Henning, G.., Cerdá, J., 2000. Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. Comput. Chem. Eng. 24, 2223–2245.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913–946.

Pinedo, M.L., 2016. Scheduling: Theory, Algorithms, and Systems, Fifth Edit. ed, Scheduling: Theory, Algorithms, and Systems. Springer, New York.

Shen, L., Dauzère-Pérès, S., Neufeld, J.S., 2018. Solving the flexible job shop scheduling problem with sequence-dependent setup times. Eur. J. Oper. Res. 265, 503–516.