Author's accepted version.
The final authenticated version is available online at https://doi.org/10.1007/978-3-030-76307-7_13
D. A. Rossit et al. (Eds.): ICPR-Americas 2020, CCIS 1407, pp. 171–176, Springer 2021.

# Efficient Scheduling of a Real Case Study of the Pharmaceutical Industry using a Mathematical-Algorithmic Decomposition Methodology

Josías A. Stürtz[1] and Pablo A. Marchetti[1,2]

[1] Universidad Tecnológica Nacional, Facultad Regional Santa Fe
Lavaisse 610, 3000 Santa Fe, Argentina
[2] Instituto de Desarrollo Tecnológico para la Industria Química (INTEC)
UNL-CONICET, Güemes 3450, 3000 Santa Fe, Argentina
pmarchet@intec.unl.edu.ar

**Abstract.** This work presents a mathematical-algorithmic methodology to handle scheduling decisions on multiproduct multistage batch facilities, developed with the aim to solve problems of industrial size. The proposal is based on the iterative construction of a solution by solving a sequence of subproblems associated to each stage, identifying and fixing the critical decisions at each step, and keeping rigorous information on the bounds. The methodology is applied to a real case study obtaining good quality solutions in competitive computing times.

**Keywords:** Multiproduct batch plants, Scheduling, Optimization

## 1 Introduction

In the last 30-40 years, scheduling decision tools have become a critical asset for the success and competitiveness of manufacturing organizations. Scheduling problems have been widely studied, and different mathematical formulations have been proposed that can be broadly classified into two groups: discrete and continuous time representations [1]. When comparing different models the quality of the solution and time required for the computation are critical. In general, a tighter formulation obtains better solutions and more competitive performances by reducing the number of alternatives to be explored by the solver, usually a branch-and-cut algorithm.

Despite countless efforts in formulating and solving scheduling problems, from both researchers in academia and engineers in industry, there is still a significant gap between theory and practice. One of the main causes of this gap is the intrinsic difficulty of solving industrial-scale problems, due to the underlying combinatorial explosion of alternatives. Harjunkoski et al. [2] present an extensive review of the strengths and weaknesses of different scheduling approaches, focusing on the development and deployment of industrial applications. They argue that current optimization tools can be successfully applied to increase plant

floor production efficiency, although they note there is still a large potential for improvement.

This work presents a scheduling methodology for multiproduct multistage batch plants considering parallel equipment, unlimited storage, and sequence-dependent transition times, which is applied to an industrial-scale case study from the pharmaceutical industry [3,4]. The approach is based on the idea of prioritizing the tasks that require the critical resources of the installation, and subsequently accommodating the tasks that have more slack time available. In particular, the existence of a critical stage or "bottleneck" is assumed.

## 2 Mathematical-Algorithmic Proposed Methodology

The proposed method consists of an iterative algorithm that solves a sequence of mixed-integer linear programming (MILP) mathematical models. Each of these models corresponds to a subproblem of the complete problem being tackled, and considers one of the stages together with a subset of the preceding and succeeding tasks required at the associated upstream and downstream stages, respectively, of the process. To model the subproblems a new version of the continuous time representation based on time-slots by Pinto and Grossmann [5] has been developed, in which it is possible to add a "wildcard" slot to a given unit. That is, a multi-valued slot to which an arbitrary number of batches can be assigned.

The algorithm is structured with a main iteration and an inner iteration. The main iteration, which considers all the stages of the process, gradually builds a solution of the complete problem by fixing one by one the solutions of each stage. On each main iteration, the set of pending (or not fixed) stages is analyzed, solving a sequence of subproblems (inner iteration) for each stage. The aim is to identify the critical stage, that is, the one whose solution produces the greatest deterioration to the value of the objective function (makespan). Once this stage is identified, the allocation and sequencing decisions obtained in the corresponding subproblem are fixed, and the set of pending stages is reduced. The main iteration ends when all the stages have been fixed and a complete solution of the problem (schedule) is obtained.

On the other hand, the inner iteration is focused on solving the subproblems corresponding to each stage. In this iteration, an approximate model and an exact model are alternately solved. The approximate subproblem, which includes multi-valued slots in all units, is used to obtain a candidate slot configuration for the stage. Subsequently, the exact subproblem fixes the previously obtained slot configuration to find a feasible solution for the stage. Based on the quality of this solution and the parameters of the algorithm it is decided to end the inner iteration or repeat the process. If it continues, the recently analyzed slot configuration is removed from the following approximate subproblems by adding appropriate integer cuts. The inner iteration returns the best feasible solution found and accurately reports the bounds (MIP solution and best possible solution) to the main iteration.

The parameters taken into account for the inner iteration on a given stage are the following: (a) the number of slots $R$ to be included on the equipment units of the preceding (upstream) and succeeding (downstream) stages of the process, (b) a time limit $T$ to solve each subproblem, (c) a time limit for the inner iteration process, and (d) the maximum number of iterations allowed. The inner iteration concludes after a solution is obtained for an exact subproblem such that its best possible solution cannot be improved, or when the limits set in (c) or (d) are exceeded.

## 3 Results and Discussion

The proposed method has been applied to a real case study of the pharmaceutical industry (see [4]) comprising a multiproduct batch installation with 17 processing units running in parallel and distributed in 6 stages. One of the main complexities of the problem is the presence of sequence-dependent transition times, which are larger than the processing times in some stages. Based on the 30 batches problem, instances including the first 10, 15, 20, 25 and 30 batches have been evaluated. Different alternatives were considered for the parameters (a) with values $R = 2$ and $R = 3$, and (b) with values $T = 300$, 600, 1200 s. Besides, a maximum of 1 h and a limit of 10 cycles were set for the parameters (c) and (d) of the inner iteration.

The proposed methodology has been implemented in GAMS, using the MILP solver GUROBI 7.5 for the inner iteration subproblems. In order to compare the computational performance and the quality of the solutions obtained (objective function value and integrality gap), two formulations of the complete problem (i.e., considering all stages) were also solved, setting in this case a time limit of 5 h. On the one hand, a model with a fixed number of slots on each unit, which is called FULL FIXED, has been evaluated. Here, the number of slots is equivalent to the slot configuration of the best solution obtained with the proposed method. On the other hand, a model that sets the number of slots based on the maximum number of batches that can be processed on each unit, which is denoted FULL, has been solved. All computations were performed on a generic PC with an Intel Core i7 3.2 GHz processor and 16 GB of RAM.

A comparison of the solutions obtained is presented in Fig. 1 where, for each instance considering a given number of batches, the first column shows the best solution obtained with the proposed methodology, and the remaining columns the solutions of the complete formulations. The total height of each column corresponds to the optimal MIP value found, and is composed by the best possible solution in blue and the remaining gap to prove optimality in red. Fig. 1 shows how, when the number of batches increases, the proposed methodology allowed to obtain better quality solutions with lower bounds similar to the complete problems (FULL and FULL FIXED), even though it only solves subproblems.

The model sizes and CPU times for the 30 batch problem are compared in Table 1. Regarding the sizes of the models, it is observed that the FULL FIXED

alternative reduces more than by half the number of binary variables of the FULL model and, however, the improvement in the solution is not substantial. In contrast, given that partial subproblems are solved, for the proposed methodology the minimum and maximum values of these statistics are shown. The largest evaluated subproblem includes 1556 binary variables compared to 4090 for the FULL FIXED formulation. Besides, in this case the CPU time required to complete the algorithm is 1 h 27 min, compared to 5 h for the complete models. Finally, the Gantt chart of the best solution found is presented in Fig. 2.

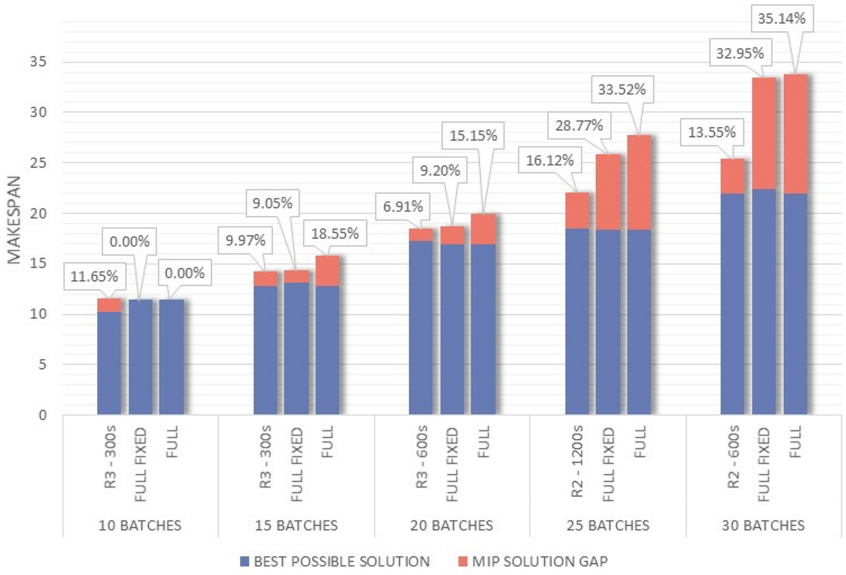


Fig. 1: Comparison of the best solutions obtained with the proposed methodology and the solutions of complete formulations with fixed (FULL FIXED) or maximum possible (FULL) number of slots.

## 4 Conclusions

A mathematical-algorithmic methodology for multiproduct multistage batch plant scheduling that relies on solving a sequence of subproblems has been presented. The proposed method has been applied to an industrial size case study from the pharmaceutical industry, allowing not only to obtain good quality solutions in competitive times but also to provide rigorous information on the associated integrality bounds.

**Acknowledgments.** The authors gratefully acknowledge financial support from Universidad Tecnológica Nacional through grants PID 4932 and PID 7768.

Table 1: Comparison of model sizes and CPU times for the 30 batch problem.

| | | Binary vars | Cont. Vars | Equations | Time (s) |
|---|---|---|---|---|---|
| $R$=2, $T$=600s | min | 120 | 395 | 2935 | 5221 |
| | max | 1556 | 775 | 19583 | |
| FULL FIXED | | 4090 | 775 | 19847 | 18000 |
| FULL | | 10264 | 1384 | 50190 | 18000 |

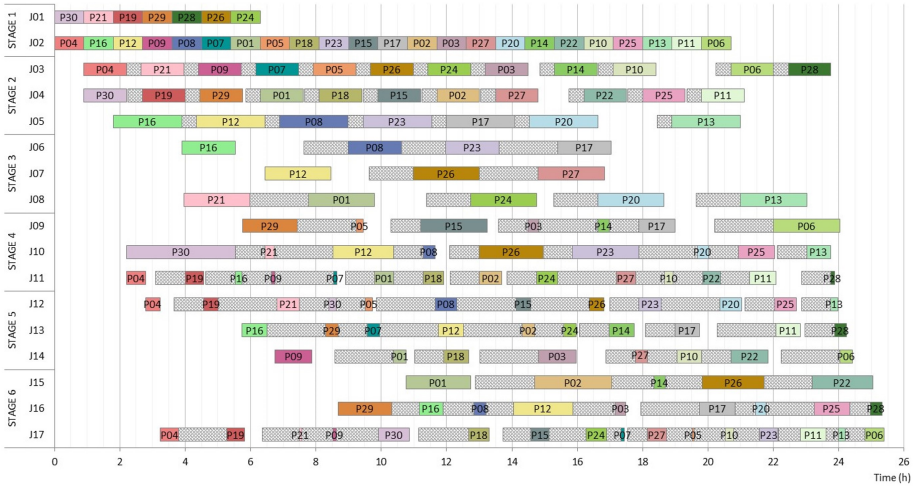


Fig. 2: Gantt chart of the best solution obtained (makespan: 25.38 h) for the 30 batch problem, with parameters $R = 2$ and $T = 600$ s.

## References

1. Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913–946 (2006). doi:10.1016/j.compchemeng.2006.02.008
2. Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassicki, J.: Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. 62, 161–193 (2014). doi:10.1016/j.compchemeng.2013.12.001
3. Castro, P.M., Harjunkoski, I., Grossmann, I.E.: Optimal short-term scheduling of large-scale multistage batch plants. Ind. Eng. Chem. Res. 48, 11002–11016 (2009). doi:10.1021/ie900734x
4. Kopanos, G.M., Méndez, C.A., Puigjaner, L.: 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 (2010). doi:10.1016/j.ejor.2010.06.002
5. Pinto, J.M., Grossmann, I.E.: A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. Ind. Eng. Chem. Res. 34, 3037–3051 (1995). doi:10.1021/ie00048a015