A MILP Model for the Minimization of Cycle Time in Periodic Production Scheduling using Flexible Operation Shifts

Georgios Georgiadisa, Alexandros Koulourisb,\*

aIntelligen Europe, S. Kazantzidi 47, Thermi 57001, Greece

bInternational Hellenic University. P.O. Box 141, Sindos 57400, Greece

akoul@ihu.gr

Abstract

Cycle time is critical for the efficiency of the many industries that choose to operate in a periodic scheduling mode since it largely affects the production throughput. This paper proposes a novel MILP model that addresses the cycle time minimization problem, while considering flexible unit allocation and timing of operations. Furthermore, a new process representation is proposed that successfully encompasses all details of the production reality and incorporates them to the optimization model. Thus, optimized schedules are generated without any loss in representation accuracy. The applicability and efficiency of the developed solution is demonstrated through an illustrative example.

**Keywords**: periodic production scheduling, cycle time minimization, MILP, flexible shifts

* 1. Introduction

Many industries find cyclic and, in particular, periodic scheduling favorable since it minimizes shop floor nervousness and results in easily applicable schedules. Moreover, it tends to generate more robust schedules compared to non-cyclic approaches. The main drawback of periodic scheduling is that it imposes timing restrictions that usually have a negative impact on the production makespan. This may be the reason behind the reduced efforts of the PSE research community to study this problem.

Periodic schedules follow a pattern defined by the process cycle time, which is the constant time interval between the initiation (or completion) of two successive batches. The cycle time is closely tied to throughput, hence minimizing the cycle time is critical for an effective plant operation, as well as for capacity-related studies. Most of the researchers that examined the cyclic scheduling problem assumed that the cycle time is a predefined parameter, while each task could only be processed in exactly one unit during the cyclic schedule (Rodrigues et al., 2014; Vieira et al., 2018). Wu and Maravelias (2019) addressed the latter by proposing an STN-based model that allows for the flexible assignment of tasks to units. Recently, Koulouris and Georgiadis (2023) were the first to address the cycle time minimization problem, while allowing for flexible task allocation. However, in this work the timing of each operation was assumed rigid, lacking any flexibility.

The optimal production scheduling problem is NP-hard, so it can easily become intractable. A way to moderate the complexity of large-scale scheduling problems is by deducing the model representation to an approximated rather than an actual depiction of the production reality, thus deteriorating the applicability of optimization-based solutions. This is one of the main arguments against optimal production scheduling in real industrial scenarios, especially in a time where the implementation of digital twins in production is highly desired (Harjunkoski, 2016).

This paper’s contribution is two-fold. First, a high-detail representation of the production process that can be implemented within a mixed-integer linear programming (MILP) model is proposed. Secondly, for the first time, a MILP model is proposed to tackle the cycle time minimization problem in periodic production scheduling, while taking into account flexible operation timing. The proposed model is general enough to be extended to include industry-specific constraints. A case study is used to demonstrate the application of the model; it has been modelled and scheduled within the scheduling software SchedulePro (Intelligen, Inc.), where the MILP models have also been implemented and solved using the SCIP open-source solver (Bestuzheva et al., 2021).

* 1. Problem Set-up

Let us assume a set of processing tasks *I* required for a single batch that is to be executed indefinitely. A cyclic schedule is conventionally defined as one that repeats every *H* time units, with *H* being the cycle time. Essentially, this means that if a processing task *i* of a batchstarts at time *Li,b,* then the same task of the subsequent batch will start at *Li,b +H.* Now assume that a facility consists of a set of resources *J*.In the examined problem each task uses a pool of available resources for its execution. Different tasks may use the same resources; therefore, the resource pools may overlap (partially or fully) between tasks.In traditional cyclic scheduling each task can be processed in exactly one resource throughout the schedule. This assumption reduces production flexibility, and the utilization of the available facility resources thus may result in lower productivity. Therefore, the concept of “periodicity” is introduced in this paper. This parameter specific to every task *i* (*πi*) denotes the periodicity with which the resource allocation decisions are repeated and is equal to the number of equipment that can process each task. For example, let’s assume that a task *i* can be processed by a set of parallel units {J1, J2, J3}, thus having a periodicity of three. This denotes that if for the first three batches task *i* is allocated to J2, J3 and J1 respectively, then the schedule will follow the same allocation pattern for all remaining batches.

The proposed way of handling unit-task assignments guarantees that every task will be repeated with the same constant frequency and in a constant pattern dictated by parameter *πi*. This is made clear in the simple example shown in Figure 1. In this example, four batches of two tasks I1 and I2 that share the same equipment pool (J1, J2, J3) are scheduled. Increasing the periodicity from 1 to 2 halves the cycle time and reduces the production makespan from 8 to 5 time units. Notice that knowing the cycle time and the resource allocation decisions of the first *πi* batches (depicted with grey color in the figure) is enough information to schedule an infinite number of batches.

A screenshot of a computer

Description automatically generated

Figure 1. Impact of periodicity on cycle time

* 1. Mathematical Formulation
     1. Detailed production process representation

To achieve a detailed representation of the production process, the concepts of procedures and operations are utilized. A production stage that utilizes a resource for its entire duration is called *procedure*. A procedure may consist of multiple distinct *operations* that may also require the use of additional resources. For example, a fermentation procedure may consist of operations such as: Loading of media and microorganisms, Heating, Fermenting, Transferring-out the broth and Cleaning-in-Place (CIP). The entire procedure will utilize a fermentation tank, but the CIP operation may, in addition, need the utilization of a CIP-skid for its execution.

The timing of all operations in a procedure and the entire production recipe is determined by strict dependency links. More specifically, the start or the end of an operation can be set to coincide with the start or the end of another operation plus a (positive or negative) fixed time shift. Additional links can be set also for the duration of operations if they are inter-dependent. Compared to traditional optimization-based approaches that represent the production process with outmost timing flexibility and then tighten up its execution with additional constraints, this is a more rigid representation, but also more realistic as far as chemical processing is concerned.

This rigid representation can be relaxed with the introduction of *flexible shifts* which represent positive or negative shifts in the execution of an operation if the production recipe allows them. For example, in the previously mentioned fermentation procedure, the CIP operation could potentially be delayed up to some time if the skid required for its execution is not available. In that case, a flexible operation shift up to a maximum value (the “dirty-hold” time) can be introduced for the start of the CIP operation with respect to the end of the Transfer-out operation in the fermentation procedure. Within this framework, intra-batch scheduling decisions can be optimized using the flexibilities introduced in the recipe either in the form of resource pools and/or flexible operation flexible shifts.

To mathematically represent the above structure, each operation is given a triplet of variables for its start, duration and end which we call “timing elements”. A set *N* is introduced that includes all these timing elements for all operations. Flexible operation shifts are modelled as dummy operations that share the same triplet of timing elements. Because of the scheduling and duration links between all operations, all these timing elements are interdependent. If all these elements are sorted based on dependency links, a lower diagonal matrix *σn,n’* can be composed which determines how a timing element can be calculated from the others of higher order in the dependency chain. The diagonal of the matrix represents fixed values (durations or shifts). In the case of dummy operations representing flexible shifts, the diagonal value of the duration element represents the upper bound in the flexible shift. For all tasks (procedures and operations) that need to be incorporated into the optimization problem, three additional mapping sets are introduced: and that correlate respectively the start, end and duration of task *i* to the above timing elements.

Figure 2 illustrates the use of these variable sets by considering the simple example of the fermentation procedure with only two operations (Fermentation and CIP) and a flexible shift on the CIP operation. For the total of 3 operations (two real and one dummy representing the flexible shift), nine timing elements are created (for the start, end and duration of each operation).

A screenshot of a computer

Description automatically generated

Figure 2. Detailed timing representation

* + 1. MILP formulation

A MILP-based approach is developed here to address the cycle time minimization process. The proposed model is based on the general precedence modeling framework and the timing representation discussed above while assuming periodic execution of batches and allocation of resources. Given a number of batches *B* of processing tasks *I* to be processed in the available units *J*, the optimization model generates an optimal production schedule with minimum cycle time. 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 *Yi,b,j* and sequencing decisions *Xi,b,i’,b’*. Moreover, continuous variables are introduced to model all time-related decisions like start (*Li,b*) and completion (*Ci,b*) of tasks, value of timing elements (*Vn,b*) and duration of flexible operation shifts (*Fn,b*). The proposed model comprises of constraints (1)-(13) and the objective function (14).

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Constraint set (1) states that each batch *b* of task *i* will be processed by exactly one unit *j*. Subset *IJi,j* denotes the units that can process a task. Notice that since the allocation decisions are repeated in the cyclic schedule, only the batches dictated by the periodicity parameter are considered. In the two following constraints sets, the value of timing elements related to all operations () and flexible shifts () are calculated based on their relative timings (*σn.n’*) and the variable duration of the flexible operation shifts. The periodicity of a timing element *πn* is equal to the periodicity of the related task *i.* Constraints (4) ensure that the start time of a task will be less than the value of all relative start elements plus a time shift imposed by the cycle time *H*. Similarly, the completion times are calculated in the next constraint set (5). Notice that the inequalities are required since in the case of procedures a task may be comprised by numerous operations. Next, constraint set (6) imposes the cycle time constraints. Constraints (7) and (8) are complementary sequencing constraints that make use of a big-M parameter. Constraint set (9) guarantees 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 unit. Finally, constraint sets (10)-(13) are tightening constraints that take advantage of the schedule’s cyclic nature to improve the model’s computational efficiency.

Note that all variables introduced by the detailed timing representation are continuous, as a result the accuracy of the generated schedules is improved without significantly increasing the model’s computational complexity. Furthermore, one of cyclic scheduling advantages is that only a small number of batches must be optimized. As a result, the model is expected to remain computationally efficient when dealing with large-scale industrial applications.

* 1. Illustrative example

An illustrative example is presented in this section to demonstrate the use of the proposed optimization model and the impact of flexible operation shifts on key production metrics. The production process consists of three procedures with 10 operations. The first and third procedures can be processed by a single equipment, T-101 and CEN-101 respectively, while the second can be processed by a pool of equipment (FR-101, FR-102 and FR-103). At the end of each procedure a cleaning operation takes place that necessitates the use of a CIP skid (CIP-1). The duration and start times in hours for all operations are {4, 0.33, 1.5, 0.33, 48, 6, 1.5, 6, 1.5, 4.5} and {0, 4, 4.33, 4, 4.33, 52.33, 58.33, 52.33, 58.33, 59.83} respectively. Flexible operation shifts with a maximum duration of 4 hours are introduced to the CIP operations. The proposed model has been applied to tackle the described problem and was able to generate optimal decisions instantaneously. The results are illustrated in Figure 3, where the Gantt charts of the optimal periodic schedule of 10 batches without (3a) and with (3b) flexible operations shifts are depicted. Introducing a flexible operation shift with duration of 1.83 h in the CIP operation of the third procedure has a significant impact on the production efficiency. In particular, the cycle time is reduced by 6.95 % (from 20 h to 18.61 h) and the production makespan by 4.4 % (from 244.3 h to 233.65 h). Notice also that applying the optimal timing flexibilities eliminates any unnecessary idle production times in the equipment pool of the second procedure.

A screenshot of a computer screen

Description automatically generated

Figure 3. Gantt chart of 10 batches a) without and b) with flexible operation shifts

* 1. Conclusions

A MILP approach is presented in this paper for the calculation of the minimum cycle time in periodic batch process scheduling with resource sharing and flexible operation shifts. The applicability and efficiency of the developed model is shown through an illustrative example. Optimal periodic schedules can be promptly generated. The results clearly underline the beneficial effects of smartly chosen timing and resource selection flexibilities in key production metrics that can unlock the production’s full potential and critically enhance its efficiency. The incorporation of industry-specific constraints in large-scale problems could potentially limit the efficient applicability of the proposed methodology. To address this issue, a meaningful future research direction would be the integration of the developed model within a decomposition scheme.

References

[Bestuzheva](https://arxiv.org/search/math?searchtype=author&query=Bestuzheva,+K), K., [Besançon](https://arxiv.org/search/math?searchtype=author&query=Besan%C3%A7on,+M), M.,  [Chen](https://arxiv.org/search/math?searchtype=author&query=Chen,+W), W., [Chmiela](https://arxiv.org/search/math?searchtype=author&query=Chmiela,+A), A., [Donkiewicz](https://arxiv.org/search/math?searchtype=author&query=Donkiewicz,+T), T., van [Doornmalen](https://arxiv.org/search/math?searchtype=author&query=van+Doornmalen,+J), J., [Eifler](https://arxiv.org/search/math?searchtype=author&query=Eifler,+L), L., et.al. 2021. The SCIP Optimization Suite 8.0. arXiv preprint arXiv:2112.08872.

Harjunkoski, I. 2016. Deploying scheduling solutions in an industrial environment. Computers & Chemical Engineering, 91, 127-135. [doi.org/10.1016/j.compchemeng.2016.03.029](https://doi.org/10.1016/j.compchemeng.2016.03.029)

Intelligen Inc. https://www.intelligen.com/products/schedulepro-overview/ (access November 2023)

Koulouris, A. and Georgiadis, G.P., 2023. On the minimization of cycle time in periodic production scheduling. Computer Aided Chemical Engineering, 52, 107-112. [doi.org/10.1016/B978-0-443-15274-0.50018-4](https://doi.org/10.1016/B978-0-443-15274-0.50018-4)

Rodrigues, D., Castro, P.M., Matos, H.A., 2014. New Multiple Time Grid Continuous-Time Formulation for the Cyclic Scheduling of an Industrial Batch Plant, Computer Aided Chemical Engineering, 33, 1807-1812. [doi.org/10.1016/B978-0-444-63455-9.50136-7](https://doi.org/10.1016/B978-0-444-63455-9.50136-7).

Vieira, M., Paulo, H., Vilard, C., Pinto-Varela, T., Barbosa-Póvoa, A.P., 2018. Risk assessment for the design and scheduling optimization of periodic multipurpose batch plants under demand uncertainty. Computer Aided Chemical Engineering, 43, 991-996. [doi.org/10.1016/B978-0-444-64235-6.50174-1](https://doi.org/10.1016/B978-0-444-64235-6.50174-1).

Wu, Y. and Maravelias, C.T., 2019. A general model for periodic chemical production scheduling. Industrial & Engineering Chemistry Research, 59(6), pp.2505-2515. [doi.org/10.1021/acs.iecr.9b04381](https://doi.org/10.1021/acs.iecr.9b04381)