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

**ISBN** 978-88-95608-71-6; **ISSN** 2283-9216

DOI: 10.3303/CET1974221

# Slots Start-up Synchronization with Shared Resources Dependency

Carlos G. Palacín[a,*], Carlos A. Méndez[b], Cesar de Prada[a,c]

[a]Dpt. Of Systems Engineering and Automatic Control, Escuela de Ingenierías Industriales, Sede Mergelina, Universidad de Valladolid, C/Real de Burgos, s/n, Valladolid 47011, SPAIN
[b]Center for Advanced Process Systems Engineering (CAPSE), INTEC (UNL - CONICET), Industrial Engineering Dpt., Güemes 3450 - Santa Fe (3000), Argentina
[c]Institute of Sustainable Processes (ISP), University of Valladolid, C/Real de Burgos, s/n, Valladolid 47011, SPAIN
carlos.gomez@autom.uva.es

Batch process entails a bottle neck in the majority of plants. This means that an optimal synchronization with the rest of the process lines is mandatory in order to increase the productivity. To obtain the best possible schedule, instead of a feasible solution, the problems usually are implemented using mathematical programming, however the batch processes are considered independent one from each other to facilitate the coding and resolution. In this work the authors present a new formulation that allows an optimal schedule of batch processes which length depend on the synchronization of the startup of those processes. This solution is implemented to use as a real time optimization tool with a rolling horizon method. The distribution of shared resources among the devices is also keep into account.

## 1. Introduction

In the majority of the industrial sites the continuous processes represent a huge percentage of the actions and decisions that have to be made. However, it is also important to assign equipment and resources to each process, to determine the starting times and duration of some of those or to manage the maintenance procedures for the devices or pipes, among other important decisions, what is called scheduling of the plant. In order to obtain good performance feasible solutions these problems can be tackle from different angles: using graphs (Osz and Hegyhati, 2018), automata theory (Abdeddaïm et at., 2006), evolutionary algorithms (Safaei et al., 2008) or heuristics (Eles et al., 2018)(Veeragan et al., 2018)(Ren et al., 2018), for example. Even though this is a well know problem in the operational research field, there are yet many open problems in the industrial sector, where the search for the optimality, that is priorized over the feasibility, depends in good measure of the formulation chosen (Harjunkoski et al., 2014).
In the food industry environment, batch processes frequently appear in the middle of continuous production lines. If the equipment required for these discontinuous processes are shared among some lines, a scheduling problem arises. These problems can be more complex if the slots have different processing time depending on the device that fulfills them, the slot size or composition, or the production line that has freed them, for example. Usually, they can be solved efficiently with a general precedence formulation (Méndez et al., 2006), that determines in which device and order each slot shall be initiated.
One batch process usually can be divided in three stages, one start-up stage, one main process stage, and one closing stage. Each one will required different resources that can have different natures, as equipment, raw material, or personal. With the classical approach, the resources are reserved when the stage starts, and freed again when it finishes.
However, there are shared resources that cannot be reserved, as steam or electrical power. If the availability of those is enough, the batch processes can be independent from each other; nevertheless, if the resource consumption has a peak in one of the stages, it may affect other slots in the same stage, as the system has been scaled for the typical consume. There are two main strategies to deal with this dependency: one is to set a maximum affordable amount of shared resources, and to synchronize the consuming profiles of the slots in

Paper Received: 20 March 2018; Revised: 17 June 2018; Accepted: 30 November 2018

Please cite this article as: Gomez Palacin C., Mendez C.A., De Prada C., 2019, Slots Startup Synchronization with Shared Resources Dependency, Chemical Engineering Transactions, 74, 1321-1326 DOI:10.3303/CET1974221

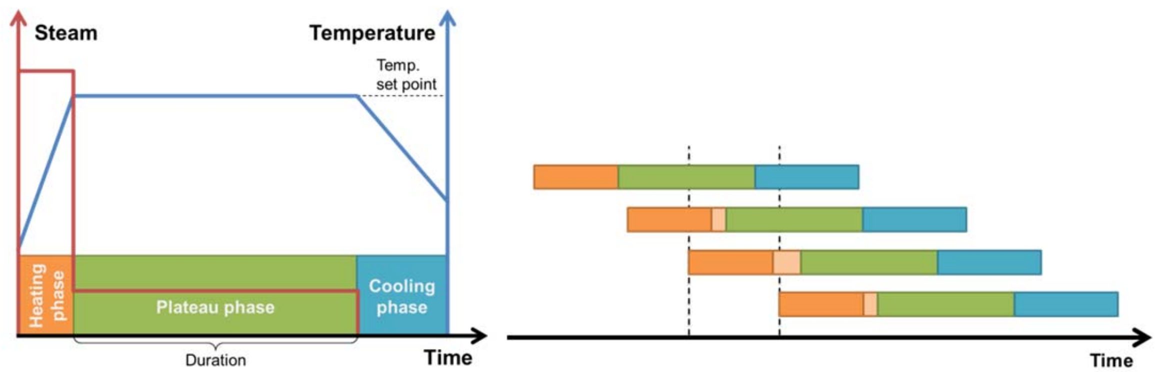
order to never surpass this bound (de Prada et al., 2019); the other one is to approximate the variations that the slots inference and to solve the scheduling optimization problem taking into account this alteration.
In this work we develop an algorithm in order to synchronize batch processes with a shared resource that affects the duration of the slots. And, we show a practical implementation in a tuna canning factory, where the steam available can only fulfil the requirements of one device start-up at a time.
The rest of the paper is structured as follows: next section summarises the real case that has motivated this work. Section 3 describes the mathematical formulation developed. Last sections show some results and conclusions with future work.

## 2. Industrial case study

The algorithm has been implemented for a real use case in a tuna canning factory. Once the tuna is cooked, it is introduced in the cans, which are fill up with different food preservatives, as oil or pickled sauce for example, and then sealed. Afterward the cans have to be sterilized by a procedure that maintains certain temperature during a fixed time. This procedure depends on the type of raw material, can and preservative, which create several different recipes (calling recipe to each sterilization program) for every combination. After the sterilization, the cans are packed and sent to storage, before being distributed to the clients.
Either the filling and sealing process like the packaging one are continuous processes; however, the sterilization is a batch process. In order to keep the rhythm from previous production lines, the sterilization is done to a huge amount of cans at the same time. The cans are placed in big industrial metal carts, which are introduced by groups up to the maximum capacity of the sterilizers. The carts are filled up directly at the ending of the production lines, this makes that each cart only contains one type of final product, which means only one of recipe is related to each full cart. Once a cart is release from a production line, it is pushed to an available autoclave or to an input buffer in front of the autoclaves. Then, a count-down starts; each can have to have its sterilization started within a security gap in order to prevent microorganism from growing, which has the same duration for all the types of products. When one autoclave is full, or if there are not carts with the same sterilization program expected before the maximum waiting time of all the carts introduced goes by, it is closed and the sterilization process starts. Once it has finished, the carts are pull out of the autoclaves and placed in the entrance of the packaging processing lines or in the input buffers if they are already busy.

### 2.1 Problem definition

Usually, the operators wait until they have an autoclave full of the same type of carts, meaning the associated recipe by type of cart. However, some types of cans differ only on the packaging and have the same sterilization process associated and can be gathered in the same group to be sterilized. On the other hand, if a few carts are going to surpass their security gap, it's preferable to introduce them with other carts which don't differ with their recipe, instead of losing the carts. The mixed group would have to be submitted to the most severe sterilization process, in order to assure the microorganism lethality for all the cans. This will harm some of the products, so this option has to be avoided as long as it's possible.




*(a) Independent process* *(b) Several processes influence*

*Figure 1: Temperature and steam consumption profiles for sterilization processes*

All the sterilization processes consist in three phases (see Figure 1a): a heating phase, when the temperature is increased as quickly as possible up to the set point fixed by the recipe; a plateau phase, when the temperature is maintain for the time set on the sterilization program; and a cooling phase, that takes place in

order to reduce the time needed to empty the autoclaves. The temperature is controlled using raw steam to heat the system, which is a shared resource among all the autoclaves. The incoming flow of steam in the heating phase is huge, which makes the pressure in the pipes drop. This means that, if more than one heating phase occurs simultaneously, the durations of these phases are increased. Meanwhile, the plateau phase and the cooling phase have no effects on the rest of autoclaves, and they are not modified by them either.
Therefore a scheduling problem appears, where the carts have to be merged in groups and sterilized, preventing a maximum waiting time to be surpassed for each cart, and trying to reduce mixed groups. Nonetheless, the synchronization of starts up appear as a main issue due to resource availability limit, as this directly influences on the starting time of each sterilization.

## 3. Mathematical formulation

The problem has been formulated as a MILP problem, in order to been able to use the efficient MILP solvers. Due to the high amount of carts and their arrival frequency, a typical scheduling solution would require too much computational power to solve it, and would be impossible to implement the solution in the factory. Therefore, a real-time optimisation problem is defined, that will give a solution for a relative short horizon (hours), within which the optimisation will be run again, applying a rolling horizon philosophy.
It has been defined three sets: the carts that have to be sterilized, called $\mathcal{I}$; the group of carts that are going to be sterilized in the same autoclave at the same time or slots, named $\mathcal{J}$; and the autoclaves $\mathcal{K}$. Then, two binary variables have been defined to represent the relations between the sets: $X_{i\in\mathcal{I},j\in\mathcal{J}}$, that sets that one cart $i$ belongs to one slot $j$; and $Z_{j\in\mathcal{J},k\in\mathcal{K}}$, that express that one slot $j$ is proceed in one device $k$. These can be seen in the following equations. Eq(1) assures that one cart can only be assigned to one slot; however, as this tool is a RTO, and it will be run recurrently, one cart doesn't have to be assigned to be sterilized if its maximum waiting time is not reached within the prediction horizon. Eq(2) and Eq(3) make that each sterilization process will have at least one cart assigned, and that the capacity of the autoclaves are not surpassed. And, Eq(4) forces each slot to be related to one and only one autoclave.

$$\sum_{j\in\mathcal{J}} X_{i,j} \leq 1 \qquad \forall i \in \mathcal{I} \tag{1}$$

$$\sum_{i\in\mathcal{I}} X_{i,j} \leq \zeta \qquad \forall j \in \mathcal{J} \tag{2}$$

$$\sum_{i\in\mathcal{I}} X_{i,j} \geq 1 \qquad \forall j \in \mathcal{J} \tag{3}$$

$$\sum_{j\in\mathcal{J}} Z_{j,k} = 1 \qquad \forall k \in \mathcal{K} \tag{4}$$

There are two input parameters defined for each cart $i$, $ta_{i\in\mathcal{I}}$ and $te_{i\in\mathcal{I}}$. The first one represent the time instant when a cart is release from the sealing line, and the second is the duration of the plateau phase plus the cooling one needed for the cart. With those inputs, we can defined $tp_{j\in\mathcal{J}}$ as the duration of the maintaining and cooling procedures for a slot, and $ts_{j\in\mathcal{J}}$ as the start time of a sterilization process.
It has to be also defined one parameter $\tau$ that represents the maximum waiting time for each cart, and one parameter $\sigma$ for the maximum clearance allowed among the sterilization process duration of carts in the same slot. Eq(5) and Eq(6) bound the starting time for a slot, between the arrival of the carts introduced in it, and the maximum waiting time of them. In order to keep the linearity of the problem an artificial parameter sufficiently large is introduced ($\mu$), that makes the constraints idle when it is not cancelled, and vice versa. The so called *big M method* to translate logical predicates into mathematical formulation (Winston, 2008).
Eq(7) and Eq(8) bound the duration of the process between the minimum and maximum durations of the sterilization process of the carts included in the slot, whit a maximum difference of $\sigma$.

$$ts_j \geq ta_i - \mu \cdot \left(1 - X_{i,j}\right) \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{5}$$

$$ts_j \leq ta_i + \tau + \mu \cdot \left(1 - X_{i,j}\right) \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{6}$$

$$tp_j \geq te_i \cdot X_{i,j} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{7}$$

$$tp_j \leq te_i + \sigma + \mu \cdot \left(1 - X_{i,j}\right) \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{8}$$

As it has been said, the carts are only forced to be included in one slot, if they arrive before a time horizon, coded as the parameter $\eta$. This can be seen in Eq(9), which is only active if the cart $i$ is not assigned.

$$ta_i + \mu \cdot \sum_{j \in \mathcal{J}} X_{i,j} \geq \eta \qquad \forall i \in \mathcal{I} \tag{9}$$

Once the groups are done, the problem has to deal with the scheduling of the sterilizations. First, it is defined one binary variable ($W_{j1 \in \mathcal{J}, j2 \in \mathcal{J}}$) that specifies that two slots ($j1$ and $j2$) share steam during the heating phase. In order to reduce the complexity of the problem this variable is only defined for half of the combinations, as if one slot affects another, it happens also the other way around. However, this creates a huge amount of constraints, making the problem heavy and slow to solve. In order to prevent this, we suggest defining an artificial pre-order for the slots which doesn't affect the result. It is a fact that each slot will start before or at the same time than a previous one, except for the first one. Then we defined an order in the set of slots, which will define the order they shall follow; as none slot is preassigned to any autoclave or cart, this assumption doesn't affect the result. This new formulation let the problem deal with the synchronization easily taking into account the shared resources. This order is express in Eq(10), where one slot $j2$ starts after or at the same time that another $j1$ if they have this order in the set.

$$ts_{j1} \leq ts_{j2} \qquad \forall j1, j2 \in \mathcal{J} : ORD(j1) < ORD(j2) \tag{10}$$

Then, the general precedence formulation is modified, as the typical binary variable that sets the order between two slots in the same equipment, is no longer needed. Now, the order is already know, and the formulation only have to check the equipment they are assigned to, as it can be seen in Eq(11), where it is introduced a new real variable $th_{j \in \mathcal{J}}$, the heating time required for a slot $j$. In this particular case, the heating time of a slot, which is the start-up time of a slot in a general formulation, has a constant minimum value (κ) plus a term proportional to the number of slots that coincide in this phase, shown in Eq(12) where it can be seen that the binary variable is only defined for the inferior triangular matrix form by the combination of slots. Eq(13) reduce the options to the solver removing impossible combinations; if one slot doesn't affect a previous one, none of the successive will.

$$ts_{j1} + th_{j1} + tp_{j1} \leq ts_{j2} + \mu \cdot \left(2 - \left(Z_{j1,k} + Z_{j2,k}\right)\right) \qquad \forall j1, j2 \in \mathcal{J} : ORD(j1) < ORD(j2), \forall k \in \mathcal{K} \tag{11}$$

$$th_j = \kappa + \alpha \cdot \left( \sum_{j' \in \mathcal{J} : ORD(j') < ORD(j)} W_{j',j} + \sum_{j' \in \mathcal{J} : ORD(j) < ORD(j')} W_{j,j'} \right) \qquad \forall j \in \mathcal{J} \tag{12}$$

$$W_{j,j1} \geq W_{j,j2} \qquad \forall j, j1, j2 \in \mathcal{J} : ORD(j) < ORD(j1) < ORD(j2) \tag{13}$$

Eq(14) and Eq(15) constraint the synchronization of the slots. The first one represents when two slots doesn't coincide, $W_{j1,j2}$ will be zero, therefore the start of the second slot will happened after the heating phase of the first one. On the other hand, if the binary variable is equal to one, this equation will have to effect, and Eq(15) will force the start of the second slot to happen before the heating is over, keep in mind that the starts are already force to happen one after the other.

$$ts_{j1} + th_{j1} \leq ts_{j2} + \mu \cdot W_{j1,j2} \qquad \forall j1, j2 \in \mathcal{J} : ORD(j1) < ORD(j2) \tag{14}$$

$$ts_{j1} + th_{j1} \geq ts_{j2} - \mu \cdot \left(1 - W_{j1,j2}\right) \qquad \forall j1, j2 \in \mathcal{J} : ORD(j1) < ORD(j2) \tag{15}$$

Finally, the ending time of the overall schedule is defined in Eq(16) in a new variable $tf$ in order to be able to minimize the makespan.

$$tf \geq ts_j + th_j + tp_j \qquad \forall j \in \mathcal{J} \tag{16}$$

## 4. Results

The problem has been solved in a laptop computer, with a i7-4510U processor, optimized to reduce energy consumption instead of computing power, which make it a good system to test tools that are going to be installed in an industrial site. Coded in GAMS 25.1.1, using Cplex 12.8.0.0 as MILP solver.
First optimization option is to minimize the makespan of the system, Eq(17). However, when the production is key to meet deadlines, the objective could be increasing the number of carts sterilized Eq(18), even though it could reduce the quality of some carts, preventing exceeding the maximum waiting time.

$$\min_{X_{\mathcal{I},\mathcal{J}},Z_{\mathcal{J},\mathcal{K}},W_{\mathcal{J},\mathcal{J}},ts_{\mathcal{J}},tp_{\mathcal{J}},th_{\mathcal{J}}} tf \tag{17}$$

$$\max_{X_{\mathcal{I},\mathcal{J}},Z_{\mathcal{J},\mathcal{K}},W_{\mathcal{J},\mathcal{J}},ts_{\mathcal{J}},tp_{\mathcal{J}},th_{\mathcal{J}}} \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} X_{i,j} \tag{18}$$

One example can be seen in Figure 2, where the schedule of two hundred carts with five different recipes is shown, minimizing the makespan. There have been set sixteen autoclaves, and fifteen slots have to be sterilized. However the optimizer has found a solution, using only thirteen devices, in less than one minute; which makes the algorithm suitable to use in a real-time software schedule support system.

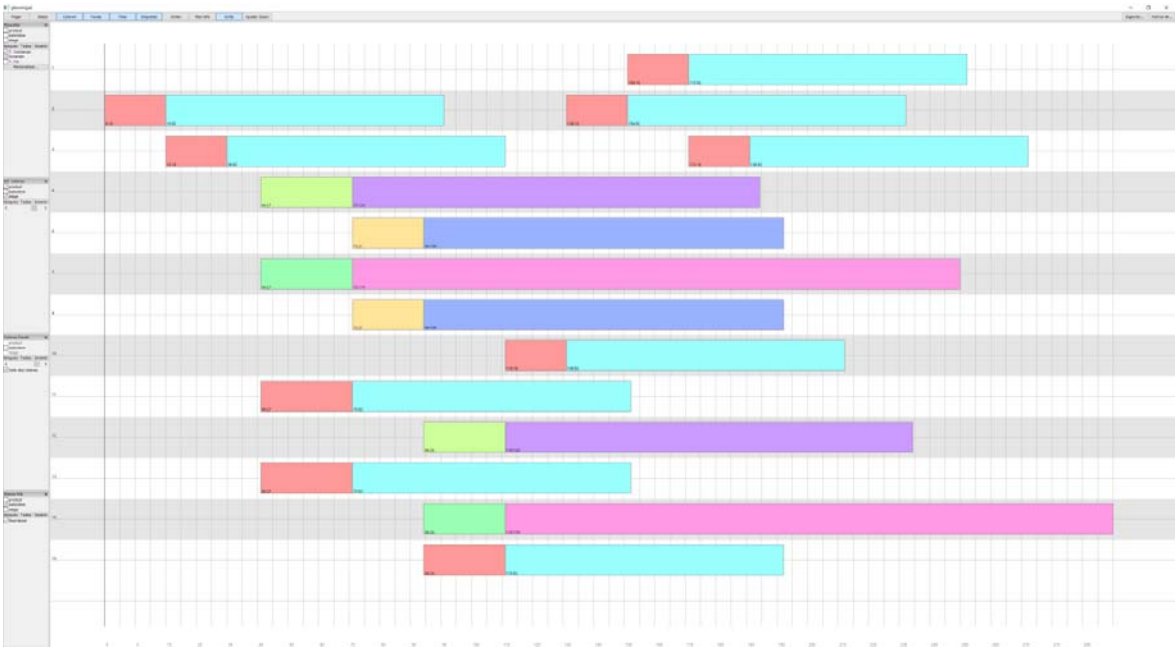


*Figure 2: Real scheduling solution example*

In the figure, the time is represented in the horizontal axis, and the vertical one represents the devices. Each batch process is divided in two subprocesses, the first part is the heating and the second one represents the plateau phase and the cooling one, as the firsts length is variable but the seconds one is not. The colors represent the different type of process available. It can be seen that the global duration of the same type of process depends on the number of sterilizations that start within the heating phase, maintaining the sterilization and cooling length but increasing the start-up phase proportionally to the number of processes.

## 5. Conclusions

A new approach to deal with the schedule of slots which duration depends on the synchronization with others has been presented in this work. This formulation increases the speed of resolution without losing optimality. It tackles the scheduling of processes which lengths depends on the synchronization itself and shared resources consumption. It can be extended for more resources or more phases by adding more binary variables, but keeping the kernel of the formulation similar.

It has been tested in a real use case, proving its efficiency and the possibility of including it in RTO tools. Nevertheless, the algorithm has to be improved in order to increase the prediction horizon and to be able to modify the perturbations profile between the heating processes. It can also be included uncertainty in the production lines frequency and resource availability using a stochastic approach.

## Acknowledgments

This research has received funding from the European Union Horizon 2020 research and innovation programme (grant number 723575) and from the Spanish Government under project (DPI2015-70975-P MINECO/FEDER). Cesar de Prada also acknowledges support by the regional government of Castilla y León and the EU-FEDER (CLU 2017-09).

## References

Abdeddaïm, Y., Asarin, E., Maler, O., 2006, Scheduling with timed automata, Theoretical Computer Science, 354(2), pp.272-300.

Eles A., Cabezas H., Heckl I., 2018, Heuristic algorithm utilizing mixed-integer linear programming to schedule mobile workforce, Chemical Engineering Transactions, 70, 895-900 DOI:10.3303/CET1870150

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G. and Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering, 62, pp.161-193.

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, Computers & Chemical Engineering, 30(6-7), 913-946

Osz O., Hegyhati M., 2018, An s-graph based approach for multi-mode resource-constrained project scheduling with time-varying resource capacities, Chemical Engineering Transactions, 70, 1165-1170 DOI:10.3303/CET1870195

de Prada, C., Mazaeda, R., & Podar, S., 2019, Receding horizon scheduling of processes with shared resources, Computers & Chemical Engineering.

Ren C., Zhao J., Chen L., Huang Y., 2018, Application of firefly algorithm in scheduling optimization of combined cooling, heating and power with multiple objectives, Chemical Engineering Transactions, 67, 829-834 DOI: 10.3303/CET1867139

Safaei, N., Banjevic, D., Jardine, A.K., 2008. Multi-objective simulated annealing for a maintenance workforce scheduling problem: a case study. In Simulated annealing. InTech

Veeraragavan S., Chan W.M., Chew I.M., Foo D.C.Y., 2018, Task-based shift scheduling through process integration technique , Chemical Engineering Transactions, 70, 1057-1062 DOI:10.3303/CET1870177

Winston, W.L., 1987, Operations Research: Applications and Algorithms, PWS-Kent, Boston, Massachusetts