MILP-based Approaches for Medium-Term Planning and Scheduling in Multiproduct Multistage Continuous Plants

Adrián M. Aguirre[1], Songsong Liu[2], Lazaros G. Papageorgiou[1,*]

[1]Centre for Process Systems Engineering, Department of Chemical Engineering, UCL (University College London),
Torrington Place, London WC1E 7JE, United Kingdom
[2] School of Management, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, United Kingdom
* E-mail: l.papageorgiou@ucl.ac.uk, phone: +44 (0)20 7679 2563

Abstract. This paper addresses the planning and scheduling problem of a multiproduct multistage continuous plant by three novel MILP-based (Mixed Integer Linear Programming) models. These models combine a TSP (Traveling Salesman Problem) formulation with the main ideas of general precedence and unit-specific general precedence concepts to provide hybrid discrete/continuous time representations of the system. Also, an efficient solution approach involving rolling horizon and iterative-improvement algorithm is derived for solving medium-size instances of the problem. Results analyses for different model's parameters demonstrate the benefits of the new formulations and the effectiveness of the solution approach presented in this work.

Keywords: Planning and Scheduling, Integer programming, Travelling salesman, medium-term optimization.

1. Introduction:

The planning and scheduling of multiproduct multistage batch/continuous plants is one of the most relevant problems rising for the industry in many different manufacturing areas nowadays, e.g., pharmaceutical industry[1-,5], assembly production lines[6,7] and automated manufacturing systems[8-11]. Specially, planning and scheduling of multiproduct multistage batch plants has been extensively studied in the past for many authors in many different applications and a large number of contributions and approaches have been developed to tackle this kind of problems. Most of the applications and approaches have been developed using mixed integer programming techniques and suitable literature reviews can be found in literature[12-16], and more recent work[17-19].

On the other hand, the planning and scheduling problem of multiproduct continuous plants has received less attention in the literature despite its practical importance in the chemical process industries[20]. A number of previous contributions for multiproduct continuous plants were developed for single-stage processes[21-24], but not many have been proposed for multistage continuous processes[25-29].

In this paper, we focus our attention on the planning and scheduling of a multiproduct multistage continuous plant with a single production unit per stage which has not been deeply studied in the past. Such process was firstly investigated by Pinto and Grossmann[26]. In that work they studied the cycle scheduling problem of multiple products in continuous plants considering a sequence of stages with a single production line. This problem also was revisited by Pinto and Grossmann[13] and also extended by many authors. Alle and Pinto[30] proposed an MILP considering a similar problem assuming a single unit per production stage in a flowshop plant with intermediate storages. Later, Alle et al.[31] addressed the cyclic scheduling of cleaning and production operations with performance decay by introducing an MINLP model. An extended version was proposed to consider finite intermediate storage and also a global optimization algorithm based on spatial branch and bound was developed to globally solve the problem[32]. Related works based on a similar problem, considering a hybrid flowshop facility with dedicated units, were found in literature[33-36]. Similar kind of problems of multistage processes with single-unit per stage are very common in batch production plants, known as "flow-shop" problems, where a set of jobs have to follow the same specific sequence of units to be produced[16,17]). Particular case studies with a single-unit per production stage can be found in semiconductor manufacturing processes[37-46], as well as in automated transportation systems[47-49], aircraft manufacturing[50], motor manufacturing[51], sanitary ware production[52], clothing production[53] and electronics industry[54], etc.

In this problem one of the most important issues relies in the ability of handling sequence-dependent activities such us, changeovers or transfers, between consecutive jobs or units. It has been demonstrated in the past, by many literature contributions, that precedence-based formulations are more effective than other approaches when sequence-dependent changeovers or transferring times occur in the systems. In particular Liu et al.[24] has demonstrated that a discrete/continuous time formulation based on the classic Travelling Salesman Problem

(TSP) with sub-tour breaking constraints works better than the slot-based approaches[22,23] for solving medium-term multiproduct single-stage problems in continuous plants. Then, Liu et al.[25] extend the previous work by introducing timing constraints to deal with multiple stages working in series commonly known as flowshop system. For this, sub-tour breaking constraints based on the immediate-precedence variables.[55] were proposed to deal with sequencing decisions. This formulation introduce a large number of sequencing binary variables and Big-M constraints that may difficult the resolution of the system in reasonable computational time. In order to overcome this limitation other precedence-based formulations were study in this paper.

General-precedence was demonstrated to be one of the best efficient representations to tackle continuous-time scheduling models (Mendez et al.[16]), by reducing the number of binary variables significantly in comparison with other existing formulations, such as immediate-precedence representation (Figure 1).

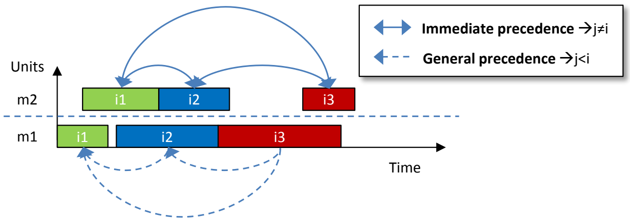


Figure 1. Precedence-based continuous-time representation.

However, there are still some particular issues regarding general-precedence models. For example, general-precedence models cannot cope well with sequence-dependent changeovers when triangle inequality violations occur in the data and some errors can be found when the solution is reported. Other important limitation relies on the Big-M constraints. This kind of constraints relaxes the domain of the continuous variables and the relaxed solution of the problem. Due to this, sometimes general-precedence models are time consuming to reduce and close the relative gap, between the best integer and the relaxed solution, especially when this value is relatively small. In order to cope with the weakness of the general-precedence formulation, many models and approaches have been developed recently. Marchetti & Cerda[56] proposed a unit-specific general precedence-based formulation for the short-term scheduling of single-stage multiproduct batch plants with parallel units and sequence-dependent changeovers, in which a three index binary variable for the sequencing of different tasks in the same unit was developed. Following the same idea, Kopanos et al.[57] developed a novel unit-specific general-precedence model for correctly tackle the problem of sequence-dependent changeovers in a single-stage batch plant with multiple units in parallel. An extra binary variable for the immediate (local) precedence of two different tasks in the same unit and a continuous variable to determine the relative position of those tasks in the unit are defined. However, the model size becomes considerably large due to additional sequencing-allocation constraints, including some Big-M constraints. Then, a new formulation based on general-precedence representation was proposed by Aguirre et al.[58] for the multiproduct multistage batch process in the automated manufacturing systems with sequence-dependent transferring times, where binary variables are proposed for transferring products between stages by using the ideas of general-precedence representation. The sequencing variables there were reduced by a half from the other formulations described above. Recently, a general-precedence approach was proposed by Cóccola et al.[59] for the optimal scheduling of process operations within considering sequence-dependent issues. This new approach introduced a new constant parameter into the sequential inequalities constraints, eliminating triangle inequalities and solving the problem to the global optimal solution.

The precedence-based models presented above were tested in different medium-size planning/scheduling problems demonstrating the efficiency of the solution performance regarding other formulations. But, is still a challenge for researchers and practitioners the development of computational efficient approaches for solving large-scale planning/scheduling examples with sequence-dependent issues (Harjunkoski et al.[17]) especially in multiproduct multistage continuous plants.

The aim of this work is to develop computationally efficient optimization models and solution approaches to improve a literature work by Liu et al.[25], where a hybrid discrete/continuous time model for the planning of multiproduct multistage plants with sequence-dependent changeovers was proposed. Three novel precedence-based reformulations for multiproduct multistage processes with sequence-dependent issues are developed here, based on the integration of the well-known Traveling Salesman Problem (TSP) formulation with the sequencing variables and constraints of the general-precedence and unit-specific general precedence representations. In addition, solution approaches integrating Rolling Horizon (RH) approach and Iterative Improvement (II) algorithms is proposed. Related works for the process industry can be found in the literature[60-68]. Finally, the solutions obtained by these reformulations are compared with those by literature models by Liu et al.[25], Kopanos et al.[57] and Aguirre et al.[58].

The remaining structure of this paper is presented as follows. In Section 2 we describe the multiproduct multistage planning and scheduling problem for continuous batch plants originally presented in Liu et al.[25]. Then, the existing and our proposed mixed integer linear programming (MILP) models are presented in Section 3 and the solution methodologies describing the principal approaches based on Rolling Horizon (RH) and Iterative Improvement (II) techniques are introduced in Section 4. Section 5 presents the details of the problem solved. Finally, the main computational results of small size and medium-scale instances are analyzed in Section 6. Concluding remarks are given at the end in Section 7.

2. Problem Statement

In this work, a planning and scheduling problem of a multistage multiproduct continuous plants with a single production unit *m* per stage *s* (see Figure 2) is addressed, same as the one investigated by Liu et al.[25]. Here, a hybrid discrete/continuous time representation is used (Figure 3), and the whole planning horizon *TH* was divided into weeks *w*. In each week, several products are processed in a continuous plant between a minimum *(LB)* and maximum processing times *(UB)* per production stage. Here it is assumed that all products must follow the same sequence of production stages where unlimited intermediate storage policies *(UIS)* are applied. It important to notice that transition times between stages are ignored, so instantaneously after is processed the material is flowed to the intermediate storage waiting for the next stage. According to this, the production starting time of product *i* (TS_{isw}) in stage *s* should greater equal than the starting time of the same product in the previous stage *s-1*, while the processing time (T_{imw}), is forced to end after the processing time of the previous stage's unit. A simple example for product *i1* in between stages *s2* and *s3* is shown in Figure 3. In here, production times (T_{imw}) are defined by the production amounts (P_{isw}), where the unit production rates (r_{mi}) and yields (n_{si}) between consecutive stages are known in advance for each product. Sequence-dependent changeover times (t_{mij}) are also considered when switching production from one product to another in the same unit *m*. Thus, the starting time of one product should be greater than the end of the previous product plus the changeover time. Due to this, and assuming that products are processed following the same production sequence with a single production unit per stage, it is only necessary to define the production sequence at each week *w* by Z_{ijw}. Changeovers between two products in consecutive weeks are also considered by ZF_{ijw}.

The sales of product *i* (SL_{ciw}) must be delivered at the end of each week to satisfy the weekly demand (D_{ciw}) of the final customer *c*. In case of unsatisfied demand, products can be delivery late, by allowing backlogs (B_{ciw}). The inventory level of final products at the end of each week is limited between a minimum (V^{min}_{iw}) and maximum (V^{max}_{iw}) bounds following local storage policy.

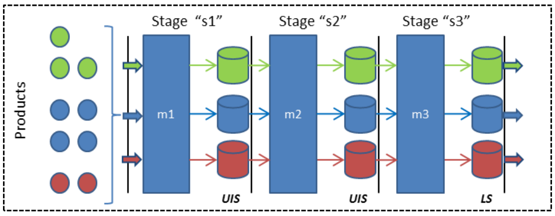


Figure 2. Multiproduct multistage single unit production scheme

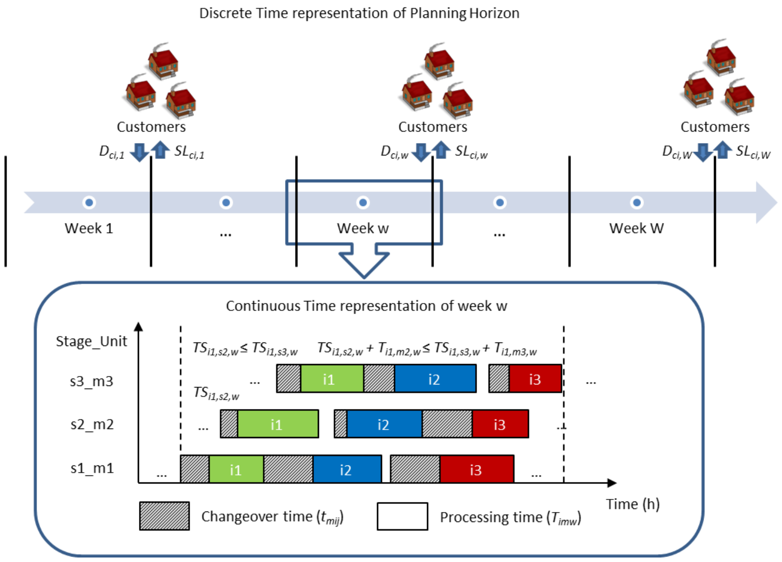

Discrete Time representation of Planning Horizon
Customers
$D_{ci,1}$
$SL_{ci,1}$
Customers
$D_{ci,w}$
$SL_{ci,w}$
Customers
$D_{ci,W}$
$SL_{ci,W}$
Week 1
...
Week w
...
Week W
Continuous Time representation of week w
Stage_Unit
$TS_{i1,s2,w} \leq TS_{i1,s3,w}$
$TS_{i1,s2,w} + T_{i1,m2,w} \leq TS_{i1,s3,w} + T_{i1,m3,w}$
s3_m3
$TS_{i1,s2,w}$
s2_m2
s1_m1
i1
i2
i3
Time (h)
Changeover time (t_{mij})
Processing time (T_{imw})

Figure 3. Discrete/Continuous time representation of the system

The problem statement of the above problem in this work is presented as follows:

Given are:

- a multiproduct multistage production process
- single unit *m* per production stage *s* in *MS*
- products *i* and customers *c*
- weeks *w* in the planning horizon *TH*
- demands from customers in each week D_{ciw}
- product prices in each week PS_{ciw}
- production rates r_{mi} and yields n_{si} of products at each stage *s*
- sequence-dependent changeover times t_{mij} and costs CC_{ijm}
- unit inventory cost CI_i and backlog penalty cost CB_{ci}

to determine:

- product assignments E_{iw}, F_{iw} & L_{iw} and production amounts P_{isw}
- production sequences Z_{ijw} & ZF_{ijw}
- production starting times TS_{isw} and processing times T_{imw}
- inventory V_{iw} and backlog levels B_{ciw}
- sales volumes SL_{ciw}

by assuming that,

- every product must follow the same production sequence throughout all stages
- transition times are not considered to transfer products between stages
- unlimited intermediate storage policies in consecutive stages are applied
- zero inventories between stages at the end of the week

so as to maximize the total profit *TP* of the system.

3. Mathematical Formulations

In this section, we introduce three different MILP-based models for the problem presented above originally introduced by Liu et al.[25]. The nomenclature and the main equations, objective function, assignment, timing, sequencing, production, backlog and inventory constraints, of the previous work in Liu et al.[25] are described in detail in [M1] in the Appendix section. It is worth emphasizing that these constraints are based in the main ideas TSP and immediate-precedence concepts. Also, two adapted versions of the models proposed by Kopanos et al.[573] and Aguirre et al.[58] based on general-precedence concepts, are presented and described in detail in [M2] and [M3] in the Appendix section.

According to the previous models, three reformulations are presented in [H1], [H2] and [H3]. The first [H1] is a reformulated version of model [M1] by using a tightening TSP. The second [H2] and the third [H3] are hybrid version of models [M2] and [M3] by using the main ideas of TSP and general-precedence concepts. Models and equations used in this paper are summarized in Table 1. A discussion about possible extensions to generalize these models to consider parallel units per production stage is described at the end of this section.

Table 1. Main full-space MILP models presented in this paper

Model	Description	Equations
[M1]	TSP/Immediate-precedence model (Liu et al.[25])	Eqs.(12-32)
[M2]	Unit-specific General Precedence model (Kopanos et al.[57])	Eqs.(12),(21-23),(26-32),(33-39)
[M3]	General precedence model (Aguirre et al.[58])	Eqs.(12),(21-23),(26-32),(40-48)
[H1]	Tightening TSP/Immediate Precedence model	Eqs.(1-3),(12-14),(21-32)
[H2]	Hybrid TSP/Unit-specific General Precedence model	Eqs.(1-6),(12-14),(21-23),(26-32)
[H3]	Hybrid TSP/General Precedence model	Eqs.(1-3),(7-14),(21-23),(26-32)

3.1. Tightening TSP/immediate-precedence reformulation [H1]

Here, we formulate a tightening version of the asymmetric TSP that the one in Liu et al.[25]. The reformulation appeared in Eqs.(1-3), is developed to reduce the number of constraints. These equations are activated for all products *i* assigned to a particular week *w* defined by E_{iw}. The immediate successor of product *i* in week *w* or between consecutive weeks are determined by Z_{ijw} and ZF_{ijw}. Also, variables F_{iw} and L_{iw} are introduced to determine if *i* is the first or the last product in each week *w*.

3.1.1. Tightening asymmetric TSP formulation

$$\sum_{j\neq i} Z_{ijw} + \sum_{j\neq i} Z_{jiw} = 2\cdot E_{iw} - L_{iw} - F_{iw} \qquad \forall i, w \leq TH \tag{1}$$

$$\sum_{j\neq i} Z_{ij,w-1} + \sum_{j} ZF_{ijw} = E_{i,w-1} \qquad \forall i, 1 < w \leq TH \tag{2}$$

$$\sum_{j\neq i} Z_{jiw} + \sum_{j} ZF_{jiw} = E_{iw} \qquad \forall i, 1 < w \leq TH \tag{3}$$

3.2. Hybrid TSP/Unit-specific general-precedence reformulation [H2]

In this section, we present a novel hybrid formulation using the main ideas of the unit-specific general-precedence developed by Kopanos et al.[57]. The reformulated version of this model is provided in [M2] in the Appendix section. The new formulation presented here is a tightening version of the one in [M2]. The new equations added are presented below in Eqs.(4-5). For this, additional binary variables X_{ijw} are introduced, following the main concepts of unit-specific general precedence formulation, for the sequencing decisions between different products *(i≠j)* in the same unit *m*.

In order maintain the same number of binary variables than in Liu et al.[25], Eq.(6) is introduced. This equation tights the domain of Z_{ijw} according with the information of the new binary variable X_{ijw}. Thus, due to the unimodularity behavior of the model, Z_{ijw} can be redefined as positive variable, by relaxing its domain, and the model could be solved without losing the global optimal solution.

3.2.1. Unit-Specific Precedence-based Constraints

$$TS_{jsw} \geq TS_{isw} + T_{imw} + t_{mij}\cdot Z_{ijw} - UB\cdot(1 - X_{ijw}) \qquad \forall i, j \neq i, s, m \in MS, w \leq TH \tag{4}$$

$$TS_{jsw} \geq \sum_{i} t_{mij}\cdot ZF_{ijw} \qquad \forall j, s, m \in MS, 1 < w \leq TH \tag{5}$$

3.2.2. Additional Tightening Constraints

$$X_{ijw} \geq Z_{ijw} \qquad \forall i, j \neq i, w \leq TH \tag{6}$$

3.3. Hybrid TSP/General precedence reformulation [H3]

The third formulation proposed in this work relies on the previous ideas of Aguirre et al.[58]. The reformulated model for this problem is presented in [M3] in the Appendix section. In this new formulation, additional binary variables X_{ijw} are proposed for the sequencing decisions of different products *(j<i)* in the same week *w* and unit *m* in production stage *s* by Eqs.(7-9). This new variable is only proposed for *(j<i)*, reducing a half the number of binary variables proposed in comparison with the previous two models presented above. In the same way that model in [H2], tightening constraints are introduced in Eqs.(10-11) to tackle with sequence-dependent issues by using variables Z_{ijw}. These equations allow solving the whole model to the optimal solution by relaxing variables Z_{ijw}.

3.3.1. General Precedence-based Constraints

$$TS_{jsw} \geq TS_{isw} + T_{imw} + t_{mij} \cdot (Z_{ijw}) - UB \cdot (3 - X_{ijw} - E_{iw} - E_{jw}) \ \forall i, j < i, s, m \in MS, w \leq TH \tag{7}$$

$$TS_{isw} \geq TS_{jsw} + T_{jmw} + t_{mji} \cdot (Z_{jiw}) - UB \cdot (X_{ijw}) - UB \cdot (2 - E_{iw} - E_{jw}) \ \forall i, j < i, s, m \in MS, w \leq TH \tag{8}$$

$$TS_{jsw} \geq \sum_i t_{mij} \cdot ZF_{ijw} \qquad \forall j, s, m \in MS, 1 < w \leq TH \tag{9}$$

3.3.2. Additional Tightening Constraints

$$X_{ijw} \geq Z_{ijw} \qquad \forall i, j < i, w \leq TH \tag{10}$$

$$1 - X_{ijw} \geq Z_{jiw} \qquad \forall i, j < i, w \leq TH \tag{11}$$

3.4. Model extension

As we explain in the introduction section, real application problems in the process industry commonly consider multiple parallel units per production stage. The MILP models presented in this section can be easily extended to consider multiple units, for certain processing structures and under some assumptions. For example, under flowshop configuration with parallel units, for a single unit assignment, model [M1] can be easily extended adding an extra index *m* in the assignment (E_{iw}, L_{iw} *&* F_{iw}) and sequencing (Z_{ijw} & ZF_{ijw}) variables and also introducing some changes in many of the constraints presented in this work. More complex changes are needed to consider job-shop configurations as in multipurpose continuous and semicontinuous plants[69-74]. Thus, mathematical approaches that consider complex production process in continuous plants, involving multiple units working in parallel, are going to be part of future developments.

4. Solution Methodology

In this section, we develop an integrated solution method for solving medium-scale instances of the problem. This method is proposed by combining an iterative solution technique based on a rolling horizon (RH) approach (Liu et al.[25]), and iterative improvement (II) algorithm (see Castro et al.[75]; Aguirre et al.[8]). The structure of the sequential solution approach is described as follow in Figure 4.

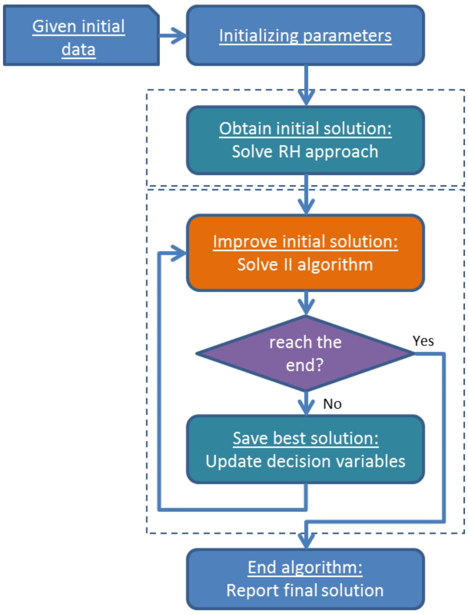


Figure 4. Sequential solution approach using RH and II methods

4.1. Rolling Horizon (RH)

This solution approach is focusing on providing a good quality solution for the whole planning horizon *(PH)* in short computational time. Given the production rate *(r_{mi})*, yields *(n_{si})*, changeover times *(t_{mij})*, and customer's demand *(D_{ciw})* and defining the planning horizon at each iteration *(TH)*, the length period *(LP)* and the fixed period *(FP)*, the algorithm starts running, in a closed loop, to obtain an initial feasible solution of the entire problem by using a RH approach (Figure 5).

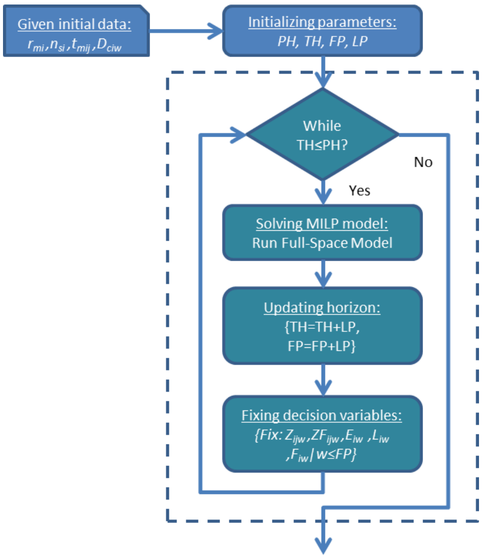


Figure 5. Closed loop of RH algorithm of our iterative procedure

RH was demonstrated to be an efficient solution algorithm for solving large-scale problems in short computational time. Thus, the entire horizon is divided into weeks and reduced sub-problems with a planning horizon of *TH* weeks are considered for each iteration (see Liu et al.[25,76]). For each sub-problem, a reduced MILP model is solved in the first *TH* weeks while fixing the integer decisions Z_{ijw}, ZF_{ijw}, E_{iw}, L_{iw}, F_{iw} in the fixed period, defined as the first *FP* weeks. Next, *TH* and *FP* are increased by a length period of *LP* weeks at this iteration and move to the next one. The whole procedure terminates when the whole planning horizon is covered. An illustrative example of RH algorithm for an 8-weeks problem horizon *(PH=8)* is shown in Figure 6, by considering an initial planning horizon of 4 weeks *(TH=4)*, a length period of 2 weeks *(LP=2)* and initial fixed period in zero *(FP=0)*. Once a planning horizon *(TH)* reaches the problem horizon *(PH)*, the algorithm ends reporting an initial feasible solution of the entire problem.

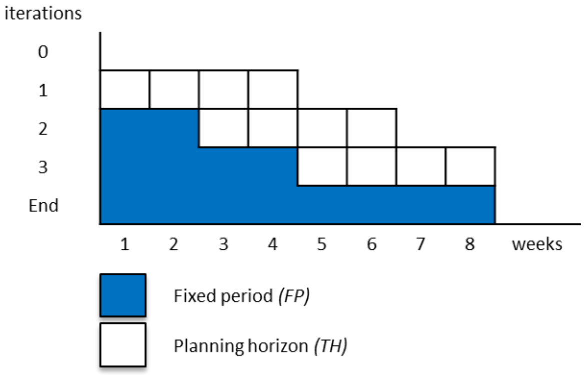


Figure 6. Illustrative example RH approach.

The final solution reported by the RH approach is considered as an approximate solution of the full problem. Also the quality of this solution depends on *TH* and *LP* values adopted. Due to iterative algorithm only knows what is happening in the following *TH* periods; it is possible to make local decisions that affect much more the final solution at the end.

In order to mitigate the effects of these local solutions, a large value of *TH* must be considered or a small value of *LP*. In the first case, larger models have to be solved at each iteration consuming much more CPU time. While in the second case, much more number of iteration should be considered, which impacts in the CPU time of the algorithm. In any of those cases, if the number of weeks of the planning problem is high, the computational time required to solve the problem could grow significantly, and also the final solution could not be good enough as is expected.

In the way to overcome these limitations, an *Iterative Improvement (II) algorithm* was proposed, to try to improve the initial solution reported by RH. The main idea behind the II algorithm relies on the possibility to decompose the whole problem in terms of products for all the weeks of the planning horizon. Thus, one or more different products for non-necessarily consecutive weeks can be rescheduled from the current solution. This rescheduling approach works in an isolated loop improving and updating decision variables until stop criteria is reached. Thus, if any other product or product/week can enhance the actual solution or if a total number of iterations are reached, the algorithm stops reporting a final result. A detailed description of the II algorithm is provided in the following section.

4.2. Iterative Improvement (II)

The II algorithm proposed in this work relies on the main ideas of Castro et al.[75] and Aguirre et al.[8] applied for the scheduling of batch plants. In order to deal with the main features of the problem, we adapt this algorithm to decompose the problem in terms of number of weeks *(N)* and products *(NOP)* (see Figure 7).

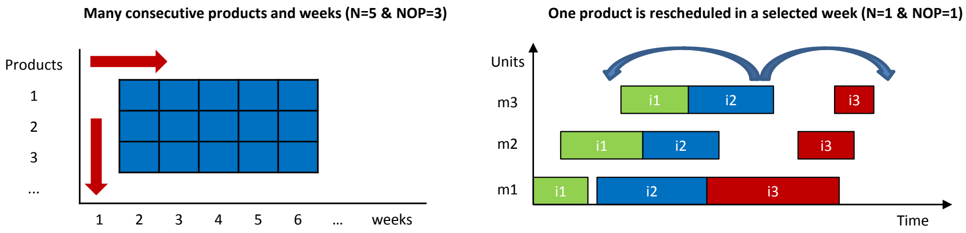

Many consecutive products and weeks (N=5 & NOP=3)
Products
1
2
3
...
1
2
3
4
5
6
...
weeks
One product is rescheduled in a selected week (N=1 & NOP=1)
Units
m3
m2
m1
i1
i2
i3
Time

Figure 7. Main decomposition techniques

According to this, different alternatives can be managed to decompose the entire problem into sub-problems. In the left side, many consecutive products can be chosen for consecutive weeks while, in the right side, only a single product is rescheduled in a selected week. The algorithm proposed in this work can deal with the selection and the rescheduling of many consecutive weeks *(N)* and products *(NOP)* in simultaneous, allowing the possibility of product crossovers and/or reassignments (see Figure 8).

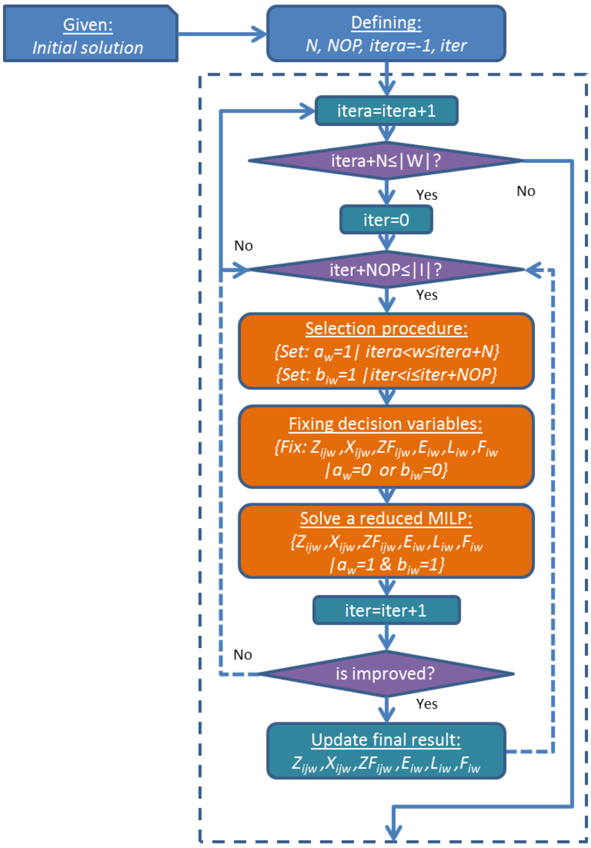

Given:
Initial solution
Defining:
N, NOP, itera=-1, iter
itera=itera+1
itera+N≤|W|?
Yes
No
iter=0
No
iter+NOP≤|I|?
Yes
Selection procedure:
{Set: a_w=1| itera<w≤itera+N}
{Set: b_{iw}=1 |iter<i≤iter+NOP}
Fixing decision variables:
{Fix: Z_{ijw} ,X_{ijw},ZF_{ijw},E_{iw},L_{iw} ,F_{iw}
|a_w=0 or b_{iw}=0}
Solve a reduced MILP:
{Z_{ijw} ,X_{ijw},ZF_{ijw},E_{iw},L_{iw} ,F_{iw}
|a_w=1 & b_{iw}=1}
iter=iter+1
No
is improved?
Yes
Update final result:
Z_{ijw} ,X_{ijw},ZF_{ijw} ,E_{iw},L_{iw} ,F_{iw}

Figure 8. General view of the Iterative Improvement algorithm (II)

Figure 8 shows a general view of the *Iterative Improvement algorithm (II)* proposed in this work. This algorithm has an outer loop and an inner loop inside. The first iterates in terms of weeks until a number of iterations *itera* reach the maximum number of weeks to reschedule. In the same way, the inner loop iterates till the maximum number of products *(iter)* are reached. After that, when the maximum number of weeks and products are reached the outer loop ends reporting this final solution. Thus, main idea of the algorithm is to define a subset of weeks and then, for the selected weeks, defines a set of consecutive products per iteration to reschedule.

In order to do that, two sets of boolean parameters a_w and b_{iw} were defined. These parameters represent together if the week w and the product i are released for rescheduling in each iteration adopting value 1. The way to set these values depends to the selection procedure chosen by the user. So, the key of the algorithm relies on the decomposition method and the selection procedure we use for the rescheduling.

Once these parameters are defined, decision variables (Z_{ijw}, ZF_{ijw}, E_{iw}, L_{iw}, F_{iw}) for $a_w=0$ *or* $b_{iw}=0$ are fixed from the current schedule and then the MILP model is solved for all decision variables that satisfy $a_w=1$ & $b_{iw}=1$ in order to find a better result. After that, if the solution is improved the result is updated with the new values of the decision variables. In case of no improvements, the values of the previous solution remain the same.

4.3. Decomposition method

As was explained in Aguirre et al.[8], the way to decompose the problem infers directly on the results and the CPU time of the algorithm. A good method can improve the solution much more and also reduce the computational effort. To deal with this complex selection issue, we just propose a simple method by releasing a single product/week tuple per iteration. Thus, starting from *w1* the algorithm selects one product to release in the next iteration, by following a lexicographic order of products. This procedure continues releasing products one-by-one until the last one, $i=|I|$, and then move to the next week $w+1$. The algorithm finishes when all products/weeks are released after $|W|\cdot|I|$ iterations. This behavior is emulated by the II algorithm described in Figure 8 for $N=1$ and $NOP=1$.

As an example, Figure 9 shows how product *i2* at week *w* is selected for rescheduling from the current schedule *(i1-i2-i3)* while sequencing variables X_{ijw} are recalculated to define the new sequence. Thus, considering that X_{ijw} is a general precedence variable, it is easy to demonstrate that even if *i2* is moved before *i1* or after *i3* and also even *i2* is removed from the schedule, only two sequencing variables ($X_{i2,i1,w}$ & $X_{i3,i2,w}$) need to be optimized by the model while the relation between *i1* and *i3* remains the same. Assignment variables of the current week *w* (E_{iw}) are also optimized by the model, allowing the possibility of adding and removing products at this weed *w* from the schedule.

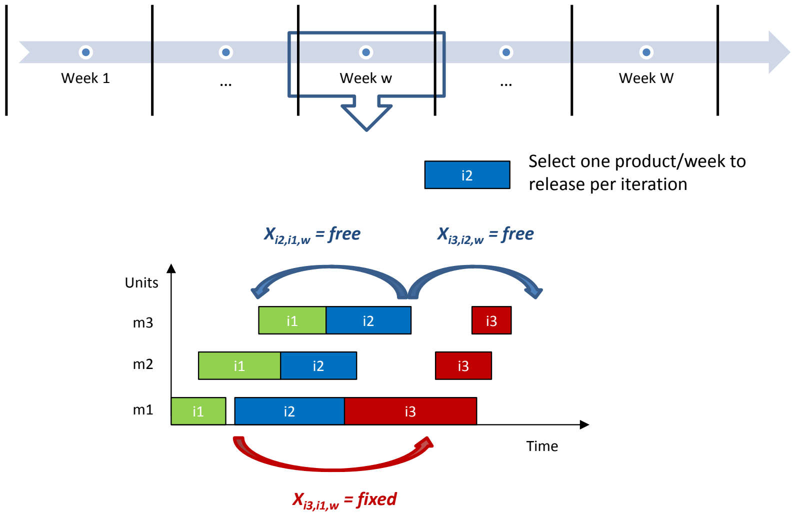


Figure 9. Decomposition method per product/week using general-precedence variables.

In Figure 9, we tried to explain the main idea of the decomposition method with a simple example. Then, additional changes could be done in order to study the behavior of different selection procedures and their impacts on the final solution. These ideas are presented by considering a general-precedence formulation using X_{ijw} for ($j < i$). For unit-specific or immediate-precedence representations using Z_{ijw}, X_{ijw} for ($j \neq$ i), the changes can be also implemented.

5. Problem instances

The problem presented here is based on the one in Liu et al.[25]. In this work we analyze the original problem with 7 products *(i1-i7)* and an extended version of 20 products *(i1-i20)*. The problem consists of a multistage continuous production plant where a set of products have to be produced within three consecutive single-unit stages *(s1-s3)* in order to satisfy the weekly demand of 10 customers *(c1-c10)* for a given planning horizon expressed in weeks. The additional data related with production rate (r_{mi}), changeover times (t_{mij}), customer's demand (D_{ciw}) and product prices (PS_{ciw}) are provided as Supporting Information file. Here it is worth noticing that all customers have the same product prices, except for customer c10 whose price is 50% higher, as in Liu et al.[25]. The unit inventory cost and the backlog cost are defined as 10% and 20% of the product prices, and the changeover cost is 10 times the changeover times defined in hours.

The selected instances of the problem are solved considering different planning horizon *PH* (weeks). All weeks have the same total available processing time defined by lower and upper bounds, *LB=0* and *UB*=168 h. For full-space models, the planning horizon is defined as the problem horizon. But, for RH approach, the planning horizon is set in 4 weeks *(TH=4)* and the length period is defined in 2 weeks *(LP=2)*. While, for II algorithm, a single product-week is selected at each iteration to be solved *(N=1 & NOP=1)*. The minimum inventory level for each week is V^{min}_{iw}=0 while the maximum is a big number V^{max}_{iw}=1000 tons and the production yield *(n_{si})* is set in 0.9 between consecutive stages *s* for the each product *i*.

In order to analyze the impact of main parameters, production rate *(r_{mi})*, yields *(n_{si})*, changeover times *(t_{mij})*, and customer's demand *(D_{ciw})*, we run many instances by changing those parameters in between +/- 20%. The results of these analyses are shown in the computational analysis and results section.

6. Computational analysis

This section presents the principal results of the problem for many instances by using different methods. First we report the results of existing full-space models [M1], [M2], [M3] and the ones proposed in this work [H1], [H2] and [H3] for a small-size problem considering 7 products and 6, 12, 18-weeks instances. Then, we extend our analysis for a medium-size example by considering 20 products and 18 weeks. For this, we also introduce in the analysis RH and our sequential solution technique RH+II based on the decomposition algorithm described above. A detailed configuration of these algorithms is presented below. After that, we compare the solution reported by the full-space model and the algorithms for a particular problem instance. All the solutions were reported by using GAMS® v24.6 with solver Gurobi® v6.5 on an Intel® Xeon® CPU 3.5GHz with12 parallel threads. The following configurations were used for the RH+II algorithm in the all runs presented later:

RH+II configuration:
First step: model [H1] is solved in each iteration until 5% of gap by using RH with *TH=4& LP=2.*
Second step: model [H3] is solved in each iteration until 1% of gap by using II with *N=1* & *NOP=1.*

6.1. Comparison of full-space MILP models for small-size problems

Statistics and results for existing models [M1], [M2], [M3], proposed model reformulations [H1], [H2], [H3] and sequential approaches RH, RH+II for 6, 12, 18 weeks are presented in Table 2. As it can be seen, models [M2] and [M3] have fewer integer variables than [M1] but at expenses of more equations and/or continuous variables. Despite of that, results reported by [M2] and [M3] cannot achieve the one reached by model [M1], in a time limit imposed of 1000 seconds, leaving in some cases an important relative gap. Thus, general-precedence based models cannot close the gap easily in comparison with model [M1] using TSP. In the other hand, proposed models in [H1], [H2] and [H3], have been able to provide good quality results within a short small relative gap which indicates the effectiveness of using a the hybrid TSP/precedence-based formulations. Thus, proposed models provide good quality solutions for small-size problems after 1000 seconds in comparison with full-space model [M1]. If we compare these results with sequential approaches, it can be seen that RH and RH+II have been able to provide good results, less than within 4% worse than of the solution by [M1] model, in only few seconds for all instances, when which is 2 to 3 orders of magnitude shorter than the full-space models spend several minutes for some of them. This behavior motivates the idea of using sequential approaches for solving larger problems, as it can be seen in the following section.

Table 2. Comparison of results of full-space models for small-size problems

MILP	Existing full-space models			Proposed full-space Models			Sequential approaches*	
	[M1]	**[M2]**	**[M3]**	**[H1]**	**[H2]**	**[H3]**	**RH**	**RH+II**
TIME HORIZON	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
EQUATIONS	2978	3070	2896	2852	2474	2474	2852	2474
VARIABLES	1506	1723	1765	1506	1758	1758	1632	1919
DISCRETE	378	294	168	378	378	252	252	91
PROFIT ($)	**5583**	**5583**	**5490**	**5583**	**5583**	**5583**	**5504**	**5530**
CPU time (sec.)	42	1000(1)	1000(1)	42	129	46	2	13
GAP (%)	0%	8.82%	10.8%	0%	0%	0%	-	-

IP (%)	**-**	**0%**	**-1.7%**	**0%**	**0%**	**0%**	**-1.4%**	**-0,9%**
TIME HORIZON	**12**	**12**	**12**	**12**	**12**	**12**	**12**	**12**
EQUATIONS	6116	6202	5854	5864	4982	4982	5864	4982
VARIABLES	3060	3487	3571	3060	3564	3564	3564	3893
DISCRETE	756	588	336	756	756	504	252	175
PROFIT ($)	**10619**	**10502**	**10302**	**10619**	**10608**	**10607**	**10199**	**10449**
CPU time (sec.)	1000(1)	1000(1)	1000(1)	1000(1)	1000(1)	1000(1)	4	16
GAP (%)	0.32%	10.4%	12.5%	0.34%	0.58%	0.57%	-	-
IP (%)	**-**	**-0.9%**	**-2.9%**	**0%**	**-0.1%**	**-0.1%**	**-3.9%**	**-1.6%**
TIME HORIZON	**18**	**18**	**18**	**18**	**18**	**18**	**18**	**18**
EQUATIONS	9254	9334	8812	8876	7490	7490	8876	7490
VARIABLES	4614	5251	5377	4614	5370	5370	5496	5867
DISCRETE	1134	882	504	1134	1134	756	252	259
PROFIT ($)	**16498**	**16240**	**16059**	**16521**	**16511**	**16512**	**16011**	**16267**
CPU time (sec.)	1000(1)	1000(1)	1000(1)	1000(1)	1000(1)	1000(1)	5	36
GAP (%)	1.15%	10.5%	11.8%	0.93%	1.00%	1.11%	-	-
IP (%)	**-**	**-1.6%**	**-2.7%**	**0.14%**	**0.08%**	**0.08%**	**-2.9%**	**-1.4%**

(1) Termination Criteria = if any solution improve the objective function or time limit imposed of 1000 sec. %GAP = relative gap. %IP=improvement percent in comparison with model [M1]. *Statistics for the last iteration.

6.2. Comparison of different solution approaches for a medium-size problem

Table 3 shows the comparisons between existing full-space models, proposed hybrid models and the rolling horizon approach and our sequential algorithm developed in this work for medium-size problems. For a fair comparison we evaluate the solution methods imposing a time limit of 500, 1000 and 3600 seconds.

Table 3. Comparison of results of different solution approaches for a medium-scale problem

MILP	Existing full-space models			Proposed full-space Models			Sequential approaches*	
	[M1]	**[M2]**	**[M3]**	**[H1]**	**[H2]**	**[H3]**	**RH**	**RH+II**
TIME HORIZON	**18**	**18**	**18**	**18**	**18**	**18**	**18**	**18**
EQUATIONS	53571	59033	52895	52491	39951	39951	52491	39951
VARIABLES	17601	24101	24461	17601	25041	24441	23761	28881
DISCRETE	7920	7200	3780	7920	7920	4500	1760	60
PROFIT ($)	**5037**	**3329**	**2169**	**5944**	**6721**	**6402**	**6221**	**6972**
CPU time (sec.)	500(1)	500(1)	500(1)	500(1)	500(1)	500(1)	124	385
GAP (%)	45%	71%	81%	30%	24%	26%	-	-
IP (%)	**-**	**-34%**	**-56%**	**18%**	**33%**	**27%**	**24%**	**38%**
PROFIT ($)	**5037**	**3329**	**2897**	**5944**	**6721**	**6402**	**6221**	**6972**
CPU time (sec.)	1000(2)	1000(2)	1000(2)	1000(2)	1000(2)	1000(2)	124	385
GAP (%)	41%	71%	75%	30%	21%	25%	-	-
IP (%)	**-**	**-34%**	**-42%**	**18%**	**33%**	**27%**	**24%**	**38%**
PROFIT ($)	**5037**	**3329**	**4057**	**5944**	**6721**	**6402**	**6221**	**6972**
CPU time (sec.)	3600(3)	3600(3)	3600(3)	3600(3)	3600(3)	3600(3)	124	385
GAP (%)	40%	71%	65%	29%	20%	24%	-	-
IP (%)	**-**	**-34%**	**-20%**	**18%**	**33%**	**27%**	**24%**	**38%**

Termination Criteria = time limit imposed of (1) 500, (2) 1000, (3) 3600 seconds of CPU time. %GAP = relative gap. %IP=improvement percent in comparison with model [M1]. *Statistics for the last iteration.

The results reported by existing full-space models [M1], [M2], [M3] after 1000s are far away from the best solution found for this problem ($ 6972) and the relaxation of these models are very weak, leaving a huge relative gap, 41%, 71% and 75% respectively. The proposed hybrid models in [H1], [H2], [H3], can improve (IP) the solution found by [M1] in 18%, 33% and 27% after 1000s but the relative gap (GAP) is still in between 24%-30%, which denotes a very bad relaxation of the models. As it can be seen, the three investigated CPU time limits do not really affect the final solution reported. In most of the cases, the solution found after 500 seconds remains the same after 1 hour, and the relative gap is reduced in at most 5% except [M3] whose solution was significantly improved. These results trend to demonstrate the bad relaxation of the full-space models for this problem and the necessity to apply decomposition techniques to obtain better solutions in less computational time. Thus, results for this problem indicate that our sequential approach RH+II was able to find the best final solution after 385s enhancing the one reported by [M1] in more than 38% (%IP) obtaining also in between 6%-20% more improvements than proposed full-space models [H1], [H2] and [H3] in a half of CPU time. The initial solution of the RH+II method was provided by RH in only 124s, obtaining a total improvement of 24% in comparison with [M1]. Then, the II algorithm was able to enhance the initial solution iteratively until the last iteration, achieving 14% of improvement after 261s reporting the best final result for this problem.

This particular example allows us to demonstrate the weakest point of full-space models when larger problems are trying to be solved in a monolithic way. That is why it is necessary to introduce efficient sequential techniques to decompose the large instances and solve small reduced sub problems in tractable CPU time.

6.3. Sensitivity analysis

In this section, we study the individual effect of the variation of different parameters (a) Demand, (b) Changeover, (c) Rate, (d) Yield in the performance of the models by analyzing the trade-off between Profit ($), CPU time and the number of feasible solutions (FS). The details of the main results reported by different approaches are summarized at the end in Figure 10. These bar-charts represent the key performance indicators for each analyzed model. Here, each measure is normalized between [0,1], where the worse performance is 0 and the best is 1.

For a fair comparison, we developed a ranking system which gives more priority to the Profit and using CPU time and FS as tiebreakers. This ranking gives the higher value (1-8) to a better model in terms of the normalized Profit. If two models have the same Profit then the CPU performance is used to determine the ranking. In case the results remain the same, the one with more FS is chosen. The rankings of the approaches for different analysis, (a) Demand, (b) Changeover, (c) Rate, (d) Yield, are shown in Figure 11. The details of the results in the following sub-sections are available in the supporting information file (Tables S8-S11).

*6.3.1. **Effect of variation in demand forecast***

In this section, we analyze the effect of +/-20% of demand variation in the total profit (Profit), the computational time (CPU time) and the solution performance (FS) of different solution approaches in Figure 10 a). Detailed information of the results provided by these methods for 10 random scenarios are reported in the Supporting information file. The ranking of these approaches is summarized in Figure 11a).

- ***Profit:*** the best average solution was found by the sequential approach RH+II ($6740) while the worse average was found by model [M3] ($2409). Solution reported by RH+II is 2% better than the one reported by model [H2], 7% better than the RH approach, 13% better than [H3] and more than 20% in comparison with the rest of the models including [M1].

- ***CPU time:*** RH method could report the best average CPU time (90s), followed by RH+II (28% of the best average = 345s). All the other methods report an average value of 1000 seconds.

- ***Feasibility:*** except for model [H3], all the other methods could find a feasible solution for the instances analyzed. Model [H3] only could find feasible solutions in 40% of the cases analyzed due to the time limit imposed.

- ***Ranking:*** the best solution approach for the demand variation was RH+II, followed by [H2], RH, [H3], [M1] and [H1].

*6.3.2. **Effect of variation in changeover times***

The effect of the changeover times is represented by Figure 10b), showing the impact of Profit ($) and the CPU time (s) and feasible solutions FS for all the methods analyzed. Figure 11b) shows the ranking of these approaches for the instances analyzed.

- ***Profit:*** our sequential algorithm RH+II could have an average Profit ($6895) which was 6% better than the one reported by RH method ($6460). This solution is comparable with the one reported by [H2] model for all cases analyzed. In this case, proposed full-space models [H1], [H2], [H3], behave better than existing model [M1] with enhanced results of 4%, 16% and 10%.

- ***CPU time:*** the best average computational time was reported by RH method (77s). Our proposed RH+II reported a final time which was 30% worse (356s) than RH method. The computational time spent by model full-space models was many times the one reported in RH.

- ***Feasibility:*** models [M1] and [H2] could find feasible solutions in 90% of the cases while model [H3] only in 20% of the cases. The rest of the models found feasible solutions for all the instances run.

- ***Ranking:*** Figure 11b) shows the superior performance of RH+II and RH methods in comparison with proposed models, [H2], [H3], [H1] and existing models [M1], [M2], [M3].

6.3.3. ***Effect of variation in production rate***

In Figure 10c), we studied the behavior of the system by introducing +/-20% variation in production rate. Figure 11c) is provided to show the ranking between the solution approaches.

- ***Profit:*** full-space models [H2] provided the best average Profit ($4801) for the instances solved. In this case, RH and RH+II could provide good-quality results, 85% and 94% of the best average solution reported by [H2] while models [H3] and [M1] could find a final solutions which was similar to the RH method.

- ***CPU time:*** full-space models provided good quality results but only in some cases after a CPU time of 1000 seconds. The average computational time for RH was 183s while RH+II was 453s.

- ***Feasibility:*** the numbers of feasible solutions were crucial in this case. Only 30% of the cases were solved by [M1], [M3], 50% by [M2], 70% by [H2], [H3] and 80% by [H1]. Sequential approaches could find feasible solutions for all instances.

- ***Ranking:*** the ranking demonstrated that RH+II and RH were promising approaches always reaching feasible solutions in short CPU time, while proposed models [H1], [H2], [H3] could also be considered as possible alternative solution methods for this particular case.

6.3.4. ***Effect of variation in production yield***

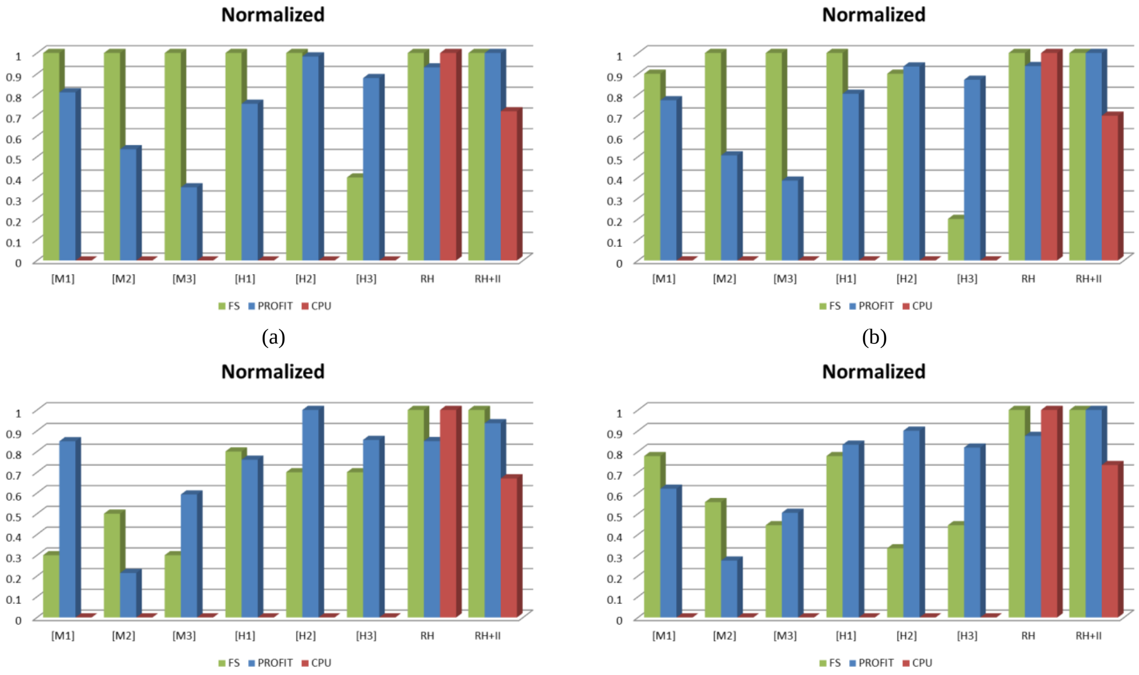
A similar behavior as the production rate occurs when we change the production yields in +/-20%. The results of the system are summarized in Figures 10d). Solution ranking is shown at the end in Figure 11d).

- ***Profit:*** the average Profit reported by full-space models [H1], [H2], [H3] were quite good, improving in more than 20% the one reported by existing models [M1], [M2] and [M3] for the feasible cases. RH obtained good quality results, similar to the one reported by proposed models. This solution was improved by RH+II, reaching more than 13% better average Profit ($5354).

- ***CPU time:*** RH used only the 20% of the CPU time needed by proposed full-space models (1000s) to obtain similar results. The computational time reported by RH was 75s while RH+II was only 138s.

- ***Feasibility:*** the numbers of feasible solutions reached by full-space models were very poor. Models [M1] and [H1] provide feasible solutions in 80% of the cases, while models [M2],[M3], [H2] and [H3] only could find results in 60%, 40%, 30% and 40% of the cases.

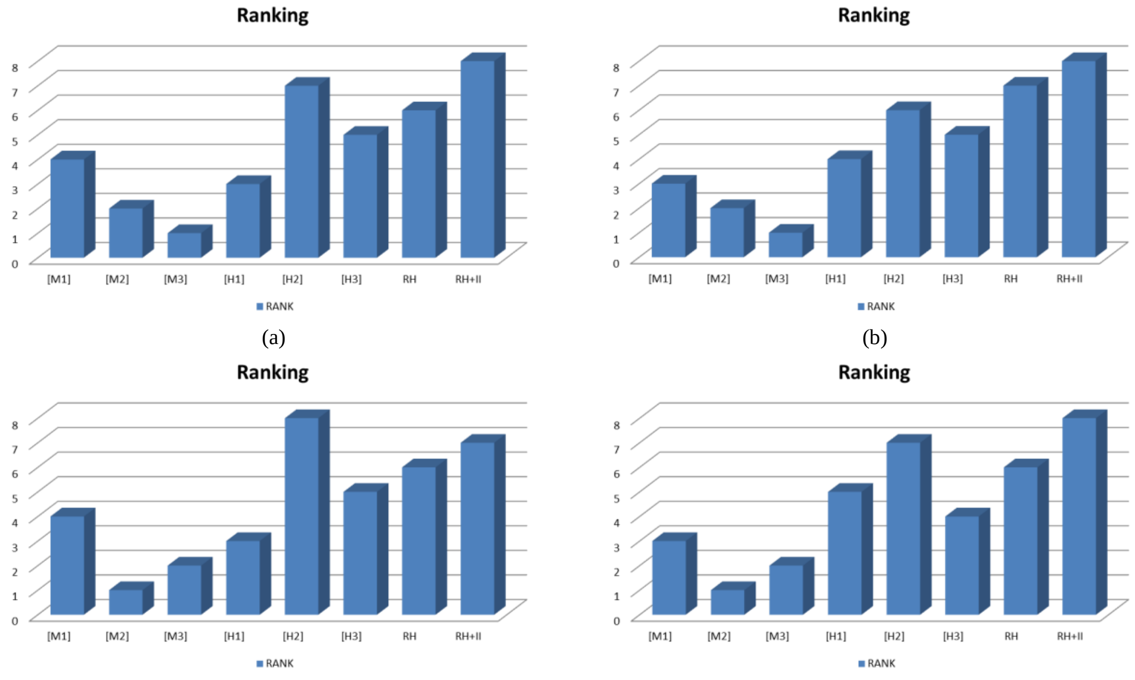
- ***Ranking:*** Figure 11d) demonstrates the superior performance of sequential RH+II followed by RH method and proposed full-space models [H2], [H1] and [H3].



(a) (b)

(c) (d)

Figure 10. Sensitivity analysis for (a) demand, (b) changeover, (c) rate and (d) yield variation



(a) (b)

(c) (d)

Figure 11. Ranking for (a) demand, (b) changeover, (c) rate and (d) yield variation

6.4. Experimental analysis

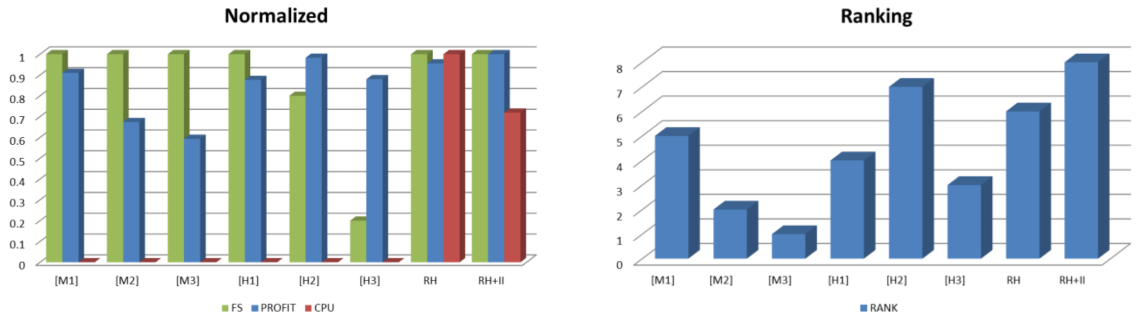
In order to study the real behavior of the algorithms, ten different instances of the problem presented above are created and evaluated with the same solution approaches. These instances are generated by changing the production rate and changeover times in a uniform way between their minimum and maximum values. Thus, ten new feasible instances were proposed for the comparison. The results and comparisons of different approaches for proposed problems are presented in the Supporting Information file (see Table S12 and Figure 12a). Finally, given the results we shown a normalized ranking (Figure 12b), of the best approaches tested for this particular problem from the worse to the best, by analyzing the Profit ($) vs. CPU time (s) and feasible solutions FS(#).

- ***Profit:*** Figure 12a) demonstrates that the average solution provided by proposed model [H2] was more than 8% better than other full-space models while the solutions of models [H1] and [H3] were very similar than the average reported by [M1].

- ***CPU time:*** in comparison with sequential approaches, RH always provides a feasible solution in average computational time of 100 seconds. This algorithm behaved better than [M1] in average, providing good quality results. After the RH, the II algorithm enhanced the solution in about 5% within a reasonable average CPU time (342s). Solutions provided by RH+II were 10% better in average than [M1] and 2% in comparison with [H2].

- ***Feasibility:*** it is worth emphasizing that existing models [M1], [M2], [M3] and model [H1] could provide more feasible solutions in comparison with [H2] and [H3] that only reported solutions in 80% and 20% of the cases respectively. Besides of this, the quality of the solutions provided by [M2] and [M3] were more than 20% worse than the one reported by model [M1].

- ***Ranking:*** Figure 12b) indicates the superior performance of the sequential approach RH+II and full-space model [H2] in comparison with other existing and proposed approaches in this work.



(a) (b)

Figure 12. a) Sensitivity and b) Ranking analysis of many approaches for the instances solved.

7. Discussion

Three different hybrid models [H1], [H2], [H3] based on TSP and precedence-based formulations and a sequential algorithm RH+II by using a RH and iterative improvement (II) methods were proposed in this work to solve medium-scale planning and scheduling problems with sequence-dependent changeovers in a multiproduct multistage continuous plant in an efficient way in comparison with the existing approaches in the literature based on a previous contribution of Liu et al.[25] here identified by model [M1] and also other contributions [M2] and [M3] by Kopanos et al.[57] and Aguirre et al.[58] respectively.

Results show that, small size problems can be easily tackle by proposed models [H1], [H2], [H3] and existing model [M1], providing a good quality solution for all cases within 2% of relative gap, when existing models [M2], [M3] only could find a solution within 8%-12% gap after 1000 seconds. This indicates that hybrid models can close the relative gap very well, reaching also better integer results than other existing formulations based on precedence-based representation. Here it is important to emphasize that despite of models [M2] and [M3] have less binary variables, they have more continuous variables and big-M constraints that difficult the convergences of the model in a reasonable computational time. In this sense the hybridization of TSP and precedence-based formulations takes the advantages, allowing finding better results than using solely precedence-based models within a reasonable computational time. Beside of that, is still hard to find a global optimal solutions for small size problems by using the existing and proposed models in this work. One of the main weakness points of these models relies on the use of a huge number of big-M constraints and integer variables that difficult the convergence of the model providing a weak initial relaxation. In consequence, is still clear the necessity of developing efficient robust continuous formulations that can be used for solving planning and scheduling problems within considering sequence-dependent issues in a reasonable computational time.

For medium-size problems, we start analyzing the behavior of full-space models [M1], [M2], [M3], [H1], [H2] and [H3]. Comparing the solutions with model [M1], we can identify that proposed models [H1], [H2], [H3] report significant improvements, while existing models [M2] and [M3], only can obtain feasible results without any improvement. Despite of this, it is worth remarking that the relative gap of the solutions provided by the full-space models is always greater than 15% after 1000 seconds of CPU time which indicates the necessity of new methods for solving large-scale problems.

In order to improve the solutions obtained by the full-space models, we introduce a RH and II algorithms. The solutions obtained by RH compete very well with the one reported by model [M1] and also with the proposed models [H1], [H2] and [H3], and the computational time was reduced many times. This is one of the main advantages of using RH approach in order to solve very big problems with many products, weeks and stages in continuous plants.

According to this and in order to take the advantage of the fast and good quality solutions reported by RH we implement an II algorithm. This tailored algorithm starts from the solution provided by RH and aims to find an enhanced solution in short CPU time. The solution reported by RH+II demonstrates a total improvement of 5%-15% in comparison with RH method for many cases analyzed, and a significant improvement in comparison with full-space models.

After sensitivity analysis and experiment analysis, we could strongly demonstrate that even changing the demand, the changeovers and the production rate or yields, the RH+II algorithms behave always in an efficient way, better than RH and full-space models, providing good-quality solutions with a reduced CPU time. Also we could emphasize that proposed models in [H1], [H2], [H3], and more specially model [H2], can compete very well with RH and existing model [M1] providing even improved results.

Despite of these good-quality results, it is worth to emphasize that it is impossible to ensure global optimality with this method and only local optimal solutions can be found after a specific number of iterations or after a certain amount of time. Other important limitation is that this algorithm is based on initial parameters that affect the solutions quality of the algorithm. For example, how to decide the sequence of orders or weeks to release at each iteration maybe represents one of the most important decisions to make by the algorithm, and any improvement in this particular procedure will affect the direction of the search and also the solution quality.

8. Concluding remarks

An efficient solution approach combining RH+II algorithm and full-space MILP models have been developed for solving industrial-scale planning and scheduling problems in a multiproduct multistage continuous plant. The main contribution relies not only in the solution approach proposed by mixing RH and II algorithms, but also in the MILP formulations developed in [H1], [H2] and [H3] to tackle this complex combinatorial problem. The possibility to have new formulations that handle sequence-dependent changeover issues in an efficient way, using less binary variables and constraints, allow us to decompose the problem in an effective way, reducing the size of the sub-problems and solving medium size instances with reasonable computational effort. After solving many cases, with different configuration of model's parameters, we could demonstrate the effectiveness and the robustness of the solution approach presented in this work.

Future work relies in the possibility to extend these models and approaches to different areas in which sequence-dependent issues play a key role in the resolution of the problem. In addition, these models can be further extended to consider multiple parallel units per production stage, as it is common in real applications for the process industry, and also can be used for solving complex production processes, as flow-shop or job-shop structures in continuous/semicontinuous plants, by considering sequence-dependent changeovers and/or transferring times and many other interesting features, e.g. resources, maintenance operations, units degradation etc.

Supporting Information

This file contents all data related with the case studies in Tables S1-S7 and detailed result of the instances solved in this work for different cases in Tables S8-S12.

Acknowledgements

The authors gratefully acknowledge financial support from the UK Engineering and Physical Sciences Research Council (EPSRC) under project EP/M027856/1.

Nomenclature

Indexes

— i, j, k	*products*
— m	*units*
— s	*stages*
— w	*weeks*
— c	*customers*

Parameters

— PS_{ciw}	*selling prices of product i for customer c at week w ($)*
— CC_{ijm}	*changeover cost for different products i,j at unit m ($)*
— CB_{ci}	*backlog cost of product i for customer c ($)*
— CI_{i}	*inventory cost of product i ($)*
— D_{ciw}	*demand of product i at week w for customer c (tons)*
— r_{mi}	*production rate of product i at unit m (tons/week)*
— n_{si}	*production yields of product i at stage s*
— t_{mij}	*changeover times between products i,j at unit m (hours)*
— $V^{min}{}_{iw}$	*Minimum inventory level of product i at week w (tons)*
— $V^{max}{}_{iw}$	*Maximum inventory level of product i at week w (tons)*
— UB	*upper bound of production time (hours)*
— LB	*lower bound of production time (hours)*
— BM	*big-M value defined by $\lvert I \rvert$*
— MS	*boolean parameter that is 1 if unit m belongs to stage s*
— a_{w}	*boolean parameter that is 1 if week w is selected for rescheduling*
— b_{iw}	*boolean parameter that is 1 if product i at week w is the selected for rescheduling*

Binary variables

— F_{iw}	*1 if product i is the first one processed in week w*
— L_{iw}	*1 if product i is the last one processed in week w*
— E_{iw}	*1 if product i is processed in week w*
— Z_{ijw}	*1 if product i is processed immediate before product j in week w*
— X_{ijw}	*1 if product i is processed before product j in week w*

Positive Variables

— ZF_{ijw}	*1 if product i in week w-1 is processed immediate before product j in week w*
— O_{iw}	*production index of product i in week w*
— T_{imw}	*production time of product i at unit m on week w (hours)*
— P_{isw}	*production amount of product i at stage s in week w (tons)*
— V_{iw}	*inventory of product i in week w (tons)*
— SL_{ciw}	*sales of product i to customer c in week w (tons)*
— B_{ciw}	*backlog of product i to customer c in week w (tons)*
— TS_{isw}	*starting time of product i at stage s in week w (hours)*

Free Variables

— TP	*total profit ($)*

References

(1) 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.

(2) Stefansson, H., Sigmarsdottir, S., Jensson, P., & Shah, N. (2011). Discrete and continuous time representations and mathematical models for large production scheduling problems: A case study from the pharmaceutical industry. *Eur. J. Oper. Res.,* 215, 383-392.

(3) Calfa, B. A., Agarwal, A., Grossmann, I. E., & Wassick, J. M. (2013). Hybrid bilevel-lagrangean decomposition scheme for the integration of planning and scheduling of a network of batch plants. *Ind. Eng. Chem. Res.,* 52, 2152-2167.

(4) Costa, A. (2015). Hybrid genetic optimization for solving the batch-scheduling problem in a pharmaceutical industry. *Comput. Ind. Eng.,* 79, 130-147.

(5) Vieira, M., Pinto-Varela, T., Moniz, S., Barbosa-Póvoa, A. P., & Papageorgiou, L. G. (2016). Optimal planning and campaign scheduling of biopharmaceutical processes using a continuous-time formulation. *Comput. Chem. Eng.,* 91, 422-444.

(6) Gomes, M. C., Barbosa-Póvoa, A. P., & Novais, A. Q. (2013). Reactive scheduling in a make-to-order flexible job shop with re-entrant process and assembly: a mathematical programming approach. *Int. J. Prod. Res.,* 51, 5120-5141.

(7) Sukkerd, W., & Wuttipornpun, T. (2016). Hybrid genetic algorithm and tabu search for finite capacity material requirement planning system in flexible flow shop with assembly operations. *Comput. Ind. Eng.,* 97, 157-169.

(8) Aguirre, A. M., Méndez, C. A., & Castro, P. M. (2014). A hybrid scheduling approach for automated flowshops with material handling and time constraints. *Int. J. Prod. Res.,* 52, 2788-2806.

(9) Jin, L., Tang, Q., Zhang, C., Shao, X., & Tian, G. (2016). More MILP models for integrated process planning and scheduling. *Int. J. Prod. Res.,* 1-16.

(10) Kim, S. H., & Lee, Y. H. (2016). Synchronized production planning and scheduling in semiconductor fabrication. *Comput. Ind. Eng.,* 96, 72-85.

(11) Boysen, N., & Stephan, K. (2016). A survey on single crane scheduling in automated storage/retrieval systems. *Eur. J. Oper. Res.,* 254, 691-704.

(12) Pinto, J. M., & Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of process systems. *Ann. Oper. Res.,* 81, 433-466.

(13) Shapiro, J. F. (1993). Mathematical programming models and methods for production planning and scheduling. *Handbooks in Operations Research and Management Science*, 4, 371-443.

(14) Kallrath, J. (2002). Planning and scheduling in the process industry. *OR spectrum,* 24, 219-250.

(15) Floudas, C. A., & Lin, X. (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Ann. Oper. Res.,* 139, 131-162.

(16) 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.

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

(18) Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res., 246, 345-378.*

(19) Allahverdi, A. (2016). A Survey of Scheduling Problems with No-Wait in Process. *Eur. J. Oper. Res.,* 255, 665-686.

(20) Shaik, M. A., & Floudas, C. A. (2007). Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements. *Ind. Eng. Chem. Res., 46,* 1764-1779.

(21) Papageorgiou, L. G., & Pantelides, C. C. (1996). Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation. *Ind. Eng. Chem. Res.,* 35, 488-509.

(22) Chen, P., Papageorgiou, L. G., & Pinto, J. M. (2008). Medium-term planning of single-stage single-unit multiproduct plants using a hybrid discrete/continuous-time MILP model. *Ind. Eng. Chem. Res.,* 47, 1925-1934.

(23) Erdirik-Dogan, M., & Grossmann, I. E. (2008). Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines. *Comput. Chem. Eng.,* 32, 2664-2683.

(24) Liu, S., Pinto, J. M., & Papageorgiou, L. G. (2008). A TSP-based MILP model for medium-term planning of single-stage continuous multiproduct plants. *Ind. Eng. Chem. Res.,* 47, 7733-7743.

(25) Liu, S., Pinto, J. M., & Papageorgiou, L. G. (2009). Medium-term planning of multistage multiproduct continuous plants using mixed integer optimisation. *In 19th European Symposium on Computer Aided Process Engineering: ESCAPE-19: June 14-17, 2009, Cracow, Poland (Vol. 26, p. 393). Elsevier.*

(26) Pinto, J. M., & Grossmann, I. E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput. Chem. Eng.,* 18, 797-816.

(27) Pinto, J. M., & Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.,* 34, 3037-3051.

(28) Gupta, S., & Karimi, I. A. (2003). An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Ind. Eng. Chem. Res.*, 42, 2365-2380.

(29) Mendez, C. A., & Cerda, J. (2002). An efficient MILP continuous-time formulation for short-term scheduling of multiproduct continuous facilities. *Comput. Chem. Eng.*, 26, 687-695.

(30) Alle, A., & Pinto, J. M. (2002). Mixed-integer programming models for the scheduling and operational optimization of multiproduct continuous plants. *Ind. Eng. Chem. Res.,* 4, 2689-2704.

(31) Alle, A., Papageorgiou, L. G., & Pinto, J. M. (2004a). A mathematical programming approach for cyclic production and cleaning scheduling of multistage continuous plants. *Comput. Chem. Eng.,* 28, 3-15.

(32) Alle, A., & Pinto, J. M. (2004b). Global optimization for the cyclic scheduling and operation of multistage continuous plants. *Ind. Eng. Chem. Res.*, 43, 1485-1498.

(33) Munawar, S. A., & Gudi, R. D. (2005). A multilevel, control-theoretic framework for integration of planning, scheduling, and rescheduling. *Ind. Eng. Chem. Res.*, 44, 4001-4021.

(34) Bhattacharya, S., & Bose, S. K. (2007). Mathematical model for scheduling operations in cascaded continuous processing units. *Eur. J. Oper. Res.,* 182, 1-14.

(35) Bose, S. K., & Bhattacharya, S. (2008). A two pass heuristic algorithm for scheduling 'blocked out' units in continuous process industry. *Ann. Oper Res.,* 159, 293-313.

(36) Bose, S. K., & Bhattacharya, S. (2009). A State Task Network model for scheduling operations in cascaded continuous processing units. *Comput. Chem. Eng.,* 33, 287-295.

(37) Geiger, C. D., Kempf, K. G., & Uzsoy, R. (1997). A tabu search approach to scheduling an automated wet etch station. *J. of Manuf. Syst.,* 16, 102-116.

(38) Bhushan, S., Karimi, I.A. (2003). An MILP approach to automated wet-etch scheduling. *Ind. Eng. Chem. Res.,* 42, 1391-1399.

(39) Bhushan, S., & Karimi, I. A. (2004). Heuristic algorithms for scheduling an automated wet-etch station. *Comput. Chem. Eng.,* 28, 363-379.

(40) Karimi, I. A., Tan, Z. Y., & Bhushan, S. (2004). An improved formulation for scheduling an automated wet-etch station. *Comput. Chem. Eng.,* 29, 217-224.

(41) Aguirre, A. M., Méndez, C. A., Castro, P. M. (2011). A novel optimization method to automated wet-etch station scheduling in Semiconductor Manufacturing Systems, *Comput. Chem. Eng.*, 35, 2960-2972.

(42) Aguirre, A. M., Méndez, C. A., Castro, P. M., De Prada, C. (2012). MILP-based approach for the scheduling of Automated Manufacturing System with sequence-dependent transferring times. *Comput. Aided Chem. Eng.,* 30, 477-481.

(43) Castro, P.M., Aguirre, A.M., Zeballos, L.J., Mendez, C.A. (2011), Hybrid mathematical programming discrete-event simulation approach for large-scale scheduling problems, *Ind. Eng. Chem. Res.,* 50, 10665-10680.

(44) Castro, P. M., Zeballos, L. J., & Méndez, C. A. (2012). Hybrid time slots sequencing model for a class of scheduling problems. *AIChE J.,* 58, 789-800.

(45) Zeballos, L. J., Castro, P.M., Mendez, C.A. (2011). Integrated Constraint Programming Scheduling Approach for Automated Wet-Etch Stations in Semiconductor Manufacturing, *Ind. Eng. Chem. Res.*, 50, 1705-1715.

(46) Novas, J. M., & Henning, G. P. (2012). A comprehensive constraint programming approach for the rolling horizon-based scheduling of automated wet-etch stations. *Comput. Chem. Eng.,* 42, 189-205.

(47) Phillips, L.W., Unger, P.S., (1976). Mathematical programming solution of a hoist scheduling program, *AIIE Transactions*, 8, 219-225.

(48) Shapiro, G.W., Nuttle, H.L. (1988). Hoist Scheduling for a PCB electroplating facility, *IEE Transactions*, 20, 157-167.

(49) Crama, Y., (1997). Combinatorial optimization models for production scheduling in automated manufacturing systems. *Eur. J. Oper. Res.,* 99, 136-153.

(50) Paul, H. J., Bierwirth, C., & Kopfer, H. (2007). A heuristic scheduling procedure for multi-item hoist production lines. *Int. J. of Prod. Econ.*, 105, 54-69.

(51) Lu, C., Gao, L., Li, X., Pan, Q., & Wang, Q. (2017) Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. of Clean. Prod.*, 144, 228-238.

(52) Rahman, H.F., Sarker, R., & Essam, D. (2015) A real-time order acceptance and scheduling approach for permutation flow shop problems. *Eur. J. Oper. Res.,* 247, 488-503.

(53) Chang, P.-C., Lin, Y.-K., & Chen, J.C. (2015) A fuzzy-based assessment procedure for a clothing factory with waste-prevention consideration. *J. of Clean. Prod.,* 108, 484-493.

(54) Gholami-Zanjani, S.M., Hakimifar, M., Nazemi, N., & Jolai, F. (2016) Robust and fuzzy optimisation models for a flow shop scheduling problem with sequence dependent setup times: A real case study on a PCB assembly company. *Int. J. Comput. Integr. Manuf.,* 1-12.

(55) Cerdá, J., Henning, G. P., & Grossmann, I. E. (1997). A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Ind. Eng. Chem. Res.*, 36, 1695-1707.

(56) Marchetti, P. A., & Cerdá, J. (2009). A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers. *Comput. Chem. Eng.*, 33, 871-886.

(57) Kopanos, G. M., Laínez, J. M., & Puigjaner, L. (2009). An efficient mixed-integer linear programming scheduling framework for addressing sequence-dependent setup issues in batch plants. *Ind. Eng. Chem. Res.,* 48, 6346-6357.

(58) Aguirre, A. M., Méndez, C. A., Castro, P. M., & De Prada, C. (2012). MILP-based Approach for the Scheduling of Automated Manufacturing System with Sequence-Dependent transferring times. *Comput. Aided Chem. Eng.,* 30, 477-481.

(59) Cóccola, M. E., Cafaro, V. G., Méndez, C. A., & Cafaro, D. C. (2014). Enhancing the General Precedence Approach for Industrial Scheduling Problems with Sequence-Dependent Issues. *Ind. Eng. Chem. Res.,* 53, 17092-17097.

(60) Castro, P. M., Harjunkoski, I., & Grossmann, I. E. (2011). Greedy algorithm for scheduling batch plants with sequence-dependent changeovers. *AIChE J*., 57, 373-387.

(61) Baumann, P., & Trautmann, N. (2013). A continuous-time MILP model for short-term scheduling of make-and-pack production processes. *Int. J. Prod. Res.,* 51, 1707-1727.

(62) Baumann, P., & Trautmann, N. (2014). A hybrid method for large-scale short-term scheduling of make-and-pack production processes. *Eur. J. Oper. Res.,* 236, 718-735.

(63) Yun, Y., Chung, H., & Moon, C. (2013). Hybrid genetic algorithm approach for precedence-constrained sequencing problem. *Comput. Ind. Eng.,* 65, 137-147.

(64) Li, Q., & Milne, R. J. (2014). A production scheduling problem with sequence-dependent changeover costs. *Int. J. Prod. Res.,* 52, 4093-4102.

(65) Ciavotta, M., Minella, G., & Ruiz, R. (2013). Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. *Eur. J. Oper. Res.,* 227, 301-313.

(66) Fumero, Y., Corsano, G., & Montagna, J. M. (2014). Simultaneous Batching and Scheduling of Batch Plants That Operate in a Campaign-Mode, Considering Non identical Parallel Units and Sequence-Dependent Changeovers. *Ind. Eng. Chem. Res., 53, 17059-17074.*

(67) Naderi, B., & Ruiz, R. (2014). A scatter search algorithm for the distributed permutation flowshop scheduling problem. *Eur. J. Oper. Res., 239, 323-334.*

(68) Cóccola, M. E., & Méndez, C. A. (2015). An iterative MILP-based approach for the maritime logistics and transportation of multi-parcel chemical tankers. *Comp. Ind. Eng., 89, 88-107.*

(69) Papageorgiou, L. G., & Pantelides, C. C. (1996). Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation. *Ind. Eng. Chem. Res.,* 35, 488-509.

(70) Ierapetritou, M. G., & Floudas, C. A. (1998). Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes. *Ind. Eng. Chem. Res.,* 37, 4360-4374.

(71) Mockus, L., & Reklaitis, G. V. (1999). Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Ind. Eng. Chem. Res.,* 38, 197-203.

(72) Pistikopoulos, E. N., Vassiliadis, C. G., Arvela, J., & Papageorgiou, L. G. (2001). Interactions of maintenance and production planning for multipurpose process plants a system effectiveness approach. *Ind. Eng. Chem. Res.,* 40, 3195-3207.

(73) Giannelos, N. F., & Georgiadis, M. C. (2002). A novel event-driven formulation for short-term scheduling of multipurpose continuous processes. *Ind. Eng. Chem. Res.,* 41, 2431-2439.

(74) Castro, P. M., Barbosa-Póvoa, A. P., Matos, H. A., & Novais, A. Q. (2004). Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. Chem. Res.,* 43, 105-118.

(75) Castro, P. M., Aguirre, A. M., Zeballos, L. J., & Méndez, C. A. (2011). Hybrid mathematical programming discrete-event simulation approach for large-scale scheduling problems. *Ind. Eng. Chem. Res., 50, 10665-10680.*

(76) Liu, S., Pinto, J. M., & Papageorgiou, L. G. (2010). MILP-based approaches for medium-term planning of single-stage continuous multiproduct plants with parallel units. *Comput. Manag. Sci., 7, 407-435.*

Appendices

A. TSP/immediate-precedence formulation [M1]

The model presented here was proposed by Liu et al.[25] for the planning and scheduling of multiproduct multistage plants. The main equations of this model are stated below in Eqs. (12-32). Eq. (12) was proposed to estimate the *TP* total profit considering the benefits for the products sold and the production costs associated with changeovers, backlogs and inventories. Then, Eqs.(13-20) are developed to assess the assignment decisions of products to weeks. These equations are based in the main ideas of TSP formulation. Thus, Eqs.(13-16) defines the first and the last product processed in week *w* by F_{iw} and L_{iw} respectively. While, Eqs.(17-20) determine the sequencing decisions between different products in the same or in consecutive weeks by Z_{ijw} and $ZF_{ijw.}$ Note that, these equations, ensure that at least one product must be processed in each week E_{iw} by following a particular production sequence. In order to eliminate possible sub-tours, additional timing and sequencing constraints are formulated in Eqs.(21-28). Timing constraints stated in Eqs.(21-22) defines the bounds of the processing time T_{imw} by *LB* and *UB*. Using this information an estimation of the total processing time for each unit *m* and week *w* is derived in Eqs.(23). Then, sequencing constraints in the same unit using the immediate-precedence concepts are presented in Eqs.(24-26) while sequencing decisions for continuous plants between consecutive units are enforced by Eqs.(27-28). Finally, production amounts P_{isw}, backlogs B_{ciw}, inventory level V_{iw} and sales SL_{ciw} are determine by Eqs.(29-32).

$$TP = \sum_{ciw} PS_{ciw} \cdot SL_{ciw} - \sum_{i,j \neq i,m,TH \geq w} CC_{ijm} \cdot Z_{ijw} \cdot t_{mij} - \sum_{i,j,m,TH \geq w > 1} CC_{ijm} \cdot ZF_{ijw} \cdot t_{mij} - \sum_{c,i,TH \geq w} CB_{ci} \cdot B_{ciw} - \sum_{i,TH \geq w} CI_i \cdot V_{iw} \tag{12}$$

$$\sum_i F_{iw} = 1 \qquad \forall w \leq TH \tag{13}$$

$$\sum_i L_{iw} = 1 \qquad \forall w \leq TH \tag{14}$$

$$F_{iw} \leq E_{iw} \qquad \forall i, w \leq TH \tag{15}$$

$$L_{iw} \leq E_{iw} \qquad \forall i, w \leq TH \tag{16}$$

$$\sum_{i \neq j} Z_{ijw} = E_{jw} - F_{jw} \qquad \forall j, w \leq TH \tag{17}$$

$$\sum_{j \neq i} Z_{ijw} = E_{iw} - L_{iw} \qquad \forall i, w \leq TH \tag{18}$$

$$\sum_i ZF_{ijw} = F_{jw} \qquad \forall j, 1 < w \leq TH \tag{19}$$

$$\sum_j ZF_{ijw} = L_{i,w-1} \qquad \forall i, 1 < w \leq TH \tag{20}$$

$$T_{imw} \leq UB \cdot E_{iw} \qquad \forall i, s, m \in MS, w \leq TH \tag{21}$$

$$T_{imw} \geq LB \cdot E_{iw} \qquad \forall i, s, m \in MS, w \leq TH \tag{22}$$

$$\sum_i T_{imw} + \sum_{i \neq j} t_{mij} \cdot Z_{ijw} + \sum_{i,j} t_{mij} \cdot ZF_{ijw|w>1} \leq UB \qquad \forall s, m \in MS, w \leq TH \tag{23}$$

$$TS_{jsw} \geq TS_{isw} + T_{imw} + t_{mij} \cdot Z_{ijw} - UB \cdot (1 - Z_{ijw}) \qquad \forall i, j \neq i, s, m \in MS, w \leq TH \tag{24}$$

$$TS_{jsw} \geq t_{mij} \cdot ZF_{ijw} - UB \cdot (1 - ZF_{ijw}) \qquad \forall i, j, s, m \in MS, 1 < w \leq TH \tag{25}$$

$$TS_{isw} + T_{imw} \leq UB \cdot E_{iw} \qquad \forall i, s, m \in MS, w \leq TH \tag{26}$$

$$TS_{jsw} \leq TS_{j,s+1,w} \qquad \forall i, s < S, 1 < w \leq TH \tag{27}$$

$$TS_{isw} + T_{imw} \leq TS_{i,s+1,w} + T_{i,m+1,w} \qquad \forall i, S < S, m \in MS, w \leq TH \tag{28}$$

$$P_{isw} = r_{mi} \cdot T_{imw} \qquad \forall i, s, m \in MS, w \tag{29}$$

$$n_{si} \cdot P_{isw} = P_{i,s+1,w} \qquad \forall i, s < S, w \tag{30}$$

$$B_{ciw} = B_{ci,w-1} + D_{ciw} - SL_{ciw} \qquad \forall c, i, w \tag{31}$$

$$V_{iw} = V_{i,w-1} + P_{isw} - \sum_{c} SL_{ciw} \qquad \forall i, s = S, w \tag{32}$$

B. Unit-Specific Precedence-based formulation [M2]

The main equations of an adapted version of a unit-specific general-precedence model of Kopanos et al.[57] are presented in Eqs.(33-39).The original model was formulated for correctly tackle the problem of sequence-dependent changeovers in a single-stage batch plant with multiple units in parallel. In here we extend this model to couple with multiple stages. For this, we add a general-precedence variable X_{ijw} for all different products *(i≠j)* in the same unit *m* at stage *s* on week *w* by Eq.(33) in order to determine the production sequence. Eq.(34) is provided enforcing a minimum processing time at the beginning of the week. Then, Eqs.(35-37) are derived to ensure that X_{ijw} takes some value only if both products *i* and *j* are processing in the same week *w* by E_{iw} & E_{jw}.
In order to define the immediate (local) precedence of two different tasks *i,j* in the same unit *j* by Z_{ijw} additional sequencing-allocation constraints are introduced in Eqs.(38-39). These equations are able to deal with the sequence-dependent changeovers correctly but at the same time increase the size of the model considerably. These Big-M constraints run for all different tasks in the system *(j≠i)* which may affect the performance of the model. Finally, the *BM* value is defined as the maximum number of possible products to produce in a single week.

$$TS_{jsw} \geq TS_{isw} + T_{imw} + t_{mij} \cdot Z_{ijw} - UB \cdot (1 - X_{ijw}) \qquad \forall i, j \neq i, s, m \in MS, w \leq TH \tag{33}$$

$$TS_{jsw} \geq \sum_i t_{mij} \cdot ZF_{ijw} \qquad \forall j, s, m \in MS, 1 < w \leq TH \tag{34}$$

$$\sum_i E_{iw} \geq 1 \qquad \forall w \leq TH \tag{35}$$

$$E_{iw} + E_{jw} \leq 1 + X_{ijw} + X_{jiw} \qquad \forall i, j \neq i, w \leq TH \tag{36}$$

$$E_{i,w} + E_{j,w} \geq 2 \cdot (X_{i,j,w} + X_{j,i,w}) \qquad \forall i, j \neq i, w \leq TH \tag{37}$$

$$\sum_{k \neq i \neq j} (X_{kjw} - X_{kiw}) + BM \cdot (1 - X_{ijw}) + Z_{ijw} \geq 1 \qquad \forall i, j \neq i, w \leq TH \tag{38}$$

$$\sum_{k \neq i} (X_{kjw}) + \sum_{k \neq j} (X_{ik,w-1}) + BM \cdot (2 - E_{jw} - E_{i,w-1}) + ZF_{ijw} \geq 1 \qquad \forall i, j \neq i, 1 < w \leq TH \tag{39}$$

C. General Precedence-based formulation [M3]

In order to reduce the number of binary variables used in the previous models a reformulated version of the one proposed in Aguirre et al.[58] for flow-shop scheduling problems is presented in Eqs.(40-48) by using the main idea of general-precedence representation. Here, a sequencing variable X_{ijw} is proposed for sequence-dependent changeovers while additional constraints for sequencing decisions are introduced in Eqs.(40-42). It is worth emphasizing that this model only creates sequencing variables X_{ijw} for *(j<i)* reducing a half the number of binary variables of the previous formulations described above. Another advantage is the use of a continuous variable O_{iw} for the sequence-dependent issue. This variables defines the absolute position of the product *i* in the production sequence by Eqs.(43-45) while Z_{ijw} and ZF_{ijw} are added to consider the immediate-precedence between consecutive products *(j≠i)* in the same or consecutive weeks *w* by Eqs.(46-48). The main difference with the previous formulations relies in the way to calculate the immediate-precedence variables using only general-precedence information. Despite of the improvement in the model formulation, this formulation still require many constraints and variables which turns very complex the resolution of medium-term problems in a reasonable CPU time.

$$TS_{jsw} \geq TS_{isw} + T_{imw} + t_{mij} \cdot (Z_{ijw}) - UB \cdot (3 - X_{ijw} - E_{iw} - E_{jw}) \quad \forall i, j < i, s, m \in MS, w \leq TH \tag{40}$$

$$TS_{isw} \geq TS_{jsw} + T_{jmw} + t_{mji} \cdot (Z_{jiw}) - UB \cdot (X_{ijw}) - UB \cdot (2 - E_{iw} - E_{jw}) \quad \forall i, j < i, s, m \in MS, w \leq TH \tag{41}$$

$$TS_{jsw} \geq \sum_i t_{mij} \cdot ZF_{ijw} \quad \forall j, s, m \in MS, 1 < w \leq TH \tag{42}$$

$$O_{jw} \geq O_{iw} + 1 - BM \cdot (1 - X_{ijw}) - BM \cdot (2 - E_{iw} - E_{jw}) \quad \forall i, j < i, w \leq TH \tag{43}$$

$$O_{iw} \geq O_{jw} + 1 - BM \cdot (X_{ijw}) - BM \cdot (2 - E_{iw} - E_{jw}) \quad \forall i, j < i, w \leq TH \tag{44}$$

$$\sum_j E_{jw} \geq O_{iw} \geq E_{iw} \quad \forall i, w \leq TH \tag{45}$$

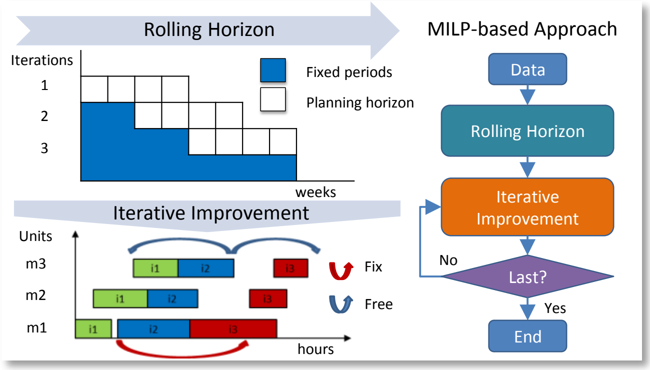
$$Z_{ijw} + O_{jw} - O_{iw} - 1 + BM \cdot (1 - X_{ijw}) \geq 1 \quad \forall i, j < i, w \leq TH \tag{46}$$

$$Z_{jiw} + O_{iw} - O_{jw} - 1 + BM \cdot (X_{ijw}) \geq 1 \quad \forall i, j < i, w \leq TH \tag{47}$$

$$ZF_{ijw} + O_{jw} - 1 + BM \cdot (1 - E_{jw}) + \left(\sum_k (E_{k,w-1}) - O_{i,w-1} \right) + BM \cdot (1 - E_{i,w-1}) \geq 1$$

$$\forall i, j \neq i, 1 < w \leq TH \tag{48}$$

Table of Contents (TOC)


Rolling Horizon
MILP-based Approach
Iterations
1
2
3
Fixed periods
Planning horizon
weeks
Data
Rolling Horizon
Iterative Improvement
Iterative Improvement
Units
m3
m2
m1
i1
i2
i3
Fix
Free
hours
No
Last?
Yes
End