

Medium-term optimization-based approach for the integration of production planning, scheduling and maintenance

Adrian M. Aguirre,^{a*} Lazaros G. Papageorgiou^{a**}

^a Centre for Process Systems Engineering, Department of Chemical Engineering, UCL (University College London), Torrington Place, London WC1E 7JE, United Kingdom.

* *a.aguirre@ucl.ac.uk*

** *l.papageorgiou@ucl.ac.uk*

Abstract

A medium-term optimization-based approach is proposed for the integration of production planning, scheduling and maintenance. The problem presented in this work considers a multiproduct single-stage batch process plant with parallel units and limited resources. An MILP continuous-time formulation is developed based on the main ideas of travelling salesman problem and precedence-based constraints to deal with, sequence-dependent unit performance decay, flexible recovery operations, resource availability and product lifetime. Small scheduling examples have been solved and compared with adapted formulations from the literature, based on discrete-time and global-time events, demonstrating the effectiveness of the proposed solution approach. Additional planning and scheduling problems have been proposed by considering several time periods. Multi-period examples have been efficiently solved by the model showing the applicability of the solution approach for medium-size problems.

Keywords: MILP-based approach, Planning, Scheduling and Maintenance, Traveling Salesman Problem, Precedence-based representation, Performance decay, Flexible recovery operations.

1. Introduction

Joint production planning, scheduling and maintenance have been attracting the attention of the research community, but not many approaches have been developed considering the integration of production planning, scheduling and maintenance with limited resources, sequence-dependent unit performance decay, flexible recovery operations and product lifetime.

In the past, Dedopoulos & Shah (1995) developed a mathematical model for the short-term scheduling production and maintenance activities within a multipurpose plant, considering production and maintenance resources. This formulation was extended in Dedopoulos & Shah (1996) to consider combined production/maintenance aggregated planning problem. Based on the previous works, Vassiliadis et al. (2000) proposed an optimization framework combining ideas of STN (state-task network) model for the production planning and Markov Chain for the maintenance model. Similarly, Pistikopoulos et al. (2001) proposed an effective approach for the interaction of production planning and maintenance in multipurpose process plants. They developed an MILP formulation based on STN representation for the production planning and maintenance by dividing the planning horizon into equal length periods and assuming a constant failure rate of the unit during the operation period. Thus, each unit operates for a number of time periods until a preventive maintenance operation is needed and then it is restored to its initial condition. Later on, based on the same assumptions, Suryadi & Papageorgiou (2004) proposed a novel integrated approach for the optimal maintenance planning incorporating preventive maintenance and novel design aspects on multipurpose batch plants.

For the cycle scheduling and maintenance of process plants, Jain and Grossmann (1998) developed a MINLP model for cycle scheduling and maintenance with production decay in continuous parallel processes. Alle et al. (2004a) presented an MINLP formulation for the production and cleaning operations in multiproduct serial plants. Then, Alle et al. (2004b) formulated a TSP-based model for cycle scheduling and cleaning operations considering linear performance decay. Casas-Liza et al. (2005) extended the previous work to consider exponential performance decay.

The integration of preventive maintenance planning and production scheduling problem for a batch plant in a single machine has been studied by Cassady and Kutanoglu (2005). They proposed an MILP model based on WSTP (weight shortest processing time) rule for the scheduling problem, assuming that the preventive maintenance (PM) restores the machine to “as good as new” condition after failure. Given the information of the MTBF (mean time between failures), they calculate the probability of a failure during the processing time, so as to decide if is necessary to perform a PM with the aim of minimising the weighted expected completion time. Bock et al. (2012) addressed a single machine scheduling and maintenance problem by considering the idea of maintenance level (ML). This ML decreased while the job is processed. Thus, partial and fully maintenance

activities (MA) were considered by the model according to the sequence of jobs processed in each unit, allowing recovery operations to happen at any time and to any level between the actual and the maximum level. Ruiz et al. (2007) extend the previous ideas for a permutation flowshop sequencing problem (PFSP) comparing different heuristics and metaheuristics specially developed for this kind of NP-hard problem. Then, Varnier and Zerhouni (2012) proposed an MILP model for the integration of production scheduling and predictive maintenance in flowshop systems by considering both makespan and remaining useful life (RUL) in the objective function.

Back to the process industry, Hazaras et al. (2012) has introduced a global event continuous-time MILP-based formulation for the flexible maintenance and scheduling problem. Nie et al. (2014) proposed a discrete-time RTN model for the optimal reactive scheduling of mixed batch/continuous process with resources. While, Velez et al. (2015) provide discrete-time STN formulation to deal with large-scale production systems. In these works, the reduction of unit life depends on the amount of production processed by the unit. Castro et al. (2014) developed a slot-based continuous-time formulation for the maintenance scheduling of a gas engine power plant by generalised disjunctive programming (GDP). They reformulate the GDP into MILP by using Big-M and Hull reformulations for comparisons. In here, the maintenance occurs after a given amount of time spent online.

More recently, Liu et al. (2014) developed a continuous-time formulation for the production and maintenance planning in the biopharmaceutical industry under performance decay. In addition, Vieira et al. (2017) proposed a continuous-time RTN single-grid formulation to deal with optimal production and maintenance planning by minimising the intermediate maintenance operations while maximizing the average service level and the utilization rate of processing units.

Finally, Biondi et al. (2017) proposed a multi-time scale MILP formulation based on STN representation for the integration of maintenance and production scheduling of the steel making process. For that, an aggregated model for the planning horizon is provided while a detailed model is proposed for the scheduling horizon. Both models are based on the previous concepts of Residual Useful Life (RUL). Hence, the unit operates in a degraded mode, reducing the RUL, and then a recovery operation is performed any time to restore the machine to the useful level. In here, the operation mode the unit uses at each time period is a decision variable that the model has to optimize in order to reduce the total cost formed by the storage cost, maintenance cost and the overall wear cost.

In this work a TSP/precedence-based continuous-time formulation is proposed to solve a class of planning, scheduling and maintenance problem. This new formulation relies on the main ideas of TSP (Pinto et al. 2007, Liu et al. 2008, 2010) and precedence-based representation (Mendez et al., 2006), to address planning, scheduling and maintenance decisions of a multi-period multiproduct single-stage process plant with parallel units and limited resource. The integration of sequence-dependent unit performance decay, flexible recovery operations and resource limitations, in a continuous way, represent the main contribution of this novel formulation. In here, the unit performance decays during the processing time by considering a linear constant degradation. This constant degradation differs for each product and unit, and the performance of the unit should be determined by the sequence of products processed. Linear performance decay is considered in this paper by a constant rate factor. Similar papers that take into account production scheduling and preventive maintenance with fixed factor for unit reliability can be found in Sanmartí et al. (1997) and in Alle et al. (2004b) for example. Some similar application examples can be found in the optimal operation and maintenance of networks of compressors of chemical plants in Xenos et al. (2016) or in typical evaporation system in the sugar industry with several multiple effect parallel lines and time decaying performance in Heluane et al. (2012) and in the cyclic scheduling of cracking furnace shutdowns in ethylene plants in Schulz et al. (2006).

In this work, unit reliability is represented in terms of unit performance, assuming units became very inefficient when the performance is too low. In order to illustrate this, we model the processing time as a function of the unit performance. This idea has been adopted by other papers in the past, assuming that unit operates in different operation modes. For example, at Biondi et al. (2017), similar ideas were applied to the steel making process by considering three operation modes (slow mode, nominal mode, fast mode). These operational modes are decision variables in the system and the actual performance of the unit may limit the choices of choosing one of them. In this work, the operation modes are considered as continuous variable that change over the time, according to the current state of the unit performance. Assuming a fixed batch size for each product-unit, the processing time is determined by the model according to the current unit performance. Thus, the sequence of products at each unit needs to be explicitly considered by the model to correct tracking the unit performance level at any time and to calculate the real processing time.

Flexible recovery operations are taken into account in order to maintain the performance of the unit in a reasonable level. Specific limitations on the number of recovery operations per time period and maximum recovery operations along the planning horizon are also enforced by the model. Also, initial and final unit performance levels at the beginning and at the end of the planning horizon are forced to be the same in order to keep the unit operating for the next planning horizon.

Product inventory, backlog and lifetime constraints are also taken into account by the model for each time period. Finally, production and maintenance costs, inventory costs, backlog costs and product prices are considered so as to maximise the total profit of the whole planning horizon.

The rest of the paper is described as follows. First, we present a general MILP model for a multi-period multi-product single-stage problem with parallel units and limited resources while considering unit performance decay and flexible recovery operations. Then, we introduce some small examples to demonstrate the effectiveness of our solution approach in comparison with adapted formulations from the literature. Finally, we provide motivating examples to demonstrate the applicability of the proposed solution approach for medium-size problems. Sensitivity analysis of the model for different possible scenarios and concluding remarks is presented at the end.

2. Problem statement

The problem presented in this work considers the integration of planning, scheduling and maintenance of multiple products in a single-stage production process with parallel units and limited resources (see Figure 1).

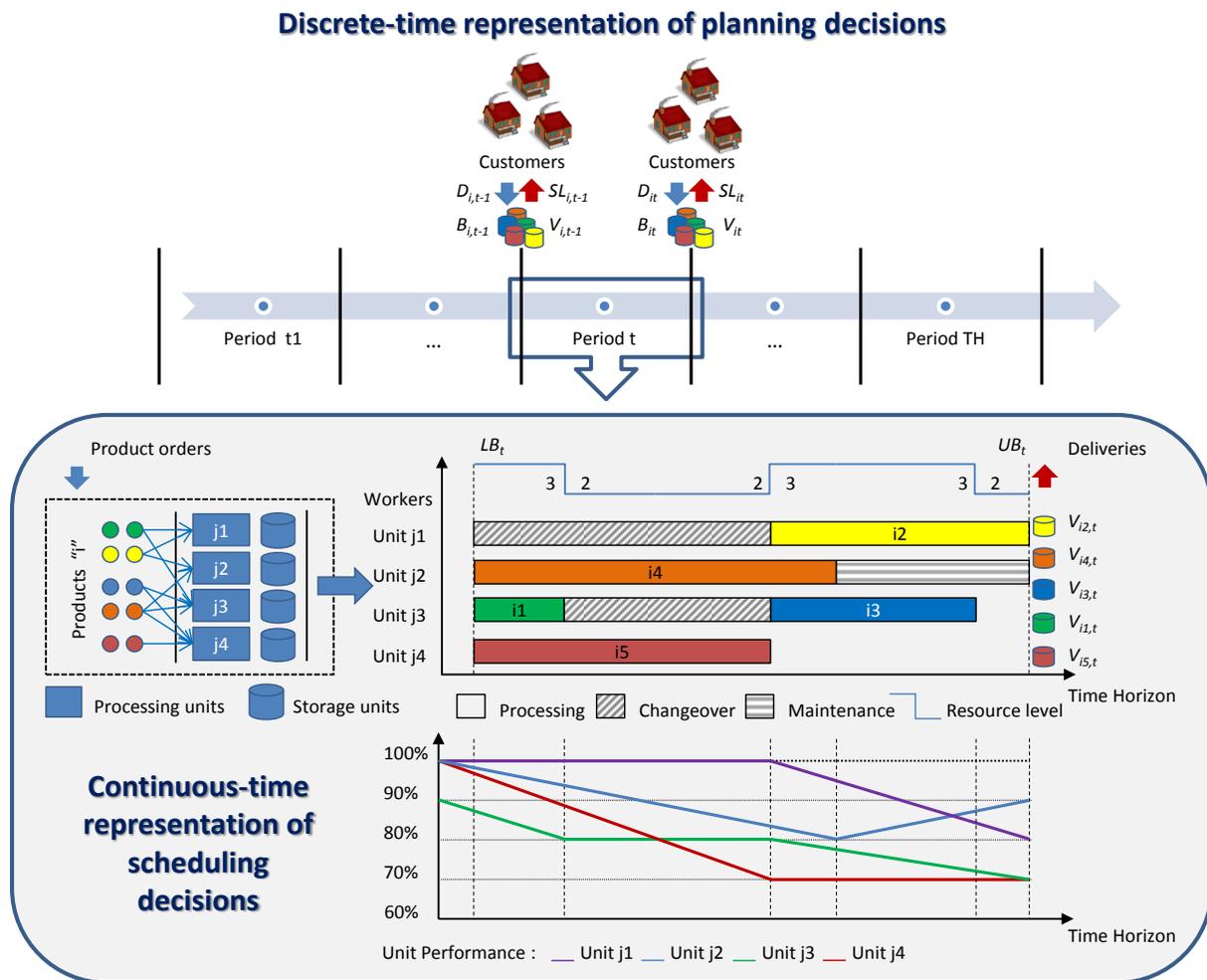


Figure 1. Integration of planning, scheduling & maintenance in a single-stage process with parallel units and shared resources

The main features of this problem are presented below:

Planning

- Product demands [kg] for each time period are known and given in advance. Product batches sold are delivered at the end of each time period. Product prices [\$] are known and fixed.
- Once produced, product batches are stored in the system. The inventory level is updated at the end of each time period. Unit inventory cost is considered as 30% of the product prices [\$/kg]. The inventory cost does not consider the storage time.

- In case of unsatisfied demand, late deliveries can be programmed as backlogs. Backlogs have a penalty cost which is 50% of the product prices [\$/kg].
- Products lifetime is defined as θ . Therefore, produced batches cannot be stored for more than θ consecutive time periods in the system.
- The planning horizon is divided into equal length time-periods. Each time period represents a month of 30 working days. Weekends are not considered, but could be easily introduced in the model by assuming time periods of 5 working days.

Scheduling

- Multiple products can be processed per time period. This should be determined by the model according to demand, sales, inventory levels and backlog limitations.
- Products should be processed in only one of the available units at each time period. Unit availability is given in advance.
- Fixed cost related to the production of a particular product at each time period and variable costs associated with the production time are known in advance.
- Product batch size is determined by the model according with the fixed batch size of each unit.
- Sequence-dependent changeover operations are taking place when two different products are immediately processed one after the other in the same unit.
- The beginning of the next product in a processing unit should be done after the current one has been completed. Thus, non-pre-emptive operation mode is applied at each processing unit.

Resources

- Resource availability related to processing and recovery operations are known in advance.
- A single type of resource, manpower, is required for both processing and recovery operations. All resource units are considered identical.
- There is no difference between processing and recovery operations for resource assignments. Thus, any resource can be assigned to processing or recovery operations.
- There is no cost associated with the limited resources. We assume that the resources are contracted in advance and there are only limitations in the amount of resources used during a time period.

Performance

- Linear unit performance degradation is considered known. There are different performance degradations for each product-unit. These performance degradations remain constant during the processing time.
- Processing time is determined by the initial performance of the unit before processing, the given batch size and the constant performance degradation.
- Unit performance at the beginning and at the end of the planning horizon is determined by the model, and should be equal.

Recovery

- Recovery operations are always taking place at the end of the processing operations. No slack time between processing and recovery operations are allowed.
- Recovery operations are determined by the model. There is a maximum number of recovery operations that could be performed at each time period and also there is a maximum number of recovery operations for each unit in consecutive time periods ϕ .
- Fixed cost related to the recovery operations and variable costs associated with the recovery time are known in advance.
- There are minimum and maximum recovery times when a recovery operation is performed. The maximum recovery time is associated with the maximum time required to fix the unit to its "as good as new" condition when the unit performance is zero.

According to all the features described above and given the information of,

- product batch $I = \{i1 \dots I^{max}\}$,
- parallel unit $J = \{j1 \dots J^{max}\}$,
- time periods $TH = \{t1 \dots T^{max}\}$,
- breakpoints $K = \{k1 \dots K^{max}\}$,
- batch-unit $IJ = \{(i,j) | p_{ij} > 0\}$,
- fixed batch size p_{ij} ,
- processing times tp_{ijk} ,

- changeover times $\tau_{ii'}$,
- maximum recovery time tr_j^{\max} ,
- production rates r_{ij} ,
- product demand D_{it} ,
- amount of resource needed ρ_i ,
- performance decay d_{ij} ,
- unit performance value y_k ,
- product lifetime θ ,
- elapsed time between maintenance operations ϕ ,
- lower LB_t and upper bounds UB_t of time period t ,
- maximum number of available resources R_t^{\max} at time period t ,
- maximum number of recovery operations M_t^{\max} at time period t ,
- big-M value BM ,

this problem aims to find the best production plan and schedule for all time periods by computing

- the starting TS_{it} and ending times TF_{it} of each product batch i at time period t ,
- assignment decisions $E_{ijt}, L_{ijt}, F_{ijt}$ of product batch i on unit j at time period t ,
- sequencing decisions $Z_{ii'jt}, X_{ii't}, Y_{ii't}$ of product batches i, i' on unit j or different units at time period t ,
- flexible processing times T_{ijt} and recovery times RT_{ijt} batch i on unit j at time period t ,
- total amount of resources R_{it} used by all batches at time TS_{it} ,
- recovery operations U_{ijt} performed at the end of the product batch i on unit j at time period t ,
- unit performance at the beginning y_{jt}^0 of each time period t ,
- the initial y_{ijt}^1 and final y_{ijt}^2 performance of product batch i at unit j ,
- production amount PR_{it} , inventory V_{it} , backlogs B_{it} and sales SL_{it} are also taken into account,

so as to maximize the total profit TP by considering, product prices p_i , fixed fpc_{ij} and variable vpc_{ij} production costs, fixed fmc_{ij} and variable vmc_{ij} recovery costs, backlog bc_i cost and inventory vc_i costs.

3. Multi-period continuous-time formulation

A multi-period continuous-time model for a multiproduct single-stage continuous plant with parallel units and limited resources is presented in this section by Eqs. (1-44). Assignment and sequencing decisions are introduced by Eqs. (1-6). Here, binary variables E_{ijt}, F_{ijt} and L_{ijt} are proposed to decide which products are produced at each unit at different time periods. Immediate-precedence variables $Z_{ii'jt}$ are derived from the sequencing decisions between different products ($i \neq i'$) at the same unit j for time period t . Timing constraints are stated by Eqs. (7-9), for the starting TS_{it} and finishing time TF_{it} of product batch i at period t . These variables are bounded between lower LB_t and upper bounds UB_t for each period t by Eqs. (10-13). Processing times defined by T_{ijt} are calculated in Eqs. (14-19). This processing time is a function of the initial y_{ijt}^1 and final unit performance y_{ijt}^2 . Assuming that the amount of product batch i produced in unit j at any time period p_{ij} is known in advance, then the processing time could be expressed by a simple quadratic equation which depends only on the initial performance of the unit y_{ijt}^1 . Thus, the non-linear expression for the processing time presented in Eq. (18) can be linearized by Eqs. (20-23) using a piecewise linearization by considering discrete unit performance y_k at different breakpoints k .

The amount of resources used by processing and recovery operations in the system is calculated based on the main ideas of Marchetti and Cerdá (2009). Thus, Eqs. (24-27) are presented to consider discrete renewable resources, as manpower, for each product batch i at time period t by R_{it} . For this, two additional binary variables, based on general-precedence concepts, are introduced $X_{ii't(i>i')}$ and $Y_{ii't(i \neq i')}$. These variables determine when two different products are overlapping in time by Eqs. (24-26). Hence, the amount of resources is calculated using the expression presented in Eq. (27).

The real performance of the unit along the time is also considered by the model in Eqs. (14, 28-34). Here the performance of the unit decays constantly according to d_{ij} . Thus, the final performance y_{ijt}^2 is obtained by the initial performance y_{ijt}^1 minus the processing time T_{ijt} multiplied by d_{ij} , as is stated in Eq. (14).

Sequence-dependent unit performance decay is stated by Eqs. (28-29). The initial unit performance $y_{i'jt}^1$ of product batch i' in unit j depends on the final performance y_{ijt}^2 of immediately previous product batch i determined by $Z_{ii'jt}$, at time period t . The recovery operation at the end of product batch i , defined by RT_{ijt}/rt^{\max} , also affects the initial performance of the product batch i' .

Then, Eqs. (30-35) enforce the same initial and final performance of the unit at the beginning and at the end of the planning horizon TH determined by $y_{j,0}^0 = y_{j,T}^0 = y_{j,T}^{\max}$. Flexible recovery times RT_{ijt} of product batch i in unit j at

time period t are enforced by U_{it} in Eqs. (36-39). This binary variable determines if a recovery operation is needed or not at the end of production batch of product batch i .

More complex and challenging features are introduced here by considering the resolution of planning, scheduling for single or multiple time periods of a multi-product single-stage continuous process plant. For this, production, inventory and backlog constraints are introduced by Eqs. (40-42). Thus, product lifetime is considered by Eq. (43) forcing to sale produced batch instead of keeping in inventory for a long time. This is commonly considered when products are deteriorated along the time. Examples of perishable products, like food (Kopanos et al., 2009) or bio-pharmaceutical products (Devapriya et al. 2017), can be found in the literature. Finally, this model is solved by maximising the total profit TP in Eq. (44) calculated by the revenues of sales while considering production, maintenance, inventory and backlog costs.

3.1. Unit assignment and batch sequencing

The unit assignment and sequencing decisions of products are represented by Eqs. (1-6). The following variables are needed to represent assignment and sequencing decisions. Note that, only E_{ijt} and $Z_{ii'jt}$ should be defined as binary variables while F_{ijt} and L_{ijt} can be defined as continuous or binary variables.

$$\begin{aligned}
 E_{ijt} & \begin{cases} 1 & \text{if product batch } i \text{ is performed in unit } j \text{ at time period } t \\ 0 & \text{otherwise} \end{cases} \\
 F_{ijt} & \begin{cases} 1 & \text{if product batch } i \text{ is the first product batch performed in unit } j \text{ at time period } t \\ 0 & \text{otherwise} \end{cases} \\
 L_{ijt} & \begin{cases} 1 & \text{if product batch } i \text{ is the last product batch performed in unit } j \text{ at time period } t \\ 0 & \text{otherwise} \end{cases} \\
 Z_{ii'jt} & \begin{cases} 1 & \text{if product batch } i \text{ is performed exactly before product batch } i' \text{ in unit } j \text{ at time period } t \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

Eqs. (1-2) ensure that at most one product should be produced at first or last in a single unit j at time period t . Eq. (3) is used to enforce a single unit assignment per product batch i at time period t . While Eq. (4) is provided to relate first and last variables with the assignment decisions allowing the possibility to have none, one or multiple products in a single unit j at time period t . Sequencing constraints based on immediate precedence concepts are stated in Eqs. (5-6). These equations enforce a single predecessor or successor for each product batch i in unit j and time period t .

$$\sum_{i \in I} F_{ijt} \leq 1 \quad \forall j \in J, t \in TH \quad (1)$$

$$\sum_{i \in I} L_{ijt} \leq 1 \quad \forall j \in J, t \in TH \quad (2)$$

$$\sum_{j \in J} E_{ijt} \leq 1 \quad \forall i, t \in TH \quad (3)$$

$$L_{ijt} + F_{ijt} \leq 2 \cdot E_{ijt} \quad \forall i, j \in IJ, t \in TH \quad (4)$$

$$\sum_{i \neq i' \in I} Z_{ii'jt} + L_{ijt} = E_{ijt} \quad \forall i, j \in IJ, t \in TH \quad (5)$$

$$\sum_{i \neq i' \in I} Z_{i'ijt} + F_{ijt} = E_{ijt} \quad \forall i, j \in IJ, t \in TH \quad (6)$$

3.2. Timing constraints

Timing decisions about the starting and finishing time of product batch i at time period t are represented by continuous variables TS_{it} and TF_{it} . Eq. (7) is provided for sequencing and timing decisions of different products i and i' in the same unit j at time period t while Eq. (8) is introduced for sequencing and timing decisions between

the same or different products in unit j at consecutive time periods $t-1$ and t . The fishing time of product batch i at time period t is calculated by Eq. (9) considering the processing time T_{ijt} and the recovery time by RT_{ijt} . Bounding constraints of product batch i produced at time period t are stated by Eqs. (10-13). Note that, Eqs.(12-13) are redundant due to Eqs.(9-11).

$$TS_{it} \geq TF_{it} + \sum_{j \in IJ} (\tau_{ij} \cdot Z_{ijt}) - UB_t \cdot \left(1 - \sum_{j \in IJ} Z_{ijt} \right) \forall i, i': (i \neq i'), t \in TH \quad (7)$$

$$TS_{it} \geq TF_{i,t-1} + \tau_{ii'} - UB_t \cdot (2 - L_{ij,t-1} - F_{i',jt}) \forall i, i', j \in IJ, t \in TH \quad (8)$$

$$TF_{it} = TS_{it} + \sum_{j \in IJ} T_{ijt} + \sum_{j \in IJ} RT_{ijt} \quad \forall i, t \in TH \quad (9)$$

$$TS_{it} \geq LB_t \cdot \sum_{j \in IJ} E_{ijt} \quad \forall i, t \in TH \quad (10)$$

$$TS_{it} \leq UB_t \cdot \sum_{j \in IJ} E_{ijt} \quad \forall i, t \in TH \quad (11)$$

$$TF_{it} \geq LB_t \cdot \sum_{j \in IJ} E_{ijt} \quad \forall i, t \in TH \quad (12)$$

$$TF_{it} \leq UB_t \cdot \sum_{j \in IJ} E_{ijt} \quad \forall i, t \in TH \quad (13)$$

3.3. Processing time estimation

The performance of the unit j at the end of the production time of product batch i is represented by Eq. (14). The final performance y^2_{ijt} is a function of the initial performance y^1_{ijt} of the unit j at any time period t minus the decay d_{ij} multiplied by the processing time T_{ijt} .

$$y^2_{ijt(\tau)} = y^1_{ijt} - \tau \cdot d_{ij} \quad \forall i, j \in IJ, t \in TH \quad (14)$$

The amount of product batch i produced in unit j at any time period, p_{ij} , is given by the integral of the final performance of unit j , along the processing time T_{ijt} , multiplied by the fixed production rate r_{ij} of product batch i at unit j , as is determined by Eq. (15).

$$p_{ij} = r_{ij} \cdot \int_{\tau=0}^{\tau=T_{ijt}} y^2_{ijt(\tau)} d\tau \quad \forall i, j \in IJ, t \in TH \quad (15)$$

Replacing Eq. (14) in Eq. (15) and solving the integral, the production amount is finally represented by Eq.(16).

$$p_{ij} = r_{ij} \cdot \left(-\frac{d_{ij}}{2} \cdot T_{ijt}^2 + y^1_{ijt} \cdot T_{ijt} \right) \forall i, j \in IJ, t \in TH \quad (16)$$

Assuming that the production amount and the processing rate are known in advance and moving p_{ij} to the right-hand side of Eq. (16), we can obtain the following quadratic equation in Eq. (17).

$$-\frac{d_{ij}}{2} \cdot T_{ijt}^2 + y^1_{ijt} \cdot T_{ijt} - \frac{p_{ij}}{r_{ij}} = 0 \quad \forall i, j \in IJ, t \in TH \quad (17)$$

The solution roots of the quadratic equation of Eq.(17) are given by Eq.(18).

$$T_{ijt} = \frac{-y^1_{ijt} \pm \sqrt{(y^1_{ijt})^2 - 4 \cdot \left(\frac{d_{ij}}{2} \cdot \frac{p_{ij}}{r_{ij}}\right)}}{-2 \cdot \left(\frac{d_{ij}}{2}\right)} \quad \forall i, j \in IJ, t \in TH \quad (18)$$

Thus, Eq. (18) provides the roots of the quadratic equation for different y^j_{ijt} , p_{ij} , r_{ij} and d_{ij} parameters. The exact solution of T_{ijt} is the one that satisfies the conditions imposed by Eqs. (14-15, 19). These equations ensure positive values on unit performances, avoiding choosing the root that generates negative areas as it is shown in Figure 2. It is worth noticing that, the determinant of negative values provides non-real numbers. For these cases, we assume that the unit fails and a corrective maintenance is urgently needed to continue using this unit.

$$1 \geq y^1_{ijt} \geq y^2_{ijt} \geq 0 \quad \forall i, j \in IJ, t \in TH \quad (19)$$

Figure 2 shows a geometrical interpretation of solution roots expressed by Eq. (18). The results of these two roots are expressed by w_1 and w_2 which are basically different values of T_{ijt} . These also represent the width of the areas in Figure 2 determine by Eq. (15). The heights of these areas are represented by h_1 and h_2 . The resulting area (\triangle) of each root is determined by the summation of the triangle (Δ) and the square (\square) areas in Figure 2. As it can be seen, the first root provides the area behind the curve by calculating $\triangle_1 = \Delta_1 + \square_1$. The second root ($\triangle_2 = \Delta_2 + \square_2$) also provides the same result ($\triangle_1 = \triangle_2$) but some of the resulting areas are negative. This behaviour is repeated for any proposed value of the y^j_{ijt} , p_{ij} , r_{ij} and d_{ij} parameters in Eq. (18).

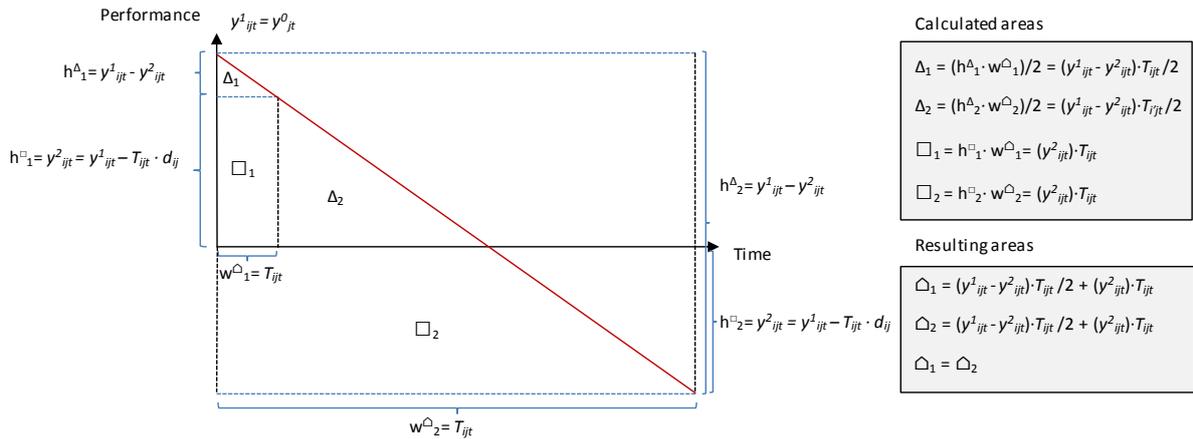


Figure 2. Geometrical interpretation of the solution roots of Eq.(18).

3.4. Piecewise linearization of processing times

Figure 3 shows an approximate solution of Eq. (18) for different values of y^j_{ijt} . In order to do this, a piecewise linearization is applied discretising the unit performance in equal length intervals represented by discrete breakpoints k . The length of the interval is defined in 0.1 and the total number of breakpoints is $|K| = 10$.

Each breakpoint k represents a different value of the unit performance y_k in between $[0,1]$ and its corresponding processing time tp_{ijk} is calculated offline by Eq. (18). Thus, we found a piecewise linear curve of the processing times based on the unit performance so as to be used as an input parameter in the model.

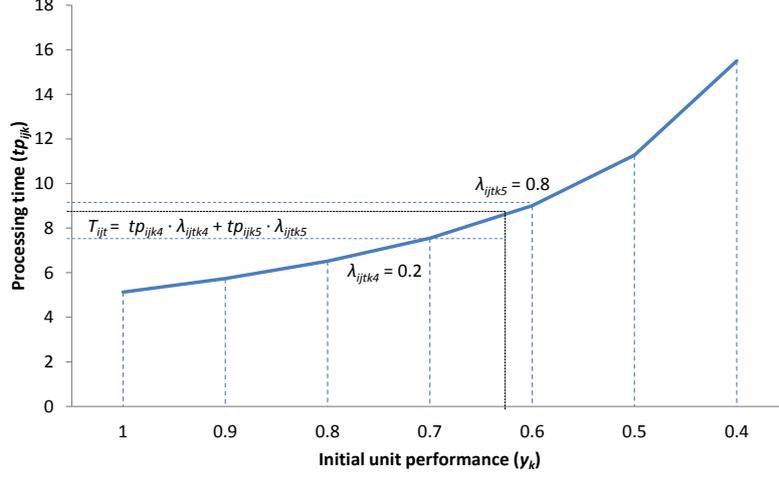


Figure 3. Processing time tp_{ijk} at different unit performance y_k .

The solution of the piecewise linearization is done offline and the real processing time is calculated by Eqs. (20-23). Here, SOS2 variable λ_{ijk} is introduced in the model select a specific value of processing times in the parametric curve. This variable λ_{ijk} represents a linear combination of breakpoints of consecutive intervals. Thus, Eqs. (20-21) are derived to force the processing time to be a specific value for T_{ijt} while Eq. (22) does the same with the unit performance y_{ijt}^1 . Finally, Eq. (23) forces the summation of all variables λ_{ijk} to be exact one for each triplet (i,j,t) . As it is defined in CPLEX, I.I. (2009), at most two of these variables λ_{ijk} may have non-zero values and the two non-zero values are adjacent.

$$T_{ijt} \leq \sum_k \lambda_{ijk} \cdot tp_{ijk} + BM \cdot (1 - E_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (20)$$

$$T_{ijt} \geq \sum_k \lambda_{ijk} \cdot tp_{ijk} - BM \cdot (1 - E_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (21)$$

$$y_{ijt}^1 = \sum_k y_k \cdot \lambda_{ijk} \quad \forall i, j \in IJ, t \in TH \quad (22)$$

$$\sum_k \lambda_{ijk} = 1 \quad \forall i, j \in IJ, t \in TH \quad (23)$$

3.5. Resource availability

Availability of renewable resources (e.g. manpower) is considered by Eqs. (24-27). For this, two additional binary variables $X_{ii't}$ and $Y_{ii't}$ are introduced. These variables denote when two different products are performed in different units at the same time (overlapped in time).

$$X_{ii't} \begin{cases} 1 & \text{if product batch } i' \text{ starts before or equal the starting time of product batch } i \text{ at time period } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ii't} \begin{cases} 1 & \text{if product batch } i' \text{ is completed whenever after the starting time of product batch } i \text{ at time period } t \\ 0 & \text{otherwise} \end{cases}$$

For example, for the case in which $i > i'$ and $X_{ii't} = Y_{ii't} = 1$ then $TS_{ii'} \geq TS_{i't}$ and $TS_{ii'} \leq TF_{i't}$, then both products batch i and i' are overlapping in time. Note that, if $X_{ii't} = 1$ then $Y_{ii't}$ could be $[0,1]$. But if $X_{ii't} = 0$ then $Y_{ii't}$ should be 1 which means that $TF_{i't} > TS_{i't} \geq TS_{ii'} + \delta$, where δ represents a small value to cope with the case of same starting time $TS_{i't} = TS_{ii'}$. The amount of resources $R_{ii't}$ used by each product batch i produced at time period t is calculated by Eq. (27). Thus, if product batch i is produced in unit j at time t then $E_{ijt} = 1$ then the amount of resources needed is going to be at least ρ_i . In case of overlapping with another product batch i' , e.g. if $i > i'$ and $X_{ii't} = Y_{ii't} = 1$, then the second term of Eq. (27) is going to be 1. On the other hand, if $i < i'$ and $X_{ii't} = 0$, $Y_{ii't} = 1$, then the third term of Eq. (27) is going to be 1. A simple example of the conditions stated above is shown in Figure 4.

$$TS_{it} \geq TS_{i't} - UB_i \cdot (1 - X_{i'it}) \quad \forall i, i': (i > i'), t \in TH \quad (24)$$

$$TS_{i't} \geq TS_{it} + \delta - UB_i \cdot (X_{i'it}) \quad \forall i, i': (i > i'), t \in TH \quad (25)$$

$$TS_{it} \geq TF_{i't} - UB_i \cdot (Y_{i'it}) \quad \forall i, i': (i \neq i'), t \in TH \quad (26)$$

$$R_t^{\max} \geq R_{it} \geq \left(\sum_{j \in I} \rho_j \cdot E_{ijt} + \sum_{i' > i} \rho_{i'} \cdot (Y_{i'it} + X_{i'it} - 1) + \sum_{i' < i} \rho_{i'} \cdot (Y_{i'it} - X_{i'it}) \right) \quad \forall i, t \in TH \quad (27)$$

Here it is worth to remark that if product batch i is not performed at time period t then both, TS_{it} and TF_{it} should be zero due to Eqs. (11,13), and then any relationship with another product batch $i' | i > i'$ should be denoted by $X_{i'it} = 0$ and $Y_{i'it} = 1$ and the second term of the RHS of Eq. (27) is going to be 0. The same for $i' | i < i'$, then $X_{i'it} = 1$ and $Y_{i'it} = 0$ and the third term of the RHS of Eq. (27) is going to be 0. Also, it is important to emphasise that if $TS_{it} = TF_{i't}$ then $X_{i'it} = 1$ and $Y_{i'it} = 0$ and so for this two product batches are not overlapping in time.

The total amount of resources used by all product batches at time TS_{it} is represented by R_{it} . This amount should be lower or equal than the maximum available resource defined by R_t^{\max} which is defined also by Eq. (27). In the case of equal starting times it is assumed arbitrarily that the batch i' starts before i and the amount of resources should be counted properly for the largest index batch i by Eq. (27). This idea is applied to equal starting times whatever the number of overlapping batches has happened. The other batches $i' < i$ basically underestimate the amount of resources required by the system at time $TS_{i't}$. Here we are using similar assumptions and equations originally provided in Marchetti and Cerda (2009). For more detailed information of this assumption please refer to Marchetti and Cerda (2009).

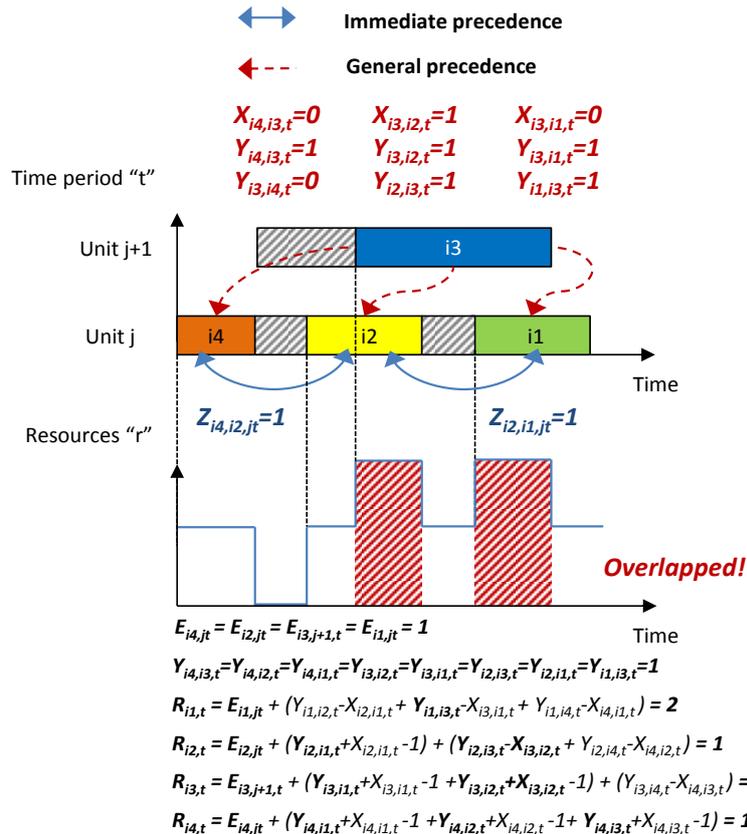


Figure 4. Resource-constrained example with product batches overlapping in time.

3.6. Unit performance

The performance of unit j is computed at the initial and at the final time event of the products processed in that unit at each time period. Thus, the performance at the beginning and at the end of product batch i performed in unit j is represented by y_{ijt}^1 and y_{ijt}^2 . Thus, the unit performance is changing according to the sequence of products produced at that unit at each time period. Then, sequence-dependent unit performance is provided in

Eqs. (28-29) by using the immediate-precedence variable defined for different products in $Z_{ii'jt}$. Thus, if $Z_{ii'jt}=1$ then product batch i' is processed immediately after product batch i in unit j at time period t and so for the initial performance of product batch i' , $y^1_{i'jt}$, should be equal to the final performance of product batch i , y^2_{ijt} , plus the percentage of unit recovery defined by RT_{ijt}/tr_j^{\max} where RT_{ijt} is the recovery time of product batch i in unit j at time period t and tr_j^{\max} is the maximum time required to recovery unit j to “as good as new” condition.

$$y^1_{i'jt} \geq y^2_{ijt} + \left(\frac{RT_{ijt}}{tr_j^{\max}} \right) - (1 - Z_{ii'jt}) \quad \forall i, i', j \in IJ : (i \neq i'), t \in TH \quad (28)$$

$$y^1_{i'jt} \leq y^2_{ijt} + \left(\frac{RT_{ijt}}{tr_j^{\max}} \right) + (1 - Z_{ii'jt}) \quad \forall i, i', j \in IJ : (i \neq i'), t \in TH \quad (29)$$

If product batch i is the first product to be produced in unit j at time period t then $F_{ijt}=1$ and Eqs. (30-31) are activated. These equations force the initial performance of the unit y^1_{ijt} to be equal to $y^0_{j,t-1}$ which represents the performance of unit j at the end of time period $t-1$. In the same way, if product batch i is the last one processed in unit j at time period t then $L_{ijt}=1$ and Eqs. (32-33) are activated forcing the final performance of product batch i , y^2_{ijt} , plus the percentage of recovery, to be equal to the initial performance of unit j at time period t . Notice that these constraints, Eqs. (30-33), are big-M constraints that are activated when $F_{ijt}=1$ or $L_{ijt}=1$, and the $BM=1$ since the unit performance varies between 0 and 1.

Performance of unit j at time period t is defined by Eq. (34). While Eq. (35) enforces the same performance decay at the beginning $y^0_{j,0}$ and at the end $y^0_{j,T^{\max}}$ of the planning horizon (TH). Figure 5 shows a simple example of unit performance decay and recovery. Notice that, for graphical reasons, recovery time is included as part of the whole processing time while changeover times are drawn separately.

$$y^1_{ijt} \geq y^0_{j,t-1} - (1 - F_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (30)$$

$$y^1_{ijt} \leq y^0_{j,t-1} + (1 - F_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (31)$$

$$y^2_{ijt} + \left(\frac{RT_{ijt}}{tr_j^{\max}} \right) \geq y^0_{jt} - (1 - L_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (32)$$

$$y^2_{ijt} + \left(\frac{RT_{ijt}}{tr_j^{\max}} \right) \leq y^0_{jt} + (1 - L_{ijt}) \quad \forall i, j \in IJ, t \in TH \quad (33)$$

$$y^0_{jt} = y^0_{j,t-1} - \sum_{i \in IJ} T_{ijt} \cdot d_{ij} + \sum_{i \in IJ} \left(\frac{RT_{ijt}}{tr_j^{\max}} \right) \quad \forall j, t \in TH \quad (34)$$

$$y^0_{jt} = y^0_{j,0} \quad \forall j, t = T^{\max} \quad (35)$$

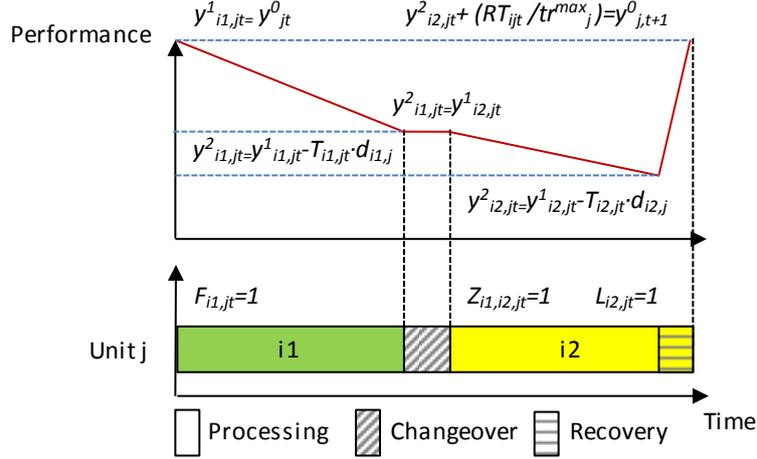


Figure 5. Unit performance decay and recovery example.

3.7. Recovery operations

A recovery operation performed at the end of the processing time of product batch i in unit j at time period t is defined by a new binary variable U_{ijt} . Eq. (36) is introduced to enforce recovery operation on product-unit that is activated during time period t . Then, Eq. (37) is proposed to limit the recovery time between a minimum and a maximum time. The maximum recovery time is the time required to recover the unit to “as good as new” state. The number of recovery operations at each time period t is limited by a maximum number defined by M_t^{max} in Eq. (38). While Eq. (39), enforces a single recovery operation for each unit j in ϕ consecutive time periods.

$$U_{ijt} \begin{cases} 1 & \text{if a recovery operation is performed on product batch } i \text{ in unit } j \text{ at time period } t \\ 0 & \text{otherwise} \end{cases}$$

$$U_{ijt} \leq E_{ijt} \quad \forall i, j \in IJ, t \in TH \quad (36)$$

$$U_{ijt} \cdot tr_j^{\min} \leq RT_{ijt} \leq tr_j^{\max} \cdot U_{ijt} \quad \forall i, j \in IJ, t \in TH \quad (37)$$

$$\sum_{ij} U_{ijt} \leq M_t^{\max} \quad \forall t \in TH \quad (38)$$

$$\sum_{i \in IJ, t' \leq t + \phi} U_{ijt'} \leq 1 \quad \forall j, t \in TH \quad (39)$$

3.8. Production amount

Production amount of product batch i at time period t is defined by PR_{it} . If product batch i is performed in unit j at that time period t then $PR_{it} = p_{ij}$, otherwise $PR_{it} = 0$ by Eq. (40). It is worth reminding that products can only be performed in one unit at any time period and the batch size is fixed and known in advance.

$$PR_{it} = \sum_{j \in IJ} p_{ij} \cdot E_{ijt} \quad \forall i, t \in TH \quad (40)$$

3.9. Product inventory

Dedicated inventory for each product batch i at any time period t is considered by V_{it} . Inventory balance constraint is defined in Eq. (41) considering the actual inventory, the production amount PR_{it} and the sales SL_{it} .

$$V_{it} = V_{it-1} + PR_{it} - SL_{it} \quad \forall i, t \in TH \quad (41)$$

3.10. Product backlog

Product backlog is considered if product batch i is not delivery on time or the amount of sales SL_{it} cannot reach the demand D_{it} at that time period. The backlog balance is defined in Eq. (42).

$$B_{it} = B_{it-1} + D_{it} - SL_{it} \quad \forall i, t \in TH \quad (42)$$

3.11. Product lifetime

Products lifetime is introduced in Eq. (43). This equation forces the total inventory of product batch i at time period t to be lower than the amount of sales of the following θ periods. The main idea is to maintain a certain level of inventory along the time avoiding huge inventory levels at some periods and nothing later on.

$$V_{it} \leq \sum_{t+\theta \geq t'} SL_{it'} \quad \forall i, t \in TH \quad (43)$$

3.12. Total profit

The total profit is stated in Eq. (44). Product prices p_i , fixed production fpc_{ij} and recovery costs fmc_{ij} , variable production vpc_{ij} and recovery vmc_{ij} unit costs, backlog cost bc_i and inventory vc_i cost are considered.

$$TP = \sum_{i,t \in TH} p_i \cdot SL_{it} - \sum_{i,j \in J, t \in TH} (vpc_{ij} \cdot T_{ijt} + fpc_{ij} \cdot E_{ijt} + vmc_{ij} \cdot RT_{ijt} + fmc_{ij} \cdot U_{ijt}) - \sum_{i,t \in TH} (bc_i \cdot B_{it} - vc_i \cdot V_{it}) \quad (44)$$

4. Benchmark example

The following example from the literature was proposed by Pinto and Grossmann (1997) and subsequently studied by Méndez and Cerdá (2002), Janak et al. (2004) and Marchetti and Cerdá (2009) for a plastics compounding facility. In here, 12 products must be produced over a time horizon of 30 days in a single-stage process with four Extruders working in parallel. Specific processing times, due-dates and setup times can be found in Pinto and Grossmann (1997). This problem considers that a single worker is required per unit for processing. Thus, three versions are derived, a) an unlimited version with 4 workers, b) a limited version with 3 workers and c) a limited version with 2 workers. The aim of this problem is to minimise the overall earliness OE defined by the following expression in Eq. (45) for a single time period with specific due-dates dd_i . But originally in Pinto and Grossmann (1997) the problem was solved by maximizing the summation of the starting times.

$$OE = \sum_i dd_i - TF_{it} \quad \forall i, t \quad (45)$$

This example was solved efficiently by other formulations in the past. The idea is to compare the statistics provided by our model with the statistic reported by the existing approaches for this simple example in Table 1. For this, Eq.(3) has been modified in order to force the model to produce all product orders and Eq. (9) has been adapted in the way to adopt a nominal processing time defined by tp_{ij} . The resulting formulation considers the main assignment, sequencing, timing, and share resources constraints by Eqs. (1,2,3bis,4-8,9bis,10-13,24-27). The objective function is defined by Eq. (45) considering that $TF_{it} \leq dd_i$.

$$\sum_{j \in J} E_{ijt} = 1 \quad \forall i, t \in TH \quad (3bis)$$

$$TF_{it} = TS_{it} + \sum_{j \in J} tp_{ij} \cdot E_{ijt} \quad \forall i, t \in TH \quad (9bis)$$

Despite of using more variables and constraints than other formulations, our model is able to solve these problem instances very well, reporting the same optimal solution for each instance with short CPU time. It is worth remarking than this specific problem does not consider performance decay and only takes into account setup times which are independent from the product sequence. Our model was formulated in a general way, allowing coupling with sequence-dependent issues, limited resources and performance degradation. That is the reason why extra variables and constraints are considered in our TSP-based and precedence-based formulation. The introduction of changeover times or performance decay may affect the performance of the existing formulations. The following motivating problem is proposed to study what is happening when a sequence-dependent issues are introduced.

Table 1. Model size and statistics for a benchmark case study

Model	Case	Discrete, Variables & Eqs.	Objective Function	CPU time(s)	Nodes
Pinto and Grossmann 1997 ¹	a	-	269.10 [#]	63.56	283
	b	-	268.24 [#]	125.42	673
	c	-	264.98 [#]	927.16	7341
Méndez and Cerdá 2002 ²	a	82, 24, 214	1.026	0.03	12
	b	127, 24, 622	1.895	2.93	4372
	c	115, 24, 490	7.334	13.28	21583
Janak et al 2004 ³	a	150, 513, 1389	1.026	0.07	7
	b	458, 2137, 10382	1.895	6.53	1374
	c	444, 2137, 10382	7.334	236.37	38621
Marchetti and Cerdá 2009 ²	a	82, 24, 214	1.026	0.03	12
	b	223, 24, 622	1.895	0.66	121
	c	223, 24, 622	7.334	7.39	3943
Proposed TSP/precedence-based ⁴	a	191, 75, 299	1.026	0.1	0
	b	389, 87, 587	1.895	0.3	475
	c	389, 87, 587	7.334	1.37	3447
	a	191, 75, 299	269.69 [#]	0.3	213
	b	389, 87, 587	268.82 [#]	0.47	72
	c	389, 87, 587	265.97 [#]	0.62	2761

¹ solution reported in Pinto and Grossmann 1997 by using IBM 6000-530 w/GAMS/OSL.

² solutions reported in Marchetti and Cerdá 2009 by using an ILOG OPL Studio 3.6 with CPLEX v. 8.0 on a Pentium IV 1.8 GHz.

³ solutions reported in Janak et al. 2004 by using GAMS 2.5 with CPLEX 8.1 on a Linux Workstation 3 GHz.

⁴ solutions reported by using AIMMS 4.3 with CPLEX 12.6 in an Intel Xeon 3.5 GHz.

[#] maximising the summation of starting times = $\sum((i,t), TS_{it})$.

5. Motivating problem

This motivating problem is a modified version of the one published in Cerdá et al. (1997). Thus, a maximum of 10 products $I=(i1 \dots i10)$ should be processed at 4 non-identical processing units $J=(j1 \dots j4)$, taken into account constant linear performance decay d_{ij} with limited resources R_t^{\max} and limited number of recovery operations M_t^{\max} per time period t . Fixed product batch size p_{ij} , nominal processing times tp_{ij} and changeover times τ_{ij} are shown in Table 2. Maximum recovery time tr_j^{\max} per unit j is known in advance. Specific demand is given by D_{it} . The fixed product batch size p_{ij} in Eq. (18) is replaced by the demand D_{it} , in order to represent the real size of the batch to be produced. Processing times tp_{ijk} at each breakpoint k for unit j is calculated offline by the adapted Eq. (18) where production rate is given by $r_{ij} = (p_{ij}/tp_{ij})$ and y_{ijt}^l is replaced by $y_k = (|K|-k+1)/|K|$ for each breakpoint k . For the case of no performance decay ($d_{ij}=0$), Eq. (18) is not used and tp_{ijk} is calculated by D_{it}/r_{ij} .

Table 2. Fixed product batch size [kg/batch], processing time [days/batch] and changeover times [days]

p_{ij}	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$i7$	$i8$	$i9$	$i10$
$j1$	100	-	140	-	-	280	-	-	200	250
$j2$	-	-	-	120	90	210	-	-	-	270
$j3$	-	210	170	-	-	-	390	-	-	-
$j4$	-	-	-	-	130	-	290	120	-	-
tp_{ij}	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$i7$	$i8$	$i9$	$i10$
$j1$	1.7	-	1.25	-	-	2.4	-	-	1.6	2.6
$j2$	-	-	-	1.7	1.4	1.8	-	-	-	1.9
$j3$	-	0.9	1.1	-	-	-	1.05	-	-	-
$j4$	-	-	-	-	0.85	-	1.65	2.1	-	-
τ_{ij}	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$i7$	$i8$	$i9$	$i10$
$i1$	0.0	-	0.0	-	-	0.65	-	-	0.9	0.4
$i2$	-	0.0	1.1	-	-	0.0	0.0	-	-	-
$i3$	1.0	0.15	0.0	-	-	0.0	0.3	-	1.6	0.2
$i4$	-	-	-	-	0.05	0.0	-	-	-	0.5
$i5$	-	-	-	0.3	0.0	0.7	0.9	0.6	-	0.0
$i6$	1.4	-	0.3	0.7	0.0	0.0	-	-	1.2	0.0
$i7$	-	1.8	0.0	-	0.85	-	0.0	0.45	-	-
$i8$	-	-	0.0	-	0.0	-	1.65	0.0	-	-
$i9$	2.1	-	1.25	-	0.0	0.8	-	-	0.0	0.65
$i10$	1.5	-	0.6	0.8	0.5	0.0	-	-	0.7	0.0

Three example problems (Examples 1, 2, 3) are presented in Table 3 considering 8, 9, 10 product orders, respectively. These examples are used for solving single and multi-period cases presented below. All the problems are solved using CPLEX® v12.6 on an Intel® Xeon® CPU 3.5GHz with 12 parallel threads.

Table 3. Demands [kg] for Examples 1-3

D_{it}	Example 1	Example 2	Example 3
$i1$	700	700	550
$i2$	1050	850	850
$i3$	900	900	700
$i4$	1000	900	900
$i5$	650	500	500
$i6$	1350	1350	1050
$i7$	950	950	950
$i8$	850	850	850
$i9$	-	450	450
$i10$	-	-	650

5.1. Single period problem (Examples 1-3)

Results for Examples 1-3 are reported in Table 4-5. These examples were proposed in Cerdá et al. (1997) for a single time period without performance decay by considering makespan MK as an objective function, which in our model is defined by Eq. (46). Our proposed formulation is run by considering Eqs. (1,2,3bis,4-8,9bis,10-13,24-27,46). In the case of no performance decay, the value of tp_{ij} in Eq.(9bis) will be equal to tp_{ijk} . Note that production amount, inventory, backlogs and lifetime constraints are not considered here. In order to compare with existing formulations, a discrete-time STN formulation of Kondili et al. (1993) and a unit-specific time event STN formulation of Janak et al. (2004) have been implemented.

$$MK \geq TF_{it} \quad \forall i, t = 1 \quad (46)$$

The results provided in Table 4 for Examples 1-3 confirm the values (MK) reported by Cerdá et al. (1997) for a single time period problem without considering performance decay. These values 24.55, 26.90 and 23.9 [days] were obtained by our formulation in less than 0.25s of CPU time. The global-time event model reaches the same solutions in 1.4s, 2.9s, 7.8s, respectively by considering 4, 4, 5 global events. Solutions reported by discrete-time STN model have some limitations due to, discrete processing, changeover and recovery times. The discrete-time model assumes a maximum recovery time tr_j^{\max} , even if the unit is not recovered to the maximum level. Processing times and changeover times are also rounded to the next integer value. Due to these limitations the discrete-time STN model only could find makespan results of 25, 27, 24 [days] in few seconds by using a daily grid (30 points).

Table 4. Results of STN vs TSP models for Examples 1-3 by using $R^{\max}=4$, $M^{\max}=4$ and $d_{ij}=0$

Example 1	STN-discrete	STN-continuous	TSP-continuous
Time points	30	3	4
Equations	51432	700	1031
Variables	3871	322	4269
Discrete	3720	72	96
Makespan	25	24.55	24.55
CPU time (s)	1.0	0.47	1.4
Example 2	STN-discrete	STN-continuous	TSP-continuous
Time points	30	3	4
Equations	69434	769	5638
Variables	4171	352	469
Discrete	4020	78	104
Makespan	27	26.9	26.9
CPU time (s)	1.5	0.45	2.9
Example 3	STN-discrete	STN-continuous	TSP-continuous
Time points	30	4	5
Equations	96436	1366	1894
Variables	4771	533	666
Discrete	4620	120	150
Makespan	24	24.2	23.9
CPU time (s)	2.4	2.7	7.8

Table 5 shows the results for a similar problem by considering limited resources $R^{\max}=3$. The addition of limited resources does not affect much the convergence of our model, providing optimal results for all cases in less than 1 second. On the other hand, the global time event STN formulation requires more events and CPU time to guaranty optimal solution. For example, best solution found for Example 3 was 26.9 [days] by using 6 global time points after 203 seconds when the optimal solution is 26.35 [days]. More events may produce better solutions but at expenses of much more CPU time. The discrete-time STN model behaves similar than our formulation, providing good near optimal solutions in less than 3 seconds.

Table 5. Results of STN vs TSP models for Examples 1-3 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0$

Example 1	STN-discrete	STN-continuous	TSP-continuous
Time points	30	5	6
Equations	51432	1410	1837
Variables	3871	536	643
Discrete	3720	120	144
Makespan	26	26.85	25.55
CPU time (s)	1.4	11	81
Example 2	STN-discrete	STN-continuous	TSP-continuous
Time points	30	5	6
Equations	69434	1565	2047
Variables	4171	586	703
Discrete	4020	130	156
Makespan	27	27.4	26.9
CPU time (s)	1.6	14	30
Example 3	STN-discrete	STN-continuous	TSP-continuous
Time points	30	5	6
Equations	96436	1894	2496
Variables	4771	666	799
Discrete	4620	150	180
Makespan	27	27.65	26.9
CPU time (s)	2.1	28	203

The third problem instance is reported in Table 6 by assuming limited resources $R^{max}=3$ and considering constant performance decay $d_{ij}=0.01$. In here, our formulation is proposed by considering Eqs. (1-14,20-39,46). The existing formulations have been adapted to consider unit performance decay by d_{ij} with limited resources R_t^{max} and recovery operations M_t^{max} . Details about adapting models are summarised in Appendix A section.

Table 6. Results of STN vs TSP models for Examples 1-3 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0.01$

Example 1	STN-discrete	STN-continuous	TSP-continuous
Time points	35	5	6
Equations	60570	1687	1926
Variables	4660	720	839
Discrete	4340	140	168
Makespan	31	29.56	28.76
CPU time (s)	10	19	56
Example 2	STN-discrete	STN-continuous	TSP-continuous
Time points	35	4	5
Equation	81572	1377	1858
Variables	5010	651	780
Discrete	4690	120	150
Makespan	34	32.11	32.11
CPU time (s)	14	8	32
Example 3	STN-discrete	STN-continuous	TSP-continuous
Time points	35	5	6
Equations	113074	2219	2882
Variables	5710	880	1025
Discrete	5390	170	204
Makespan	31	31.08	30.90 (11%)
CPU time (s)	45	110	600*

* CPU time limit = 600 seconds. Relative gap in brackets.

Results in Table 6 show the advantages of the proposed formulation in finding optimal solutions for all the cases in 0.4, 2 and 5 seconds while continuous-time STN spends 56, 32 and 600 seconds and discrete-time STN takes 10, 14 and 45 seconds of CPU time, respectively.

The superior performance of CPU time, of at least one order of magnitude in comparison with other existing models, on a simple case study considering a single time period problem, allow us to infer that a greater difference will occur when multiple time periods would be considered due to the increasing number of time points that should be taken into account in the STN formulations. That is the main reason why we have decided to prioritize the use of the proposed continuous-time formulation for the rest of the study.

The following section provides multi-period problems based on the information presented in Example 4. These new cases have been proposed for the integration of planning, scheduling and maintenance decisions. Planning decisions as production amount, inventory and backlog levels and sales are introduced. Thus, new parameters as product prices, production costs, inventory and backlog costs are also taken into account so as to maximise the total profit stated by Eq. (44).

5.2. Multi-period problem (Example 4)

A multi-period problem is proposed and three cases are solved by different planning horizon a) $TH=4$, b) $TH=6$ and c) $TH=8$, respectively. This new example considers fixed product prices by p_i and fixed and variable production costs by fpc_{ij} and vpc_{ij} . Backlog cost and inventory costs are assumed in 50% and 30% of the product prices, respectively. Maintenance costs are defined by fmc_{ij} and vmc_{ij} . Demand profile D_{it} in Table 7 is generated for all the time periods by using the information of Example 1. Table 8 shows information on product prices and costs. Variables costs are assumed in $vpc_{ij}=100$ [\$/day] and $vmc_{ij}=1000$ [\$/day]. No limits for inventory capacity are considered. The elapsed time for consecutive recovery operations in the same unit is defined by $\phi=2$ time-periods. The maximum recovery time is defined by $tr^{max}=5$ [days] while the maximum number of resources in $R^{max}=3$, the maximum number of recovery operations in $M^{max}=4$ and the degradation in $d_{ij}=0.005$. Results for Example 4 are summarised in Table 9.

Table 7. Multi-period demand [kg] of Example 4

D_{it}	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$	$t7$	$t8$
$i1$	-	700	-	700	-	700	-	700
$i2$	1050	1050	1050	1050	1050	1050	1050	1050
$i3$	900	900	900	900	900	900	900	900
$i4$	1000	-	1000	-	1000	-	1000	-
$i5$	650	-	650	-	650	-	650	-
$i6$	-	1350	-	1350	-	1350	-	1350
$i7$	950	950	950	950	950	950	950	950
$i8$	850	850	850	850	850	850	850	850
Total	5400	5800	5400	5800	5400	5800	5400	5800

Table 8. Product prices and costs [\$/] of Example 4

	p_i	fpc_{ij}	fmc_{ij}
$i1$	7	980	2450
$i2$	2	210	420
$i3$	5	450	900
$i4$	3	300	600
$i5$	8	520	1040
$i6$	1	135	270
$i7$	4	380	760
$i8$	6	510	1020

Table 9 shows the main results for different cases of Example 4 considering 4, 6 and 8 time-periods (T^{max}). Notice that in all the cases the first integer solution found by the monolithic model has been provided in few seconds while spending a few minutes to improve the initial result. These cases haven't been solved to optimality and the solutions have been reported within 5% of relative gap.

Table 9. Model statistics of Example 4 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0.005$

Example 4	Case	T^{max}	Discrete, Variables & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	4	528, 985, 2341	55923	1	40430	12	4.9
	b	6	792, 1475, 3521	83885	1	63206	124	5.0
	c	8	1056, 1965, 4701	111846	1	86315	700	4.7

Table 10 shows the comparison between the solutions obtained within 5% of relative gap and the corresponding one found by the model in a time limit imposed of 1000 seconds. As it can be seen, the solutions obtained after 1000 seconds of CPU time, improve the one reported in Table 9 in less than 1% of all the cases compared in Table 10 and only in one of these cases the relative gap has been reduced to 1%. Thus, it seems to be unnecessary to impose a time limit of 1000 seconds as a stopping criterion when the model has been able to find a good quality result (<5% gap) in less CPU time.

Table 10. Solution comparison for Example 4

Example 4	Case	T^{max}	Best Integer Solution	CPU time (s)	Gap%	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	4	40430	12	4.9	40741	1000	0.1
	b	6	63206	124	5.0	63213	1000	2.9
	c	8	86315	700	4.7	86342	1000	4.3

Revenues and costs of the solutions reported in Table 9 are reported in Table 11. Detailed scheduling and unit performance of Example 4a and Example 4c are shown in Figure 6a and Figure 6b. In here, units are shown in y-axis while time in days is represented in the x-axis. In the unit performance graph, unit performance is shown on y-axis while time is shown on the x-axis.

Table 11. Results of Example 4

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 4	a	40430	33820	9032	5877	580	89740
	b	63206	46503	12475	8496	2370	133050
	c	86315	63783	16666	10476	3780	181020

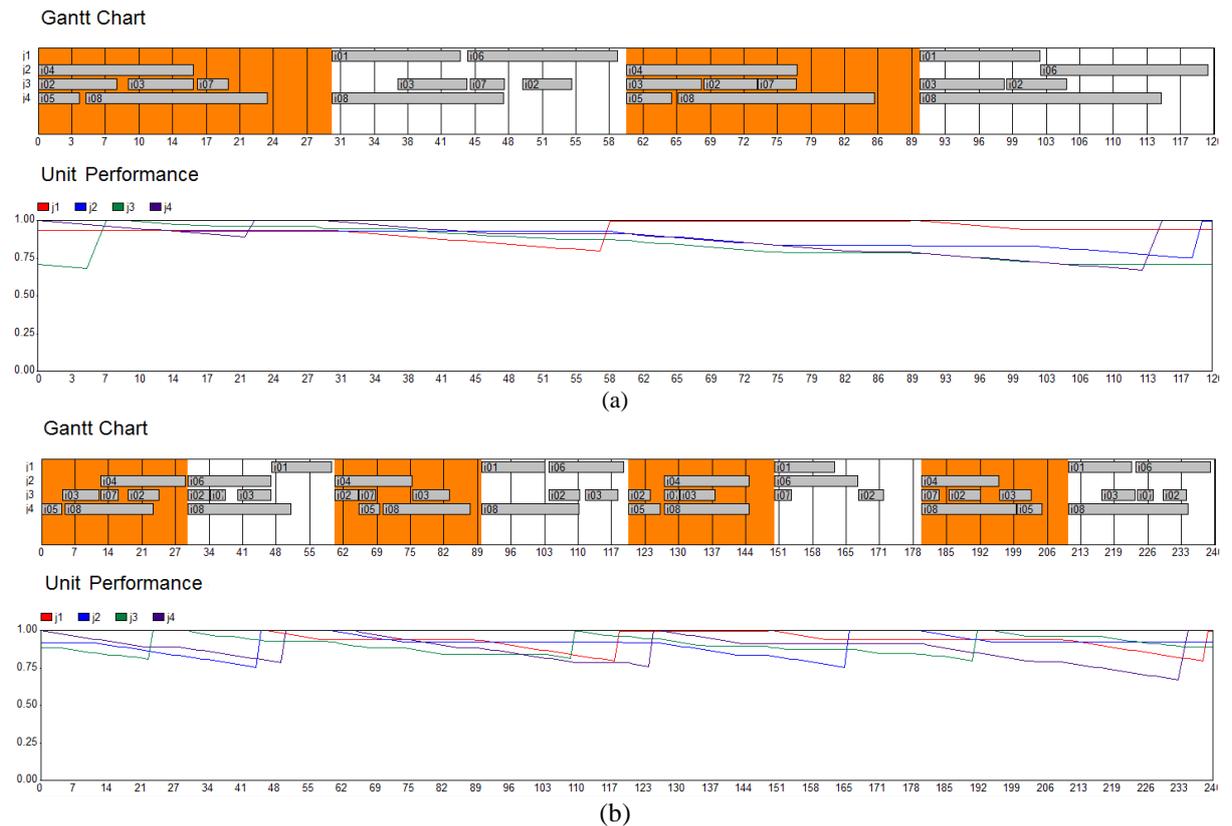


Figure 6. Gantt chart and unit performance decay along the time (days) for a) Example 4a and b) Example 4c

These examples have been solved assuming a fixed demand with a specific pattern that it is repeated every two-time period. Thus, analysing the solution schedule of Figure 6a and Figure 6b it seems possible to obtain a solution for the first two time-periods and then use this schedule to build a solution for the whole planning horizon on Examples 4a, 4b, and 4c. Or, use the solution obtained from Example 4a to build a solution for Example 4c. This kind of rolling horizon technique is very common to solve large planning problems especially when the schedule is repeated every fixed time period (cyclic schedule). Despite of this, the final solution reported by the model changes at every time period and the overall profit is improved in comparison with the estimated solution by repeated the same schedule many times. This happens mainly due to the restrictions imposed over unit performance and recovery operations stated by Eqs. (28-39). These recovery operations are taking place after ϕ time periods restricting the occurrence of successive recovery operations at consecutive time periods. It is worth noticing that each unit operates at certain performance levels, allowing flexible maintenance operations that recover the unit to an intermediate level, not necessarily to “as good as new” level. It is important to remark that each unit start and finish at specific levels at the beginning and at the end of the planning horizon, ensuring unit availability at the beginning of the next planning horizon. These limitations complicate the resolution of the system, forcing the model to create scheduling solutions that a priori seems to be very inefficient in terms productivity, leaving extensive idle times. However, the solution found by the model allows to improve the overall profit by considering more flexible decisions regarding maintenance operations, ensuring at the same time a reliable schedule in terms of unit and resource availability for the current and the next planning periods.

5.3. Scenario analysis

The following analysis is provided to study the behaviour of the solution to different values of main problem parameters as, performance decay d_{ij} , demand D_{it} , inventory capacity θ , elapsed time between recovery operations ϕ and maximum recovery time tr^{max} . This analysis is provided to understand how sensible the proposed model is according to different state conditions. Thus, five scenarios are proposed by changing some of the main deciding factors (see Table 12).

Table 12. Scenario analysis

Scenario	Example	Problem	Parameter	Model Statistics	Results	Figures
1	Example 5	Example 4a	d_{ij}	Table 13	Table 14	Figure 7
2	Example 6	Example 4a	D_{it}	Table 16	Table 17	Figure 8
3	Example 7	Example 4b	θ	Table 18	Table 19	Figure 9
4	Example 8	Example 4b	ϕ	Table 20	Table 21	Figure 10
5	Example 9	Example 4b	tr^{max}	Table 22	Table 23	Figure 11

5.3.1. Scenario 1

Example 5 is presented here to evaluate the impact of the performance decay factor d_{ij} on the solution and the performance of the proposed model. Here a more intense performance degradation d_{ij} is tested by using Example 4a for the comparison. Thus, the original Example 4a represented by case a) $d_{ij}=0.005$ is compared with case b) and c) that use respectively $d_{ij}=0.007$ and $d_{ij}=0.010$. Case d) is proposed by using a uniform distribution between 0.005 and 0.015.

Table 13. Model statistics of Example 5 by using $R^{max}=3, M^{max}=4$

Example 5	Case	d_{ij}	Discrete, Variables. & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	0.005	528, 985, 2341	55923	1	40430	12	4.9
	b	0.007	528, 985, 2341	53295	1	35156	36	5.0
	c	0.010	528, 985, 2341	49178	1	28931	235	5.0
	d	U(0.005,0.015)	528, 985, 2341	51355	1	23393	900	5.0

Results demonstrate that more intensive performance decay really affects the efficiency and the final solution reported by the model. Total profit has decreased by 13%, 28% and 42% for cases b), c) and d) in comparison with original case. The computational time has also been affected. More intensive performance decay factor more CPU time is required. Despite of that, the model has been able to converge to a good quality solution (<5% gap) in less than 1000 seconds in all cases analysed. Table 14 shows the results in terms of revenues and costs for Example 5. Here it is easy to appreciate that, more decay produces larger processing times, which affect directly to the production amount, generating more backlogs and also reducing the revenues and the total profit.

Table 14. Results of Example 5

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 5	a	40430	33820	9032	5877	580	89740
	b	35156	31636	11584	6000	2175	86550
	c	28931	31587	15021	4425	5355	85320
	d	23393	32906	18418	5214	5390	85320

Figure 7a shows the schedule for the more intensive performance decay in Example 5c. Notice that unit's performance deteriorates quickly in comparison with Example 4a (Figure 6a), requiring an immediate recovery operation after few time periods. Figure 7b shows the impact of having different performance decay per products and units. This aspect affects much the schedule, creating totally different production schedules per time period, anticipating or delaying the execution of certain products orders and also spending much time recovering the units.

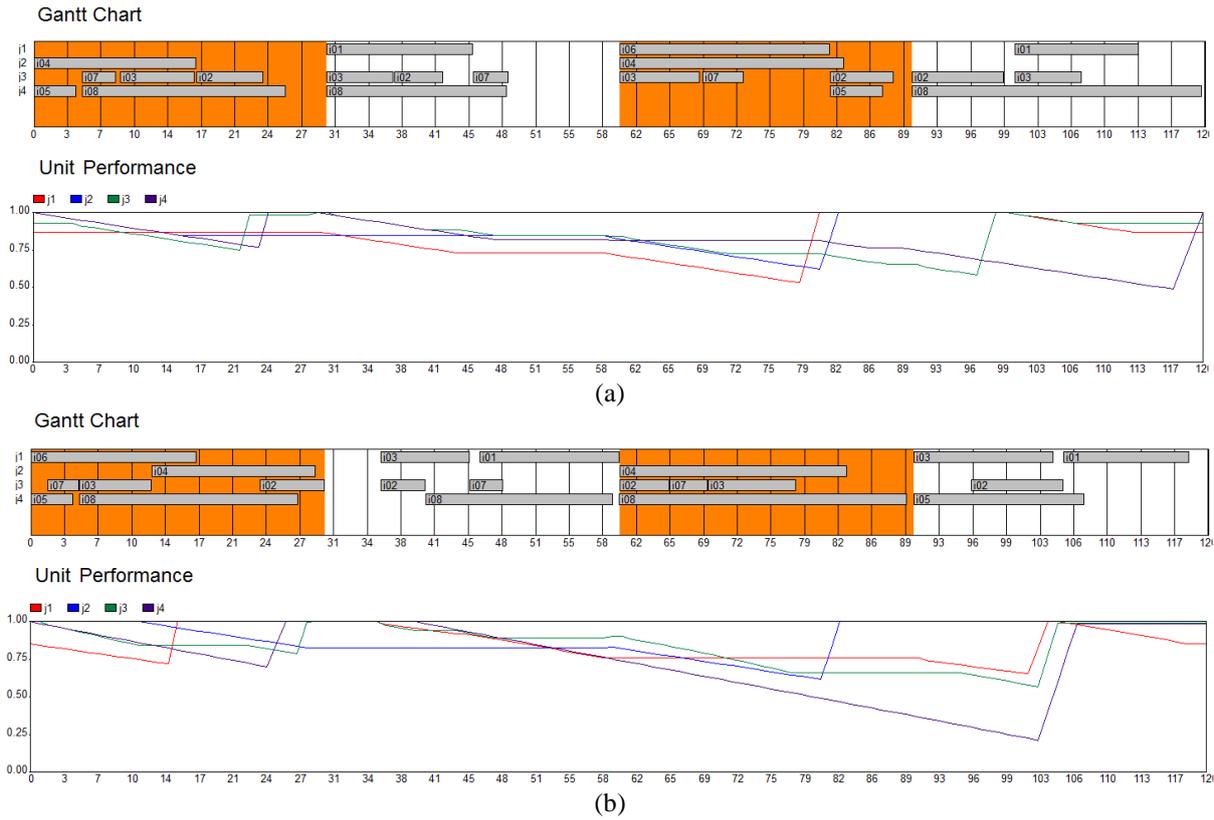


Figure 7. Gantt chart and unit performance decay along the time (days) for a) Example 5c and b) Example 5d

5.3.2. Scenario 2

The following example, Example 6, is proposed to evaluate the impact of different demand profiles. The original case study is provided in Example 6a based on the demand profile shown in Table 7, the same used for Example 4. Then, two additional cases are introduced for the comparison, case b) by using the demand profile shown in Table 15 and case c) by using the original demand profile of Example 1 provided in Table 3 for all time periods. Results and comparisons are shown in Tables 16-17. Table 16 shows the model performance and statistics while Table 17 shows the main results for Example 6.

Table 15. Multi-period demand [kg] of Example 6

D_{it}	$t1$	$t2$	$t3$	$t4$
$i1$	-	1400	-	1400
$i2$	1050	1050	1050	1050
$i3$	900	900	900	900
$i4$	2000	-	2000	-
$i5$	1300	-	1300	-
$i6$	-	2700	-	2700
$i7$	950	950	950	950
$i8$	850	850	850	850
Total	7050	7850	7050	7850

Table 16. Model Statistics of Example 6 by using $R^{max}=3, M^{max}=4, d_{ij}=0.005$

Example 6	Case	D_{it}	Discrete, Variables & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	Table 7	528, 985, 2341	55923	1	40430	12	4.9
	b	Table 15	528, 985, 2341	58518	1	39959	517	5.0
	c	Table 3	528, 985, 2341	70048	1	49353	980	5.0

Table 17. Results of Example 6

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 6	a	40430	33820	9032	5877	580	89740
	b	39959	42553	11641	8868	10740	113760
	c	49353	44772	12752	5583	2840	115300

The final solution for Example 6b, 39959 [\$], was reached after 517 seconds. Notice that increasing a half the total demand, the total revenue only has been increased by 26%, while the production cost and maintenance cost in 25% and 28%, respectively while the inventory cost and backlog cost have grown many times. Basically, despite of having more revenues the total profit remains almost the same than the original case in Example 6a.

Example 6c is proposed by considering a dense profile by assuming that all products are produced at every time period following the demand in Table 3 – Example 1. Note that, in both cases, case b) and case c), the total demand per time period is 29800 [kg] with an average value of 7450 [kg].

The total profit reached for Example 6c, 49353 [\$], was 22% higher than the original profit of Example 6a. Here the revenues have been increased by 28% while production and maintenance costs have been 32% and 41% higher, respectively. Despite of that, inventory costs were reduced and backlogs were only few times bigger than the original values. This case represents an intermediate situation between cases a) and b), in where most of the product demand can be delivered on time.

A detailed Gantt chart for this case is shown in Figure 8. Here it can be seen that most of the products are produced every time period leaving less idle time. Due to this, the model has tried to produce in advance in order to have some inventory to serve the demand of the following periods. Despite of that, the model cannot serve the demand on time in most of the time period and part of the demand is served later as backlogs.

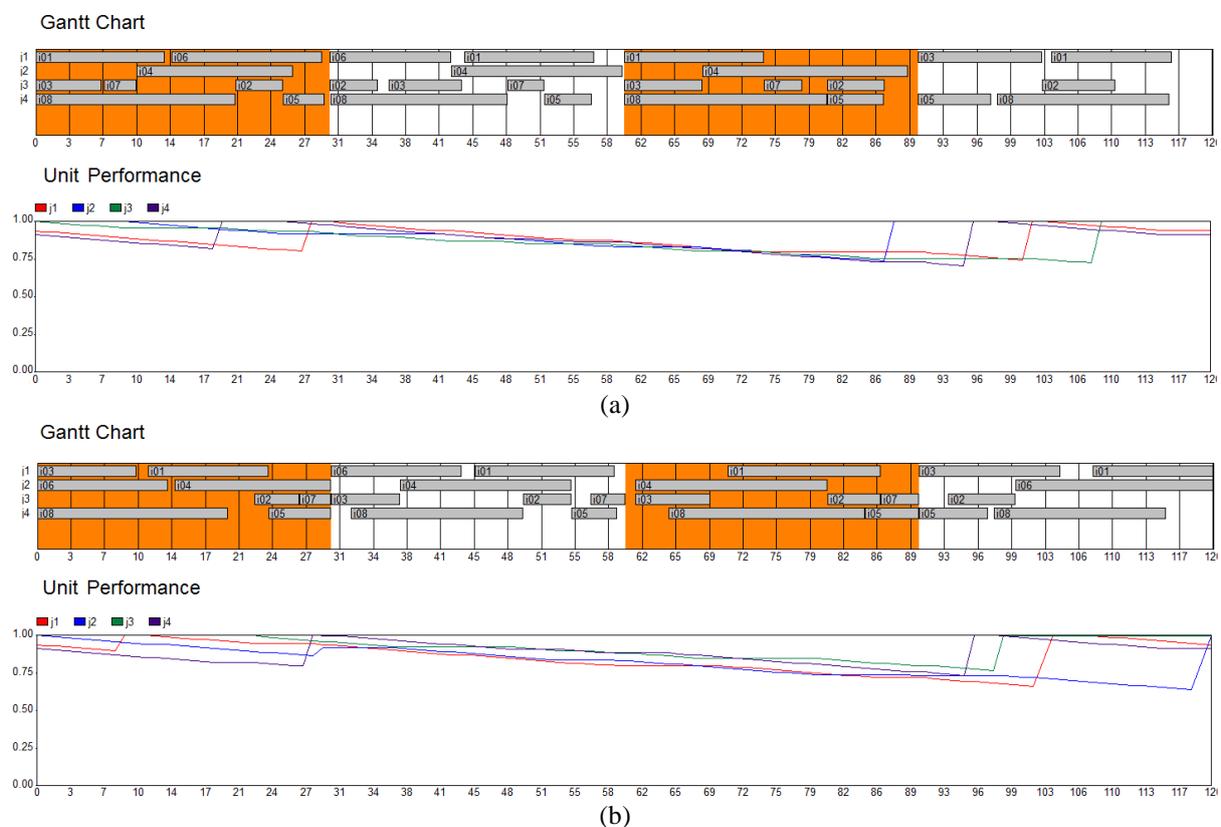


Figure 8. Gantt chart and unit performance decay along the time (days) for a) Example 6b and b) Example 6c.

5.3.3. Scenario 3

Example 7 is proposed to study the behaviour of the model with limited inventory capacity by changing product lifetime parameter. For this, Example 4b is used and the model is run by adopting different values of θ factor in Eq. (43). First, for case a), we have assumed that there are no limitations on product lifetime, and the results have been the same as the ones in Example 4b. Then, in case b), we have considered $\theta=2$ and then more restrictive in case c) by considering $\theta=1$. Results for Example 7 are shown in Table 18 and Table 19.

Table 18. Model statistic of Example 7 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0.005$

Example 7	Case	θ	Discrete, Variables & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-44)	a	-	792, 1475, 3521	83885	1	63206	124	5.0
	b	2	792, 1475, 3569	83885	1	62174	75	5.0
	c	1	792, 1475, 3569	83885	1	61759	52	5.0

Table 19. Results of Example 7

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 7	a	63206	46503	12475	8496	2370	133050
	b	62174	47057	12373	8676	2770	133050
	c	61759	46583	12514	8994	3200	133050

Results demonstrate that product lifetime parameter θ does not affect much the final solution of the original problem presented in Example 7a but improves the behaviour of the model, speeding up the convergence of the model until 5% gap. The total profit for Example 7a was reduced only 1.6% and 2.3%, respectively for cases b) and c) due to higher production costs (1.19% and 0.17% higher), inventory costs (2.12% and 5.86% higher) and basically more backlog costs (16.87% and 35.02% higher) associated with the less anticipated demand. Figure 9 shows the details about the schedule and the unit performance for Examples 7b and 7c.

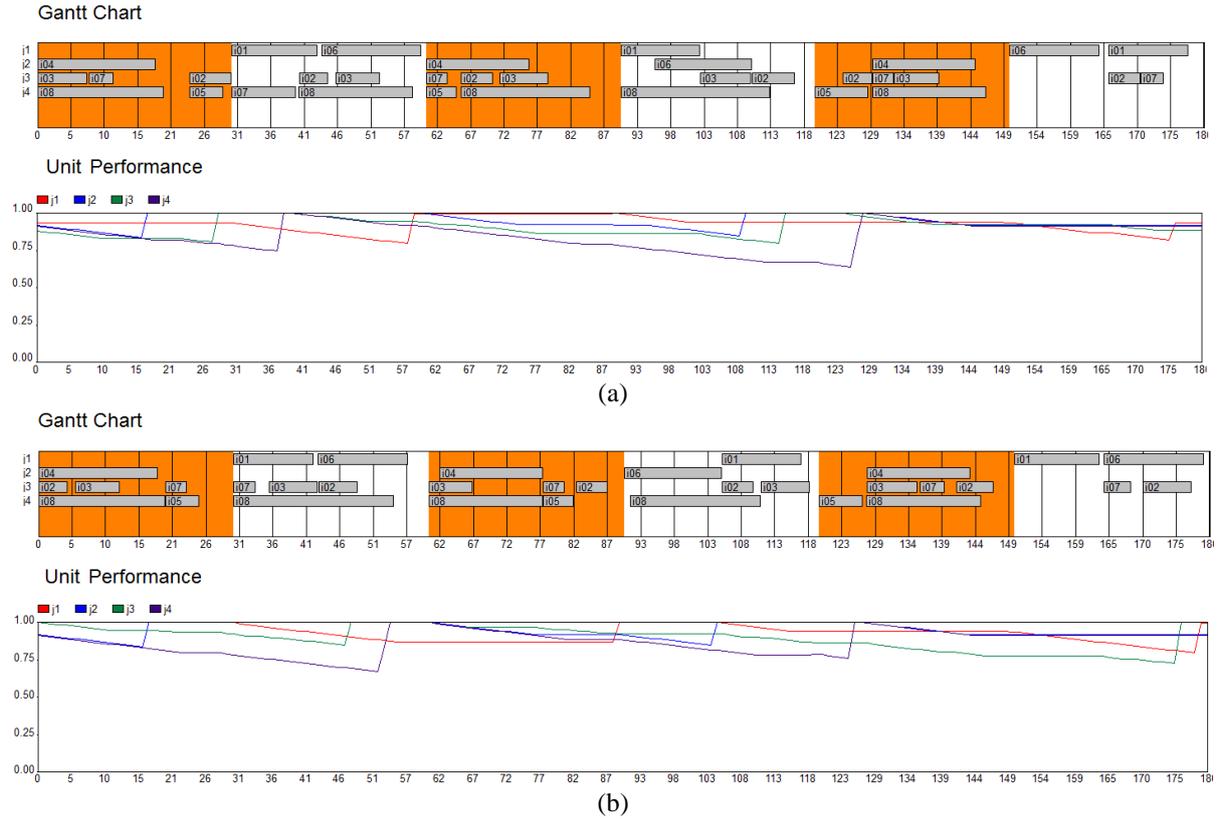


Figure 9. Gantt chart and unit performance decay along the time (days) for a) Example 7b and b) Example 7c

5.3.4. Scenario 4

Example 8 is proposed by changing the ϕ parameter, which enforced a number of elapsed time periods for consecutive recovery operations. For this example, we have considered Example 4b as reference for comparison by using $\phi=2$ as case a). Then, two additional cases have been proposed by considering $\phi=4$ and $\phi=6$ in case b) and case c), respectively. Main model statistics and results are presented below in Tables 19 and 20.

Table 20. Model statistics of Example 8 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0.005$

Example 8	Case	ϕ	Discrete, Variables & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	2	792, 1475, 3521	83885	1	63206	124	5.0
	b	4	792, 1475, 3521	83885	1	58882	938	5.0
	c	6	792, 1475, 3521	83885	1	51840	1000	5.4

Table 21. Results of Example 8

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 8	a	63206	46503	12475	8496	2370	133050
	b	58882	47818	12223	7926	6200	133050
	c	51840	47753	10988	7704	7755	126040

Results for case b) has shown a decrease of 6.85% on the total profit in comparison with original case in a) by performing at most two recovery operations per unit along over the planning horizon (see Figure 10a). Case c) has been solved allowing a single recovery operation for the whole planning horizon (see Figure 10b). The total profit for case c) has been reduced by 17% from the original case. In both cases backlog costs have increased and most of the products could not be served on time, which have generated less revenues. These cases demonstrate the impact of ϕ factor over the total profit while Figure 10 shows how this factor affects the schedule, reducing the number of batches produced, and reducing the average performance of the units over time.

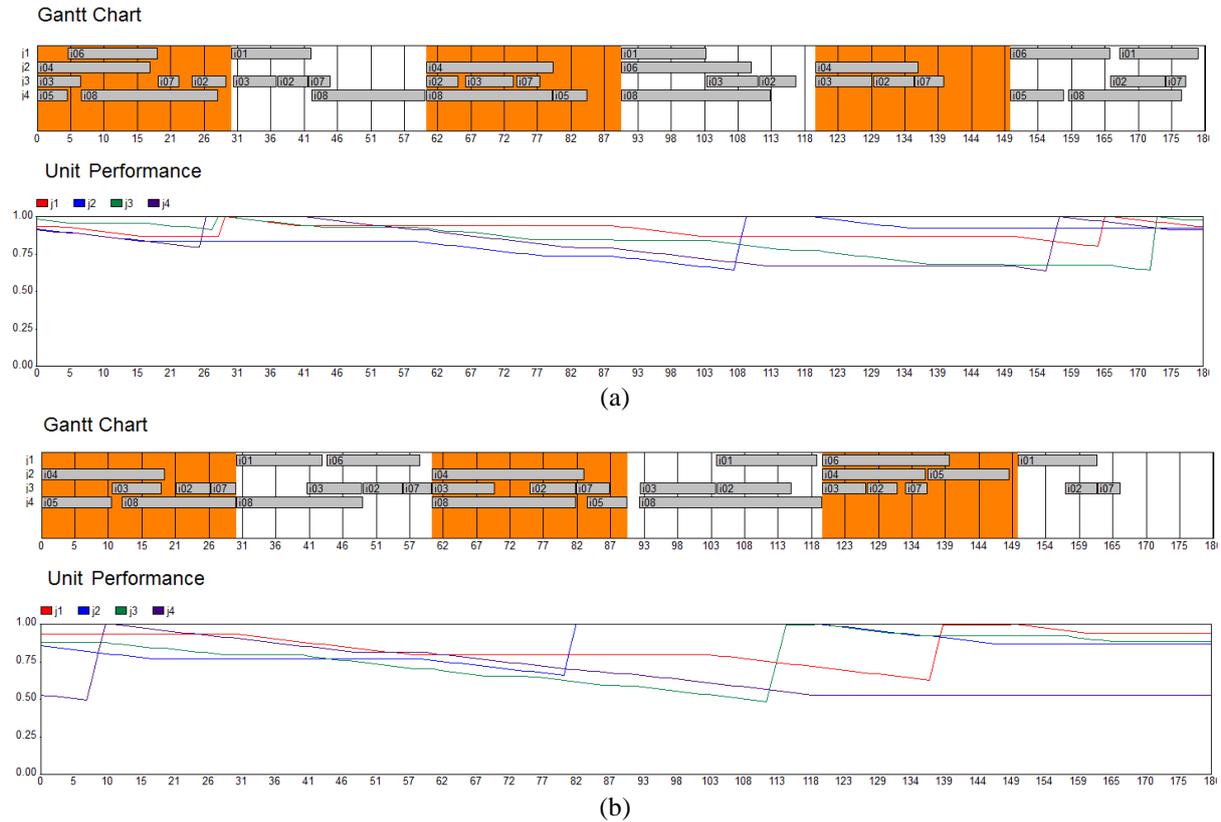


Figure 10. Gantt chart and unit performance decay along the time (days) of a) Example 8b and b) Example 8c

5.3.5. Scenario 5

An additional example is provided, in Example 9, to evaluate the impact of the maximum recovery time tr^{max} . For this, Example 9a is proposed by using the information of Example 4b in where $tr^{max}=5$. Then two extra cases are created in b) and c) by considering $tr^{max}=10$ and $tr^{max}=15$, respectively. Results are shown in Tables 22-23.

Table 22. Model statistics of Example 9 by using $R^{max}=3$, $M^{max}=4$ and $d_{ij}=0.005$

Example 9	Case	tr^{max}	Discrete, Variables & Eqs.	Best LP Solution	CPU time (s)	Best Integer Solution	CPU time (s)	Gap%
Proposed formulation Eqs. (1-14,20-42,44)	a	5	792, 1475, 3521	83885	1	63206	124	5.0
	b	10	792, 1475, 3521	76346	1	54930	360	5.0
	c	15	792, 1475, 3521	69016	1	46812	800	5.0

Table 23. Results of Example 9

	Case	Total Profit	Production cost	Maintenance cost	Inventory cost	Backlog cost	Revenues
Example 9	a	63206	46503	12475	8496	2370	133050
	b	54930	46565	20690	8496	2370	133050
	c	46812	45365	27841	8892	2960	131870

Table 22 and Table 23 show the high impact of tr^{max} over the total profit. Example 9b and Example 9c show a decrease of 13% and 26% in comparison with case a). Higher values of maximum recovery time forces higher recovery times RT_{ij} . Thus, associated maintenance costs are increased and some extra costs may also occur due

to higher backlogs or lower revenues. Figure 11 shows detailed schedule and unit performance for Example 9b and Example 9c.

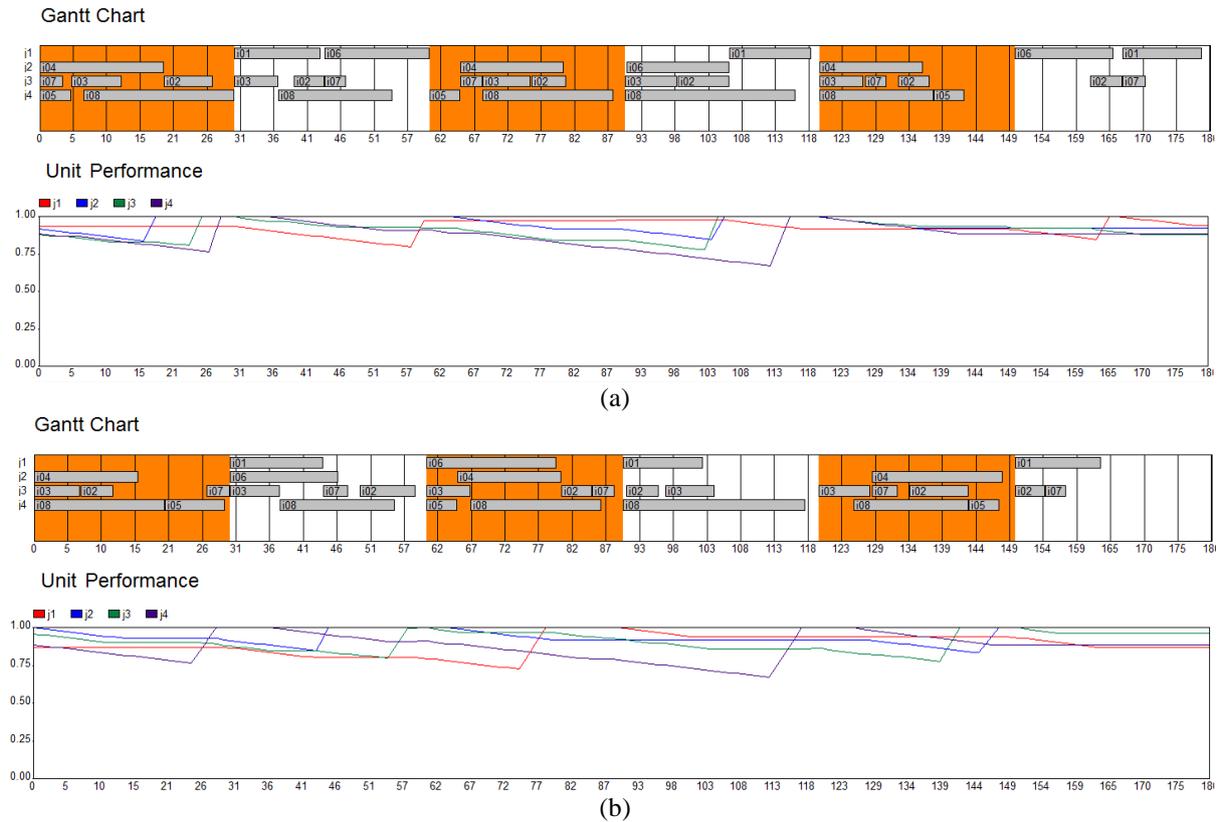


Figure 11. Gantt chart and unit performance decay along the time (days) for a) Example 9b and b) Example 9c

6. Concluding Remarks

A novel MILP model for the integration of production planning scheduling and maintenance in a multi-product single-stage process plant with parallel units, limited resources and sequence-dependent unit performance decay has been presented in this work. Travelling salesman problem and precedence-based concepts have been combined into a single continuous-time formulation to address the main issues related to this problem. The principal contribution of this work relies on the integration planning, scheduling and maintenance with sequence-dependent performance decay, flexible maintenance operations and resource limitations in a continuous-time model representation.

In order to test our model, a benchmark example from the literature has been introduced. This simple example has been solved considering scheduling and resource availability decisions without sequence-dependent issues and recovery constraints. The solution of this single time period scheduling problem has been compared with some existing solutions provided in Marchetti and Cerdá (2009), considering minimizing the overall earliness and the summation of starting times. The results demonstrate the effectiveness of the proposed formulation in terms of computational performance.

After that, three new examples (Examples 1-3) have been proposed by using the information provided in Cerdá et al. (1997). Thus, different problem instances have been solved minimizing the makespan while considering different resource availabilities and performance decay. In order to compare with other approaches, a discrete-time STN formulation (Kondili et al., 1993) and a unit-specific time event STN formulation (Janak et al., 2004) were adapted from their original versions to deal with maintenance operations and unit performance decay, by using similar ideas than the ones introduced in Biondi et al., (2017). The addition of sequence-dependent issues, as performance decay and changeover, provides a novel contribution on these existing formulations. Unfortunately, these restrictions do not provide a good performance and difficult the resolution of the STN models. In contrast, our formulation takes the advantages of TSP and precedence-based ideas, dealing efficiently with limited resources, sequence-dependent changeovers and unit performance decay, demonstrating a superior performance in comparison with other approaches.

The rest of the examples have been solved by using our proposed formulation. Thus, some cases have been proposed for the multi-period problems by considering different planning horizons and additional equations

have been introduced to deal with, inventory, backlogs and lifetime limitation in order to maximize the total profit. Results for these examples have shown the applicability of the solution approach for solving integrated planning, scheduling and maintenance problems at a reasonable computational time.

In order to analyse the behaviour of the model for planning problems, additional examples were tested by changing some of the most important model parameters. Thus, performance decay, demand profile, and parameters related to recovery operations and product's lifetime were varied from their original values in order to provide a sensitivity analysis of the model and the final solution. After solving different instances, we can conclude that the model performs very well, providing good quality solutions (<5% gap) in almost all cases in less than 1000 seconds of CPU time. The sensitivity analysis has been able to demonstrate the applicability of the solution approach and the importance of some of the main parameters involved in the resolution of the problem. For the case analyses, we have been able to demonstrate that changes on the demand profile and the inventory capacity have not affected much the final solution while changes on the other factors, especially performance decay factor, may provide a reduction in between 5% -50% on the total profit which is considerably high.

Future work will be focused on the development of efficient decomposition approaches for solving larger problem instances, considering product orders and parallel units.

Acknowledgements

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

References

- Alle, A., Papageorgiou, L. G., & Pinto, J. M. (2004a). A mathematical programming approach for cyclic production and cleaning scheduling of multistage continuous plants. *Computers & chemical engineering*, 28(1), 3-15.
- Alle, A., Pinto, J. M., & Papageorgiou, L. G. (2004b). The economic lot scheduling problem under performance decay. *Industrial & engineering chemistry research*, 43(20), 6463-6475.
- Biondi, M., Sand, G., & Harjunkoski, I. (2017). Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach. *Computers & Chemical Engineering*, 99, 325-339.
- Bock, S., Briskorn, D., & Horbach, A. (2012). Scheduling flexible maintenance activities subject to job-dependent machine deterioration. *Journal of Scheduling*, 1-14.
- Cassady, C. Richard, and Erhan Kutanoglu. "Integrating preventive maintenance planning and production scheduling for a single machine." *IEEE Transactions on reliability* 54.2 (2005): 304-309.
- Casas-Liza, J., Pinto, J. M., & Papageorgiou, L. G. (2005). Mixed integer optimization for cyclic scheduling of multiproduct plants under exponential performance decay. *Chemical Engineering Research and Design*, 83(10), 1208-1217.
- Castro, P. M., Grossmann, I. E., Veldhuizen, P., & Esplin, D. (2014). Optimal maintenance scheduling of a gas engine power plant using generalized disjunctive programming. *AIChE Journal*, 60(6), 2083-2097.
- CPLEX, I. I. (2009). V12. 1: User's Manual for CPLEX. International Business Machines Corporation, 46(53), 157.
- 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. *Industrial & Engineering Chemistry Research*, 36(5), 1695-1707.
- Dedopoulos, I. T., & Shah, N. (1995). Optimal short-term scheduling of maintenance and production for multipurpose plants. *Industrial & engineering chemistry research*, 34(1), 192-201.
- Dedopoulos, I. T., & Shah, N. (1996). Long-term maintenance policy optimization in multipurpose process plants. *Chemical engineering research & design*, 74(3), 307-320.
- Devapriya, P., Ferrell, W., & Geismar, N. (2017). Integrated production and distribution scheduling with a perishable product. *European Journal of Operational Research*, 259(3), 906-916
- Hazaras, M. J., Swartz, C. L., & Marlin, T. E. (2012). Flexible maintenance within a continuous-time state-task network framework. *Computers & Chemical Engineering*, 46, 167-177.
- Heluane, H., Blanco, A. M., Hernández, M. R., & Bandoni, J. A. (2012). Simultaneous re-design and scheduling of multiple effect evaporator systems. *Computers & Operations Research*, 39(5), 1173-1186.
- Janak, S. L., Lin, X., & Floudas, C. A. (2004). Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. *Industrial & engineering chemistry research*, 43(10), 2516-2533.

- Jain, V., & Grossmann, I. E. (1998). Cyclic scheduling of continuous parallel process units with decaying performance. *AIChE Journal*, 44(7), 1623-1636.
- Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Computers & Chemical Engineering*, 17(2), 211-227.
- Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2009). Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. *Industrial & Engineering Chemistry Research*, 49(2), 701-718.
- Liu, S., Pinto, J. M., & Papageorgiou, L. G. (2008). A TSP-based MILP model for medium-term planning of single-stage continuous multiproduct plants. *Industrial & Engineering Chemistry Research*, 47(20), 7733-7743.
- 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. *Computational Management Science*, 7(4), 407-435.
- Liu, S., Yahia, A., & Papageorgiou, L. G. (2014). Optimal production and maintenance planning of biopharmaceutical manufacturing under performance decay. *Industrial & Engineering Chemistry Research*, 53(44), 17075-17091.
- Marchetti, P. A., & Cerdá, J. (2009). A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers. *Computers & Chemical Engineering*, 33(4), 871-886.
- 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), 913-946.
- Nie, Y., Biegler, L. T., Wassick, J. M., & Villa, C. M. (2014). Extended discrete-time resource task network formulation for the reactive scheduling of a mixed batch/continuous process. *Industrial & Engineering Chemistry Research*, 53(44), 17112-17123.
- Pinto, J. M., Chen, P., & Papageorgiou, L. G. (2007). A discrete/continuous-time MILP model for medium-term planning of single stage multiproduct plants. *Computer Aided Chemical Engineering*, 24, 685-690.
- 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. *Industrial & engineering chemistry research*, 40(14), 3195-3207.
- Ruiz, R., García-Díaz, J. C., & Maroto, C. (2007). Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers & Operations Research*, 34(11), 3314-3330.
- Sanmartí, E., Espuña, A., & Puigjaner, L. (1997). Batch production and preventive maintenance scheduling under equipment failure uncertainty. *Computers & chemical engineering*, 21(10), 1157-1168.
- Schulz, E. P., Bandoni, J. A., & Diaz, M. S. (2006). Optimal shutdown policy for maintenance of cracking furnaces in ethylene plants. *Industrial & engineering chemistry research*, 45(8), 2748-2757.
- Shaik, M. A., & Floudas, C. A. (2008). Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. *Computers & Chemical Engineering*, 32(1), 260-274.
- Suryadi, H., & Papageorgiou, L. G. (2004). Optimal maintenance planning and crew allocation for multipurpose batch plants. *International journal of production research*, 42(2), 355-377.
- Varnier, C., & Zerhouni, N. (2012, May). Scheduling predictive maintenance in flow-shop. In *Prognostics and System Health Management (PHM), 2012 IEEE Conference on* (pp. 1-6). IEEE.
- Vassiliadis, C. G., Vassiliadou, M. G., Papageorgiou, L. G., & Pistikopoulos, E. N. (2000). Simultaneous maintenance considerations and production planning in multi-purpose plants. In *Reliability and Maintainability Symposium, 2000. Proceedings. Annual* (pp. 228-233). IEEE.
- Velez, S., Merchan, A. F., & Maravelias, C. T. (2015). On the solution of large-scale mixed integer programming scheduling models. *Chemical Engineering Science*, 136, 139-157.
- Vieira, M., Pinto-Varela, T., & Barbosa-Póvoa, A. P. (2017). Production and maintenance planning optimisation in biopharmaceutical processes under performance decay using a continuous-time formulation: A multi-objective approach. *Computers & Chemical Engineering*.
- Xenos, D. P., Kopanos, G. M., Ciccotti, M., & Thornhill, N. F. (2016). Operational optimization of networks of compressors considering condition-based maintenance. *Computers & Chemical Engineering*, 84, 117-131.

Appendix A

Discrete-Time Model Formulation (STN)

Given a set of discrete-time periods $H=(h1...h100)$ and operation modes $K=(k1...k10)$ and assuming fixed processing times tp_{ijk} , changeover times $\tau_{ii'}$ and recovery time tr_j with constant decay by d_{ij} , a discrete-time STN model based on the main ideas of Kondili et al. (1993) and adapted constraints from Biondi et al. (2017) for the integration of maintenance operations is proposed in Eqs.(47-60) to find the best production schedule and maintenance, taking into account task's allocation of production and recovery operations by Wh_{ijkh} and Mh_{jh} respectively, unit performance by Dh_{jh} (initial unit performance by D_j^0 and maximum unit performance by D_j^{\max}), with limited resources Rh_h and limited recovery Qh_{jh} , so as to minimise the makespan MK .

$$\sum_{j,k,h} Wh_{ijkh} = 1 \quad \forall i \quad (47)$$

$$\sum_{i,k,h \geq h' \geq h - tp_{ijk} + 1} Wh_{ijkh'} + \sum_{h \geq h' \geq h - tr_j + 1} Mh_{jh'} \leq 1 \quad \forall j, h \quad (48)$$

$$Wh_{ijkh} + Mh_{jh} + \sum_{h \geq h' \geq h - tp_{i'k'} - \tau_{ii'} + 1} Wh_{i'jk'h'} \leq 1 \quad \forall i \neq i', k, k', j, h \quad (49)$$

$$R^{\max} \geq Rh_h \geq \sum_{i,k,j,h \geq h' \geq h - tp_{ijk} + 1} Wh_{ijkh'} + \sum_{j,h \geq h' \geq h - tr_j + 1} Mh_{jh'} \quad \forall h \quad (50)$$

$$Dh_{jh} = D^0_{j|h=1} + Dh_{j,h-1|h>1} - \sum_{i,k,j,h \geq h' \geq h - tp_{ijk} + 1} Wh_{ijkh'} \cdot d_{ij} + Qh_{jh} \quad \forall j, h \quad (51)$$

$$Qh_{jh} \leq D^{\max}_j - Dh_{j,h-1} \quad \forall j, h \quad (52)$$

$$D^0_j = Dh_{j,h} \quad \forall j, h = |H| \quad (53)$$

$$Qh_{jh} \leq D^{\max}_j \cdot Mh_{jh} \quad \forall j, h \quad (54)$$

$$Qh_{jh} \geq D^{\max}_j \cdot Mh_{jh} - Dh_{j,h-1} \quad \forall j, h \quad (55)$$

$$D^0_{j|h=1} + Dh_{j,h-1} \geq \sum_{i,k} y_k \cdot Wh_{ijkh} \quad \forall j, h > 1 \quad (56)$$

$$\sum_h Mh_{jh} \leq 1 \quad \forall j \quad (57)$$

$$\sum_{j,h} Mh_{jh} \leq M^{\max} \quad (58)$$

$$MK \geq \sum_{j,k,h} Wh_{ijkh} \cdot (h - 1 + tp_{ijk}) \quad \forall i \quad (59)$$

$$MK \geq Mh_{jh} \cdot (h - 1 + tr_j) \quad \forall j, h \quad (60)$$

Continuous-Time Model Formulation (Global-time event)

Given a set of events points $N=(n1...n12)$ and operation modes $K=(k1...k10)$ and assuming continuous processing times Tn_{ijn} and recovery times Tm_{ijn} , changeover times $\tau_{ii'}$ and constant unit degradation d_{ij} , a unit-specific time event STN model is proposed here by following the main ideas of Janak et al. (2004) for scheduling tasks and adapted constraints from Biondi et al. (2017) for maintenance operations in Eqs.(61-95).

This model considers production and maintenance operations by Wn_{ijn} and Mn_{jn} , unit performance by Dn_{jn} with limited resources Rn_n and limited recovery by Qn_{jn} , so as to minimise the makespan MK . Assignment decision variables are stated by Ws_{ijn} and Wf_{ijn} in order to represent the starting event and the ending event for each task i,j at event point n . Related timing decisions for each task i,j at event point n are represented by Ts_{ijn} and Tf_{ijn} . While starting and ending times for each global event are enforced by tts_n and ttf_n variables.

$$\sum_{j,n} Ws_{ijn} = 1 \quad \forall i \quad (60)$$

$$\sum_i Wn_{ijn} \leq 1 \quad \forall j, n \quad (61)$$

$$Wn_{ijn} = \sum_{n' \leq n} Ws_{ijn'} - \sum_{n' < n} Wf_{ijn'} \quad \forall i, j, n \quad (62)$$

$$\sum_n Ws_{ijn} = \sum_n Wf_{ijn} \quad \forall i, j \quad (63)$$

$$Ws_{ijn} \leq \sum_{n' < n} Ws_{ijn'} + \sum_{n' < n} Wf_{ijn'} \quad \forall i, j, n \quad (64)$$

$$Wf_{ijn} \leq \sum_{n' \leq n} Ws_{ijn'} - \sum_{n' < n} Wf_{ijn'} \quad \forall i, j, n \quad (65)$$

$$\sum_n Tn_{ijn} \geq \sum_k \lambda_{ijk} \cdot tp_{ijk} - BM \cdot \left(1 - \sum_n Ws_{ijn}\right) \quad \forall i, j \quad (66)$$

$$\sum_n Tn_{ijn} \leq \sum_k \lambda_{ijk} \cdot tp_{ijk} + BM \cdot \left(1 - \sum_n Ws_{ijn}\right) \quad \forall i, j \quad (67)$$

$$Tf_{ijn} \geq Ts_{ijn} \quad \forall i, j, n \quad (68)$$

$$Tf_{ijn} \leq Ts_{ijn} + BM \cdot Wn_{ijn} \quad \forall i, j, n \quad (69)$$

$$Ts_{ijn} \leq Tf_{ij,n-1} + BM \cdot (1 - Wn_{ij,n-1} + Wf_{ij,n-1}) \quad \forall i, j, n > 1 \quad (70)$$

$$Ts_{ijn} \geq Tf_{ij,n-1} \quad \forall i, j, n > 1 \quad (71)$$

$$Tf_{ijn'} - Ts_{ijn} \geq \sum_{n' \leq n'' \leq n} (Tn_{ijn''} + Tm_{ijn''}) - BM \cdot (2 - Ws_{ijn} - Wf_{ijn'} + \sum_{n \leq n'' < n'} Wf_{ijn''}) \quad \forall i, j, n \leq n' \quad (72)$$

$$Tf_{ijn'} - Ts_{ijn} \leq \sum_{n' \leq n'' \leq n} (Tn_{ijn''} + Tm_{ijn''}) + BM \cdot (2 - Ws_{ijn} - Wf_{ijn'} + \sum_{n \leq n'' < n'} Wf_{ijn''}) \quad \forall i, j, n \leq n' \quad (73)$$

$$Ts_{i'jn} \geq Tf_{ijn-1} + \tau_{i'} - BM \cdot (2 - Ws_{i'jn} - Wf_{ijn-1} + \sum_{i' \neq i'', n \leq n'' < n'} Wf_{i''jn''}) \quad \forall i \neq i', j, n > 1 \quad (74)$$

$$Rn_n + \sum_{i,j} Wn_{ijn} = R^{\max} \quad \forall n = 1 \quad (75)$$

$$Rn_n = Rn_{n-1} + \sum_{i,j} Wn_{ijn-1} - \sum_{i,j} Wn_{ijn} \quad \forall n > 1 \quad (76)$$

$$Rn_n \leq R^{\max} \quad \forall n \quad (77)$$

$$tfn_n \geq tsn_n \quad \forall n \quad (78)$$

$$tsn_n = tfn_{n-1} \quad \forall n \quad (79)$$

$$Tf_{ij,n-1} \geq tsn_n - BM \cdot (1 - Wn_{ij,n-1} + Wf_{ij,n-1}^f) \quad \forall i, j, n > 1 \quad (80)$$

$$Tf_{ij,n-1} \leq tsn_n + BM \cdot (1 - Wn_{ij,n-1}) \quad \forall i, j, n > 1 \quad (81)$$

$$tsn_n \geq Ts_{ijn} - BM \cdot (1 - Wn_{ijn}) \quad \forall i, j, n \quad (82)$$

$$tsn_n \leq Ts_{ijn} + BM \cdot (1 - Wn_{ijn}) \quad \forall i, j, n \quad (83)$$

$$Dn_{jn} = D^0_{jn=1} + Dn_{j,n-1|n>1} - \sum_i (Tn_{ijn} \cdot d_{ij}) + Qn_{jn} \quad \forall j, n \quad (84)$$

$$Qn_{jn} \leq D^{\max}_j - Dn_{j,n-1} \quad \forall j, n \quad (85)$$

$$Qn_{jn} \leq D^{\max}_j \cdot Mn_{jn} \quad \forall j, n \quad (86)$$

$$D^0_j = Dn_{j,n} \quad \forall j, n = |N| \quad (87)$$

$$\sum_n Mn_{jn} \leq 1 \quad \forall j \quad (88)$$

$$\sum_{j,n} Mn_{jn} \leq M^{\max} \quad (89)$$

$$Mn_{jn} \leq \sum_i Wf_{ijn} \quad \forall j, n \quad (90)$$

$$Tm_{ijn} \geq tr_j^{\max} \cdot Qn_{jn} - BM \cdot (1 - Wf_{ijn}) \quad \forall i, j, n \quad (91)$$

$$Tm_{ijn} \leq tr_j^{\max} \cdot Qn_{jn} + BM \cdot (1 - Wf_{ijn}) \quad \forall i, j, n \quad (92)$$

$$D^0_{jn=1} + Dn_{j,n-1} \geq \sum_k y_k \cdot \lambda_{ijk} - BM \cdot (1 - Ws_{ijn}) \quad \forall i, j, n \quad (93)$$

$$\sum_k \lambda_{ijk} = 1 \quad \forall i, j \quad (94)$$

$$MK \geq Tf_{ijn} \quad \forall i, j, n = |N| \quad (95)$$