Available online at www.sciencedirect.com
ScienceDirect

Computers and Chemical Engineering 32 (2008) 2664–2683

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines

Muge Erdirik-Dogan, Ignacio E. Grossmann*

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States

Received 27 April 2007; received in revised form 24 July 2007; accepted 24 July 2007
Available online 6 August 2007

Abstract

In this paper we present a multi-period mixed integer linear programming model for the simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel units. While effective for short time horizons, the proposed scheduling model becomes computationally expensive to solve for long time horizons. In order to address this problem, we propose a bi-level decomposition algorithm in which the original problem is decomposed into an upper level planning and a lower level scheduling problem. For the representation of the upper level, we propose an MILP model which is based on a relaxation of the original model, but accounts for the effects of scheduling by incorporating sequencing constraints, which results in very tight upper bounds. In the lower level the simultaneous planning and scheduling model is solved for a subset of products predicted by the upper level. These sub-problems are solved iteratively until the upper and lower bounds converge. A number of examples are presented that show that the planning model can often obtain the optimal schedule in one single iteration.

Keywords: Planning; Scheduling; Multi-product continuous plants; MILP; Slot based formulation

1. Introduction

Despite the importance of the short-term scheduling of multi-product facilities involving continuous processes, relatively few articles have been published in this area. Sahinidis and Grossmann (1991) considered the problem of cyclic scheduling of multi-product plants for single-stage plants with parallel units. This work was extended by Pinto and Grossmann (1994) who considered the case of multi-stage continuous plants with one unit per stage and with intermediate storage. Alle, Papageorgiou, and Pinto (2004) extended this work to include cleaning considerations. Jain and Grossmann (1998) addressed the scheduling of multiple feeds on parallel continuous units where process performance models are also incorporated. Mendez and Cerda. (2002) studied resource constrained multi-product plants involving parallel mixers operating in a continuous mode. Even fewer papers have appeared to address the short term scheduling of refinery operations. Examples include the works by Lee, Pinto, Grossmann, and Park (1996) and by Shah (1996) who addressed the problem of crude-oil unloading with inventory control using MILP models. Another example is the work of Jia and Ierapetritou (2004) who addressed this problem with an MINLP model.

The integration of planning and scheduling has received increasing attention in recent years. This is due to chemical process industry's interest in improving the overall competitiveness in the global market place by reducing costs and inventories while meeting due dates. While there has been progress towards integrating planning and scheduling, performing simultaneously these tasks still remains elusive. This is due to the fact that simultaneous planning and scheduling involves in principle solving the scheduling problem for the entire planning horizon. This, however, results in a very large scale optimization problem since the problem is defined over long time horizons.

In order to address these computational difficulties, strategies based on aggregation, decomposition and heuristics have been considered in literature. A good example for the aggregation approach is the work by Wilkinson, Shah, and Pantelides (1996) who used a constraint aggregation approach to obtain approximate solutions to the large scale production and distribution planning problems for multi-site production sites that are represented with Resource Task Network (Pantelides, 1994).

* Corresponding author. Tel.: +1 412 268 3642; fax: +1 412 268 7139.
E-mail address: grossmann@cmu.edu (I.E. Grossmann).

0098-1354/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2007.07.010

Nomenclature

Indices

i, k	products
l	slots
$\bar{l}_m$	last slot of unit m
m	units
t	time periods
$\bar{t}$	last time period

Sets

I	set of products
I_m	set of products that can be processed on unit m
L	set of slots
L_m	set of slots that belong to unit m
M	set of units
M_i	set of units that can process product i

Parameters

$\mathrm{CINV}_{i,t}$	inventory cost of product i in period t
$\mathrm{COP}_{i,t}$	operating cost of product i in period t
$\mathrm{CP}_{i,t}$	selling price of product i in period t
$\mathrm{CTRANS}_{i,k,m}$	transition cost of changing the production from product i to k in unit m
$d_{i,t}$	demand of product i at the end of period t
H_t	duration of the t th time period
HT_t	time in terms of hours at the end of the t th time period
INVI_i	initial inventory of product i
$\mathrm{MRT}_{i,m}$	minimum run lengths
N_m	number of slots postulated for unit m
$\mathrm{Pen}_{i,t}$	penalty cost for unsatisfied demands
$r_{i,m}$	production rate of product i in unit m
$\tau_{i,k,m}$	transition time from product i to product k in unit m

Variables

$\mathrm{Area}_{i,t}$	over estimation of the area below the inventory time graph for product i at the end of period t
$\mathrm{INV}_{i,t}$	inventory level of product i at the end of time period t
$\mathrm{INVO}_{i,t}$	inventory level of product i at the end of period t after the demands are satisfied
$\mathrm{NY}_{i,m,t}$	total number of slots that are allocated for product i in unit m during period t
$S_{i,t}$	sales of product i at the end of period t
$\mathrm{Te}_{m,l,t}$	end time of slot l of unit m during period t
$\mathrm{TRNP}_{m,t}$	total transition time for unit m within each time period
$\mathrm{TRT}_{i,k,m,t}$	to denote if product i is followed by product k at the end of period t
$\mathrm{Ts}_{m,l,t}$	start time of slot l of unit m during time period t
$W_{i,m,l,t}$	0–1 variable to denote the assignment of product i to slot l of unit m during period t
$X_{i,m,l,t}$	amount of product i produced in slot l of unit m during period t
$\tilde{X}_{i,m,t}$	amount of product i produced in unit m during period t
$\mathrm{XF}_{i,m,t}$	0–1 variable to denote if product i is the first product in unit m during period t
$\mathrm{XL}_{i,m,t}$	0–1 variable to denote if product i is the last product in unit m during period t
$\mathrm{YOP}_{i,m,t}$	0–1 variable to denote if product i is assigned unit m during period t
$\mathrm{YP}_{i,m,t}$	0–1 variable to denote the assignment of product i to unit m during period t
$Z_{i,k,m,l,t}$	to denote if product i is followed by product k in slot l of unit m during period t
$\mathrm{ZP}_{i,k,m,t}$	0–1 variable to denote if product i precedes product k in unit m during time period t
$\mathrm{ZZP}_{i,k,m,t}$	0–1 variable to denote if the link between products i and k are broken
$\mathrm{ZZZ}_{i,k,m,t}$	transition variable denoting the changeovers across adjacent periods
$\Delta_{i,t}$	slack variable
$\tilde{\theta}_{i,m,t}$	production time of product i in unit m during period t
$\Theta_{i,m,l,t}$	production time of product i in slot l of unit m during period t

Another example is the work by Birewar and Grossmann (1990) for batch plants with multiple stages and zero-wait policy, where the batches that belong to the same products are aggregated and sequencing considerations for scheduling are accounted at the planning level.

Decomposition techniques on the other hand are generally based on two-level decomposition schemes, where an aggregate planning problem is solved in the upper level to define production targets and a detailed scheduling problem is solved in the lower level with fixed binary variables as determined by the upper level so as to meet these targets. The major challenge lies in developing an aggregate planning model that not only yields tight upper bounds so as to reduce the number of total iterations, but also predicting the production as accurately as possible in order to reduce infeasibilities and mismatches that may occur between the upper and lower levels. Bassett, Pekny, and Reklaitis (1996) follow such a decomposition scheme for batch processes where an aggregate planning model is solved and separate detailed scheduling problems are subsequently solved for each planning period. Papageorgiou and Pantelides (1996) also follow a similar scheme where in the upper level an aggregate planning model that is based on State Task Network representation (Kondili, Pantelides, & Sargent, 1993) is solved and in the lower level the detailed scheduling model is solved for fixed values of binary variables such as active campaigns and active tasks as predicted by the upper level. Erdirik-Dogan and Grossmann (2006) propose a bi-level decomposition scheme for single unit multi-product continuous plants where the higher level consists of an aggregate model where sequence dependent changeovers are underestimated, while the lower level corresponds to a slot based MILP scheduling model. However, instead of solving the lower level by fixing all product assignments for each period as determined by the upper level, only the assignments that were not selected by the upper level are excluded leaving the other assignments as free decision variables. Furthermore, the computational efficiency of the algorithm is improved by introducing superset, subset and capacity cuts which make it possible to eliminate many solutions from the upper level model. Another method for dealing with different time scales is to use a rolling horizon approach where only a subset of the planning periods include the detailed scheduling decisions. Dimitriadis et al. (1997) presented RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. A recent approach proposed by Sung and Maravelias (2007) relies on the idea of finding a projection of the STN scheduling model into a lower dimensional space using computational techniques for finding the convex hull. While this approach appears to be promising, it has some difficulties when dealing with sequence-dependent changeovers.

In this work, we consider the extension of the work by Erdirik-Dogan and Grossmann (2006) for the simultaneous planning and scheduling of single unit continuous plants to the case of single stage with parallel units. A slot-based MILP scheduling model is proposed that readily accounts for sequence dependent transition times, transition costs and inventory costs. Since the proposed model becomes intractable for large problem instances, we propose a bi-level decomposition procedure that allows rigorous integration of planning and scheduling. For representing the upper level, we propose an MILP model that is based on a relaxation of the original detailed scheduling model and has a unique feature the incorporation of sequencing constraints based on the traveling salesman problem that provide very accurate upper bounds.

The paper is organized as follows. In Section 2, we present the problem statement. This is followed by the formulation of the MILP model proposed for the simultaneous planning and scheduling. In Section 4, we describe the bi-level decomposition algorithm and the mathematical models for the upper and lower levels, respectively. This is followed by the examples which demonstrate the performance of the proposed approach compared to the full space model.

2. Problem definition

Given are a number of products that are to be processed in a plant involving a single processing stage and several continuous production units operating in parallel (see Fig. 1). The products each unit can process as well as the corresponding processing rates and production costs are specified. Given are also sequence-dependent changeover times and costs, which arise when production on one unit is changed from one product to another. We assume that the planning horizon under consideration is composed of a certain number of periods given by due dates in which demands that vary over time are specified as upper bounds (see Fig. 2).

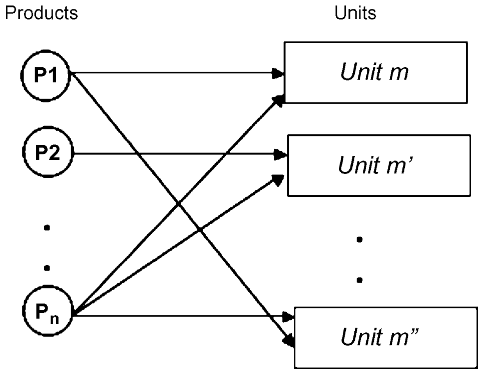

Fig. 1. Single-stage continuous plant with parallel units.

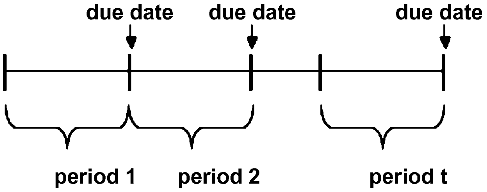

Fig. 2. Time periods defined by due dates.

The problem is then to determine:

1. The products to be produced in each unit and in each time period.
2. The sequence and detailed timing of production as well as the length. of production times in each unit and in each time period.
3. The amounts to be produced.
4. The change in inventory levels with respect to time.

The objective is to maximize the profit in terms of sales revenues, operating costs, inventory costs and transition costs.

3. Mathematical formulation

This section describes the proposed slot-based MILP model for the simultaneous planning and scheduling of multi-product single-stage continuous plants with multiple parallel units. The proposed model, which is an extension of the single unit model proposed by Erdirik-Dogan and Grossmann (2006), is based on continuous time representation. The model parameters, such as demands, due dates, prices, and cost coefficients are assumed to be deterministic.

We postulate N_m asynchronous time slots for each unit m for each time period t where N_m is the total number of products that can be processed by unit m (see Fig. 3). Assignments of products to the slots is to be determined by the model to define the sequence of production. Binary variable $W_{i,m,l,t}$ is used to model the potential assignment of product i to slot l of unit m during time period t. The length of each time slot is a variable to be determined by the model, and is equal to the summation of the assigned product's processing time and the corresponding transition time (see Fig. 4). Sequence dependent transitions are

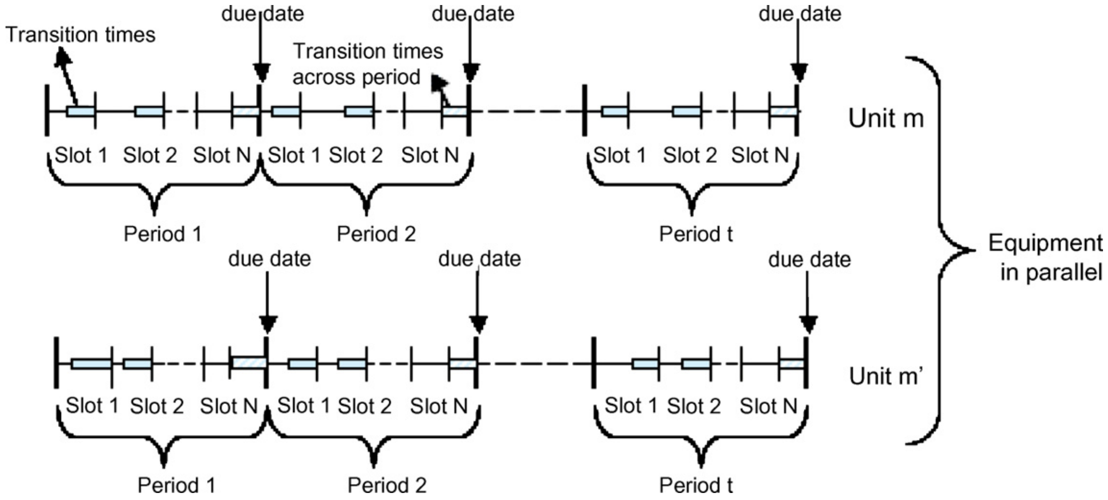

Fig. 3. Time slots postulated for each unit and each time period.

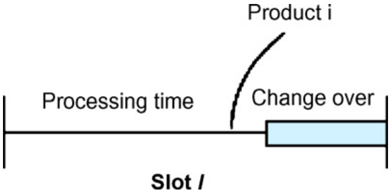

Fig. 4. Length of time slots.

activated depending on the assignments of products to slots. The indices, sets, parameters, and variables defined in the model are described in the Nomenclature section given at the end of the paper.

The MILP model (P) for the planning and scheduling problem is described in the next section.

3.1. Assignments and processing times

$$\sum_{i \in I(m)} W_{i,m,l,t} = 1 \quad \forall\, m, l \in L_m, t \tag{1}$$

$$\Theta_{i,m,l,t} \leq H_t * W_{i,m,l,t} \quad \forall\, i \in I_m, m, l \in , L_m, t \tag{2}$$

$$\Theta_{i,m,l,t} \geq \mathrm{MRT}_{i,m} * W_{i,m,l,t} \quad \forall\, i \in I_m, m, l \in , L_m, t \tag{3}$$

$$X_{i,m,l,t} = r_{i,m} * \Theta_{i,m,l,t} \quad \forall\, i \in , I_m, m, l \in , L_m, t \tag{4}$$

Eq. (1) represents the condition that exactly one product must be assigned to each slot of each unit at each time period. We enforce the utilization of each postulated slot to circumvent the difficulties that may arise when modeling the sequence-dependent changeovers across adjacent time periods. Note that the same product can be produced in more than one slot and more than one unit at each time period (see Erdirik-Dogan and Grossmann, 2006).

According to constraint (2) the processing time of product *i* in slot *l* of unit *m* during time period *t* is set to zero if product *i* is not assigned to slot *l* of unit *m* during time period *t*. Constraint (2) also defines an upper bound, the length of each time period H_t, on the processing time. Constraint (3) ensures that the processing time of an assigned product exceeds its minimum run length.

In constraint (4) we define the amount of production $X_{i,m,l,t}$, which is proportional to the processing times and production rates. Note that the production rate is a constant that is both product and unit dependent.

3.2. Transitions

Transitions arise when production on one unit is changed from one product to another. These transitions may be associated to a change in the operating conditions or to the cleaning of units.

3.2.1. Transitions within each time period

In order to take into account sequence-dependent transitions within each time period, we introduce the transition variable $Z_{i,k,m,l,t}$:

$$Z_{i,k,m,l,t}\begin{cases} 1 & \text{if product}i\text{ assigned to slot}l\text{ of unit}m\text{ is followed} \\ & \text{by product}k\text{ assigned to slot}l+1\text{ of unit}m \\ & \text{ during time}t, \\ 0 & \text{otherwise} \end{cases}$$

$$W_{i,m,l,t} \wedge W_{k,m,l+1,t} \Leftrightarrow Z_{i,k,m,l,t} \quad \forall i, \;\; k\epsilon I(m), \quad i \neq k, \;\; \forall l\epsilon L_m - \{\bar{l}_m\}, \;\; \forall m, \;\; \forall t \tag{5}$$

The proposition in (5) links these transitions variables ($Z_{i,k,m,l,t}$) with the assignment variables ($W_{i,m,l,t}$). That is, $Z_{i,k,m,l,t}$ should become 1 if and only if product *i* is assigned to slot *l* of unit *m* and product *k* is assigned to the consecutive slot ($l+1$) of unit *m* during the same time period. The above logical condition can be represented by the following inequalities:

$$Z_{i,k,m,l,t} \geq W_{i,m,l,t} + W_{k,m,l+1,t} - 1 \quad \forall i, \;\; k\epsilon I(m), \quad i \neq k, \;\; \forall l\epsilon L_m - \{\bar{l}_m\}, \;\; \forall m, \;\; \forall t \tag{6}$$

$$W_{i,m,l,t} \geq Z_{i,k,m,l,t} \quad \forall i, k\,\epsilon I(m), \quad i \neq k, \;\; \forall m, \;\; \forall l \in , L_m \;\; \forall t \tag{7}$$

$$W_{k,m,l+1,t} \geq Z_{i,k,m,l,t} \quad \forall i, k \epsilon I(m), \quad i \neq k, \quad \forall m, \quad \forall \epsilon L_m - \{\bar{l}_m\}, \quad \forall t \tag{8}$$

where $\bar{l}_m$ is the last slot of unit m for each time period.

Inequalities (7) and (8) can be easily shown to be redundant since the transition variable $Z_{i,k,m,l,t}$ is a cost item in the objective function and will tend to be zero. Furthermore, using a similar reasoning, the transition variables $Z_{i,k,m,l,t}$ need not be declared as binaries but may be relaxed in the interval [0, 1].

Another way of enforcing the same condition in (5) is to use the following set of propositions:

$$W_{i,m,l,t} \Leftrightarrow \bigvee_{k \in I_m} Z_{i,k,m,l,t} \quad \forall i \epsilon I(m), m, l \in , L_m, t \tag{9}$$

$$W_{k,m,l+1,t} \Leftrightarrow \bigvee_{i \in I_m} Z_{i,k,m,l,t} \quad \forall k \epsilon I(m), \quad m, \ l \epsilon L_m - \{\bar{l}_m\}, \quad t \tag{10}$$

According to proposition (9), there is exactly one transition from product i in unit m at time period t if and only if product i is assigned to unit m during time period t. Similarly, according to proposition (10), there is exactly one transition to product k in unit m during time period t if and only if product k is assigned to unit m of time period t.

Mathematically, these propositions can be written as follows (see also Sahinidis and Grossmann, 1991):

$$\sum_{k \in I(m)} Z_{i,k,m,l,t} = W_{i,m,l,t} \quad \forall\, i \in I(m), m, l \in L_m, t \tag{11}$$

$$\sum_{i \in I(m)} Z_{i,k,m,l,t} = W_{k,m,l+1,t} \quad \forall\, k \in I(m), m, \quad l \in L_m - \{\bar{l}_m\}, t \tag{12}$$

Note that the formulation given by constraints (11) and (12) is not only tighter than the formulation given by (6), but also requires fewer number of constraints if the cardinality of the product set is less than 4 (see Appendix A). The following property can be established for constraints (11) and (12):

Property 1. Constraints (11) and (12) are equivalent to constraints (6), (7) and (8) under constraint (1), and they exhibit a relaxation that is at least as tight as (6)–(8) (see Appendix B for proof).

3.2.2. *Transitions across adjacent periods*

Since all the postulated slots are forced to be occupied by the model through constraint (1), we only need to consider the transitions that occur between the last slot of time period, t, and the first slot of the consecutive time period $(t + 1)$ for each unit (see Fig. 5).

In order to account for sequence-dependent transitions across adjacent periods, we introduce the variable $\mathrm{TRT}_{i,k,m,t}$ which is defined by constraint (13):

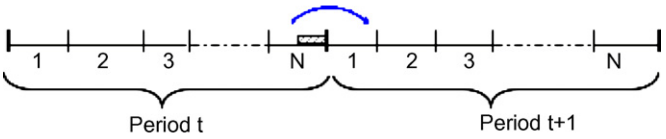

Fig. 5. Transition across adjacent periods.

$$\mathrm{TRT}_{i,k,m,t} \geq W_{i,m,l,t} + W_{k,m,l',t+1} - 1 \quad \forall i, \quad k \epsilon I(m), \quad m, \quad l = \{\bar{l}_m\}, \quad l' = 1, \quad t \epsilon T - \{\bar{t}\} \tag{13}$$

In (13), $\mathrm{TRT}_{i,k,m,t}$ will become 1 if both $W_{i,m,l,t}$ and $W_{k,m,l',t+1}$ are 1. If at least one of them is *zero*, then (13) becomes redundant.

Following a similar reasoning as in constraints (9) and (10), another way of enforcing the same condition is to use the following set of constraints:

$$\sum_{k \in I_m} \mathrm{TRT}_{i,k,m,t} = W_{i,m,l,t} \quad \forall i \epsilon I(m), \quad m, \quad l = 1, \quad t \tag{14}$$

$$\sum_{i \in I_m} \mathrm{TRT}_{i,k,m,t} = W_{k,m,l',t+1} \qquad k \epsilon I(m), \quad m, \quad l' = \bar{l}_m, \quad t \epsilon T - \{\bar{t}\} \tag{15}$$

The transition variables $\mathrm{TRT}_{i,k,m,t}$ once again need not be declared as binaries if they are treated as continuous variables with an upper bound of 1.

3.3. *Timing relations*

$$\mathrm{Te}_{m,l,t} = \mathrm{Ts}_{m,l,t} + \sum_{i \in I(m)} \theta_{i,m,l,t} + \sum_{i \in I(m)} \sum_{k \in I(m)} \tau_{i,k} * Z_{i,k,m,l,t} \quad \forall m, l \in L_m, t \tag{16}$$

$$\mathrm{Ts}_{m,l,t+1} \geq \mathrm{Te}_{m,l',t} + \sum_{i \in I(m)} \sum_{k \in I(m)} \tau_{i,k} * \mathrm{TRT}_{i,k,m,t} \quad \forall m, \quad l = \bar{l}_m, l' = 1, t \in T - \{\bar{t}\} \tag{17}$$

$$\mathrm{Te}_{m,l,t} = \mathrm{Ts}_{m,l+1,t} \quad \forall m, l \in L_m - \{\bar{l}_m\}, t \tag{18}$$

$$\mathrm{Te}_{m,l,t} \leq \mathrm{HT}_t \quad \forall l = \bar{l}_m, t \tag{19}$$

In (16), the end time slot l of unit m during time t is equal to the start time of slot l plus the processing time of the product assigned to slot l plus the corresponding transition times. Note that according to (1) and (2), exactly one processing time in the $\sum_{i \epsilon I(m)} \theta_{i,m,l,t}$ term is nonzero.

Constraint (17) accounts for the transitions across adjacent time periods. The start time of the first slot of unit m of time period t must be greater than the end time of the last slot of the previous time period $(t-1)$ summed with the corresponding transition time between the time periods. According to (18), the end time of each slot must be equal to the start time of the consecutive slot. Constraint (19) ensures that the end time of the last slot of period t is less than or equal to the end time of time period t.

3.4. Inventory balances and costs

$$\mathrm{INV}_{i,t} = \mathrm{INVI}_i + \sum_{m \in M_i} r_{i,m} * \sum_{l \in L_m} \theta_{i,m,l,t} \quad \forall i,\ t=1 \tag{20}$$

$$\mathrm{INV}_{i,t} = \mathrm{INVO}_{i,t-1} + \sum_{m \in M_i} r_{i,m} * \sum_{l \in L_m} \theta_{i,m,l,t} \quad \forall i,\ t \neq 1 \tag{21}$$

$$\mathrm{INVO}_{i,t} = \mathrm{INV}_{i,t} - S_{i,t} \quad \forall i, t \tag{22}$$

$$\mathrm{Area}_{i,t} \geq \mathrm{INVO}_{i,t-1} * H_t + \left(\sum_{m \epsilon M_i} r_{i,m} * \sum_{l \in L_m} \theta_{i,m,l,t} \right) * H_t \quad \forall i, t \tag{23}$$

Eq. (20) defines the inventory level of product i at the end of the first time period as the summation of the initial inventory for product i and the amount of product i produced in the first time period. In equation (21), the inventory level of product i at the end of time t is equal to the sum of the final inventory level of product i at the end of time period $t-1$, and the amount of product i produced during time period t. In constraint (22) the final inventory level of product i at the end of time period t is defined as the inventory level after the demands are satisfied at the end of each time period. As for the inventory cost, we use constraint(23), which is a linear overestimation as discussed in Erdirik-Dogan and Grossmann (2006).

3.5. Demand

$$S_{i,t} \geq d_{i,t} \quad \forall i,\ t \tag{24}$$

Constraint (24) states that the demands must be satisfied or exceeded for all products in the plant at the due dates, which correspond to the end time of each period. Note that demands to be satisfied are defined as lower bounds. If the demands are too high this might lead to infeasible solutions. To guard against such a case, one can add a slack variable, $\Delta_{i,t}$, to constraint (24) to yield constraint (25) and subtract the term $\sum_i \sum_t \mathrm{Pen}_{i,t} \times \Delta_{i,t}$ from the objective function (26) where $\mathrm{Pen}_{i,t}$ is a penalty cost. In this way the model would yield a solution that minimizes violations of demand constraints. The application of these constraints and derivation of a schedule with tardy delivery of orders are given in Example 5.

$$S_{i,t} + \Delta_{i,t} \geq d_{i,t} \quad \forall i,\ t \tag{25}$$

3.6. Objective function

$$\begin{aligned} Z^P = & \sum_i \sum_t \mathrm{CP}_{i,t} * S_{i,t} - \sum_i \sum_t \mathrm{CINV}_{i,t} * \mathrm{Area}_{i,t} \\ & - \sum_t \sum_m \sum_{i \epsilon I(m)} \mathrm{COP}_{i,t} * \tilde{X}_{i,t} \\ & - \sum_i \sum_k \sum_m \sum_t \sum_l \left[\mathrm{TRANS}_{i,k} * Z_{i,k,m,l,t} \right. \\ & \left. - \mathrm{CTRANS}_{i,k} * \mathrm{TRT}_{i,k,m,t} \right] \end{aligned} \tag{26}$$

The objective is to maximize the profit (26), which is given by the sum of sales revenues, the inventory costs, operating costs, the transition costs within each week and transition costs across adjacent weeks.

We should note that for the case when the same product is assigned to more than one slot of the same unit in the same time period, the consecutive utilization of the slots is ensured by optimality. Since the inventory costs have been overestimated, assigning the same product to non-consecutive slots will yield the same inventory cost as assigning the same product to consecutive slots. Furthermore, since assignment of the same product to non-consecutive slots will result in an increase in the transition costs, we do not need to enforce the consecutive utilization of slots by the same product. As an example, see Fig. 6, where

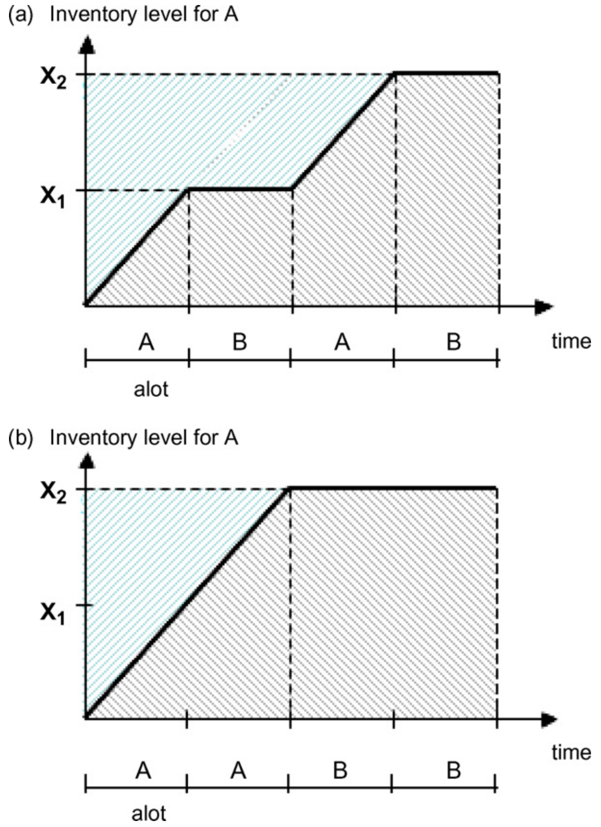

Fig. 6. Non-consecutive vs. consecutive utilization of slots.

both solutions result in the same total inventory costs shown by the shaded area. However, the solution shown in Fig. 6 a results in higher transition costs as it has two more transitions than the solution shown in Fig. 6 b.

We should also note that allowing the assignment of the same product to more than one slot of the same unit in the same time period, combined with the fact that slots have variable lengths, increases the degeneracy of the problem (Fig. 7). This is due to the fact that allocation of the total processing times of the assigned products to slots in any feasible way within a time period, while preserving the original sequence, will result in several alternative optima. In order to avoid this degeneracy, we introduce symmetry breaking constraints as proposed by Erdirik-Dogan and Grossmann (2006). Note that degeneracy in a time period occurs if the total number of available slots (N_m) is greater than the total number of products assigned to that time period ($\sum_i \mathrm{YOP}_{i,t}$). In other words degeneracy occurs if the number of flexible slots is greater than zero ($\mathrm{FS} = N - \sum_i \mathrm{YOP}_{i,t}$). The idea of the symmetry breaking constraints is to enforce the product that is assigned to the first slot to be allocated to FS number of additional slots, hence leaving just one available slot for the rest of the assigned products. For the example shown in Fig. 7, introducing symmetry breaking constraints reduces the feasible solutions only to the solution presented in Fig. 7(a).

As shown by Erdirik-Dogan and Grossmann (2006), the symmetry-breaking constraints can be written as,

$$\mathrm{YOP}_{i,m,t} \geq W_{i,m,l,t} \quad \forall i \epsilon I(m), \ \forall m, \ \forall l \in L_m, \ \forall t \tag{27}$$

$$\mathrm{YOP}_{i,m,t} \leq \mathrm{NY}_{i,m,t} \leq N_m * \mathrm{YOP}_{i,m,t} \quad \forall i \epsilon I(m), \ \forall m, \ \forall t \tag{28}$$

$$\mathrm{NY}_{i,m,t} \geq N_m - \left[\left(\sum_{i \epsilon I(m)} \mathrm{YOP}_{i,m,t}\right) - 1\right] - M * (1 - W_{i,m,l,t}) \quad \forall i \epsilon I(m), \ m, \ t \tag{29}$$

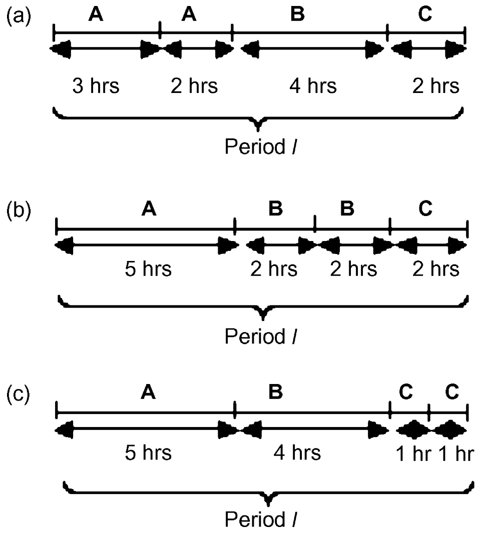

Fig. 7. Symmetric solutions.

$$\mathrm{NY}_{i,m,t} \leq N_m - \left[\left(\sum_{k \epsilon I(m)} \mathrm{YOP}_{k,m,t}\right) - 1\right] - M * (1 - W_{i,m,l,t}) \quad \forall i \epsilon I(m), \ m, \ t \tag{30}$$

where M is a parameter that is chosen with a sufficiently large value to render (29) and (30) redundant when $W_{i,m,l,t} = 0$.

The proposed MILP scheduling model (OP) described by constraints (1)–(4), (11), (12), (14)–(24), (26)–(30) can be solved directly with branch and bound enumeration procedures. For large problems and long planning horizons, however, the computational expense of solving the detailed scheduling model (OP) can be high. Therefore, there is a need for an alternative solution strategy that enables the solution of large-scale problems at reasonable computational expense. We propose a bi-level decomposition algorithm that is different in nature from the algorithm presented in Erdirik-Dogan and Grossmann (2006) in the sense that it involves a very accurate upper bound through the introduction of sequencing constraints. The proposed algorithm is described in the next section.

4. Solution strategy

In order to decrease the computational effort of solving the problem, the original detailed scheduling model is decomposed into an upper level planning and a lower level scheduling problem.

The upper level model (P) is based on a relaxation of the original model (OP) and is mainly concerned with the assignments of products to available equipment during each time period and the sequencing of the products, rather than the detailed timing of the production. Since the upper level model is a relaxation of the original problem, the solution of (P) at iteration r, (P^r), yields a valid upper bound on the objective of problem (OP). In the lower level, we solve the original problem (OP), but by excluding the products that were not selected by the upper level problem for each unit at each time period (i.e. setting to zero a subset of the variables $W_{i,m,l,t}$). The lower level problem (OP^r) is then a sub-problem of (OP) at iteration r that yields a lower bound on the profit provided that it corresponds to a feasible solution. Sub-problems (P^r) and (OP^r) are solved iteratively (see Fig. 8). Integer cuts are used to obtain new solutions at each iteration and also to exclude supersets of previously obtained feasible configurations. The solution of (OP^r) with the current lower bound becomes the final solution after convergence is achieved. We should note that due to the accuracy of the solutions obtained by the upper level, the algorithm often converges in a single iteration.

4.1. Upper level problem

The proposed planning MILP model (P) for the upper level is a simplified representation of the detailed scheduling model (OP). Specifically, the slot time representation that has been adopted for the detailed scheduling model, and hence the index l, is not considered. Therefore, the proposed model is a lower

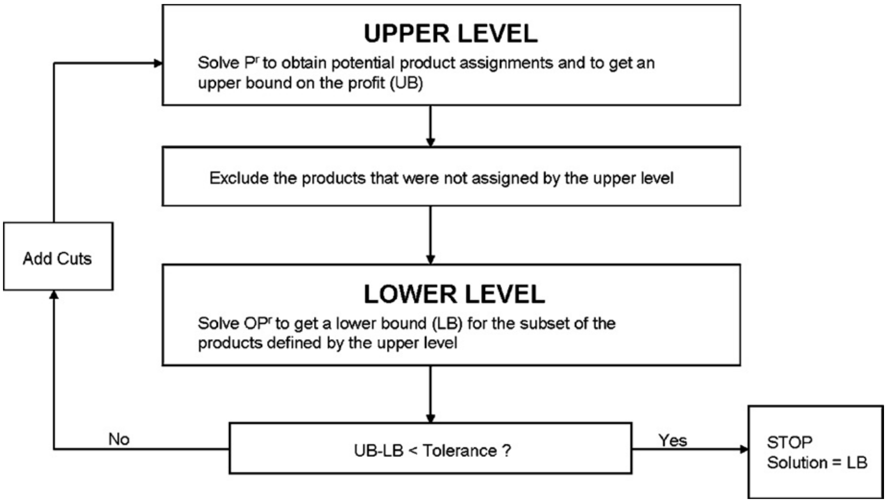
UPPER LEVEL
Solve P^r to obtain potential product assignments and to get an upper bound on the profit (UB)
Exclude the products that were not assigned by the upper level
LOWER LEVEL
Solve OP^r to get a lower bound (LB) for the subset of the products defined by the upper level
UB-LB < Tolerance ?
No
Yes
STOP
Solution = LB
Add Cuts

Fig. 8. Flowchart for the proposed algorithm.

dimensional model compared to the detailed scheduling model. However, in order to produce tight upper bounds sequencing constraints will be added as shown below. Assignment and production variables and constraints of the detailed scheduling model are aggregated over the time slots. Finally, the detailed timing variables and constraints are replaced by sequencing variables and sequencing constraints that are related to the traveling salesman problem (Nemhauser and Wolsey, 1998). The aggregated MILP model will be used to provide a rigorous upper bound on the profit of the original problem, and also to predict the products that will be excluded from the lower level.

The MILP model (P) formulated for representing the upper level is as follows.

4.1.1. Assignment and production constraints

$$\tilde{\theta}_{i,m,t} \leq H_t * \mathrm{YP}_{i,m,t} \quad \forall i \epsilon I(m), m, t \tag{31}$$

$$\tilde{X}_{i,m,t} = r_{i,m} * \tilde{\theta}_{i,m,t} \quad \forall i \epsilon I(m), m, t \tag{32}$$

Since the sequencing of the products is no longer defined through slots, we introduce the binary variable, $\mathrm{YP}_{i,m,t}$, which represents the assignment of product i to unit m during time period t. We also introduce aggregate variables, $\tilde{\theta}_{i,m,t} = \sum_l \theta_{i,m,l,t}$, and $\tilde{X}_{i,m,t} = \sum_l X_{i,m,l,t}$. Constraint (31) is obtained by summing constraint (2) over l and replacing $\theta_{i,m,l,t}$ by $\tilde{\theta}_{i,m,t}$ and $\sum_l W_{i,m,l,t}$ by $\mathrm{YP}_{i,m,t}$ (see Erdirik-Dogan and Grossmann, 2006). Constraint (32) is obtained by summing constraint (4) over l and replacing $X_{i,m,l,t}$ by $\tilde{X}_{i,m,t}$ and $\theta_{i,m,l,t}$ by $\tilde{\theta}_{i,m,t}$.

4.1.2. Inventory balances and costs

$$\mathrm{INV}_{i,t} = \mathrm{INVI}_{i,t-1} + \sum_{m \epsilon M_i} r_{i,m} * \tilde{\theta}_{i,m,t} \quad \forall i, t \tag{33}$$

$$\mathrm{INV}_{i,t} = \mathrm{INVO}_{i,t-1} + \sum_{m \epsilon M_i} r_{i,m} * \tilde{\theta}_{i,m,t} \quad \forall i, t \tag{34}$$

$$\mathrm{INVO}_{i,t} = \mathrm{INV}_{i,t} - S_{i,t} \quad \forall i, t \tag{35}$$

$$\mathrm{Area}_{i,t} \geq (\mathrm{INVO}_{i,t-1}) * H_t + \left(\sum_{m \epsilon M_i} r_{i,m} * \tilde{\theta}_{i,m,t} \right) * H_t \quad \forall i, t \tag{36}$$

Constraints (33), (34), (35) and (36) are in essence the same as (20), (21), (22) and (23) except that $\sum_l \theta_{i,m,l,t}$ is replaced by the aggregate variable $\tilde{\theta}_{i,m,t}$.

4.1.3. Demand

$$S_{i,t} \geq D_{i,t} \quad \forall i, t \tag{37}$$

Constraint (37) is the same as (24).

4.1.4. Sequencing constraints

As stated before, the proposed upper level model is based on ignoring the detailed timing constraints and is only concerned with determining the sequence of production on each unit during each time period. Hence, we ignore the detailed timing constraints (6)–(19) and determine the optimal sequence via sequencing constraints which have been proposed by Erdirik-Dogan and Grossmann (2007) for the case of parallel batch reactors. The extension of these constraints to the case of continuous plants is direct. We include a brief explanation of these constraints, although the equations are similar as in Erdirik-Dogan and Grossmann (2007).

The basic idea is first to generate a cyclic schedule for each unit within each time period, where the transition times amongst the assigned products are minimized. Then, to determine the optimal sequence for each unit in each time period one of the links in the cycle is broken (see example in Fig. 9). Since the transition times are assumed to be directly proportional to the transition costs, the link to be broken will correspond to the pair with the highest transition time, thus resulting in the minimum transition time sequence in each time period.

The key decisions we are concerned with are: (i) precedence of production in each cycle, (ii) location of the link to be broken for each cycle to determine the optimal sequence.

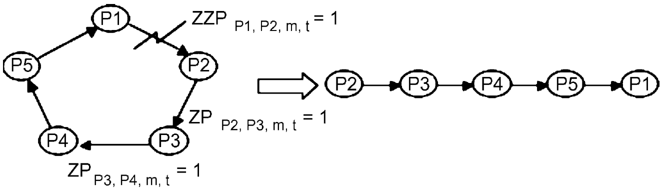

Fig. 9. Cyclic schedule and optimal sequence.

These decisions are represented by the following binary variables:

- $\mathrm{ZP}_{i,k,m,t}$: becomes 1 if product i precedes product k in unit m, time period t.
- $\mathrm{ZZP}_{i,k,m,t}$: becomes 1 if the link connecting products i and k is broken in unit m, during time period t.

The following constraints are proposed to generate a cyclic schedule within each time period:

$$\mathrm{YP}_{i,m,t} = \sum_{k \in I_m} \mathrm{ZP}_{i,k,m,t} \quad \forall i \epsilon I(m), \ \forall m, \ \forall t \tag{38}$$

$$\mathrm{YP}_{k,m,t} = \sum_{i \in I_m} \mathrm{ZP}_{i,k,m,t} \quad \forall k \epsilon I(m), \ \forall m, \ \forall t \tag{39}$$

In Eq. (38), there is exactly one transition from product i in unit m at time period t if and only if product i to any product k is assigned to unit m during period t. Similarly, according to proposition (39), there is exactly one transition from any product i to product k if and only if product k is assigned to unit m at period t. Note that these constraints can be interpreted as the assignment constraints in the traveling salesman problem (Miller and Pekny, 1991):

$$\sum_{i \in I_m} \sum_{k \in I_m} \mathrm{ZZP}_{i,k,m,t} = 1 \quad \forall m, \ \forall t \tag{40}$$

$$\mathrm{ZZP}_{i,k,m,t} \le \mathrm{ZP}_{i,k,m,t} \quad \forall i \epsilon I(m), \ \forall k \epsilon I(m), \ \forall m, \ \forall t \tag{41}$$

Eq. (40) states that exactly one of the links in the optimal cycle must be broken. Also, according to inequality (41), a link cannot be broken if the corresponding pair is not selected in the cycle.

Constraints (38)–(41) will lead to a sequence provided that the cyclic schedule obtained by (38) and (39) has no sub-cycles, which in fact should be commonly the case for sequence-dependent changeovers (i.e. asymmetric traveling salesman problem). In the case when sub-cycles are present, these constraints will lead to subsequences as seen in Fig. 10. While these will not correspond to a complete feasible sequence, they do correspond to a valid lower bounding representation for the changeover times, which in turn leads to a valid upper bound to the profit.

We should note that since in constraints (38) and (39), products i and k may correspond to the same product, we impose the following logic proposition (see Erdirik-Dogan and Grossmann, 2007):

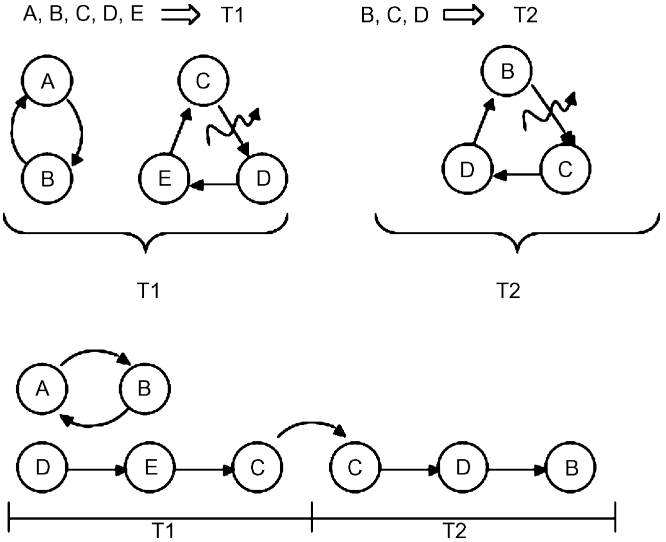

Fig. 10. Subsequences in presence of sub-cycles.

$$\mathrm{YP}_{i,m,t} \bigwedge \left[\bigwedge_{k \neq i} \neg \mathrm{YP}_{k,m,t} \right] \Leftrightarrow \mathrm{ZP}_{i,i,m,t} \quad \forall i \in I_m, m, t \tag{42}$$

Logic proposition (42) states that if product i is assigned in unit m at time period t, and none of the products k different than i are assigned in the same unit at the same time period, then product i can be preceded by product i. Also, if product i is preceded by product i in unit m at period t, then only product i is assigned in unit m for period t, and none of the products k other than i are assigned to the same unit at the same time period.

The expression in (42) can be written mathematically as follows:

$$\mathrm{YP}_{im,t} \ge \mathrm{ZP}_{i,i,m,t} \quad \forall i \epsilon I(m), \ \forall m, \ \forall t \tag{43}$$

$$\mathrm{ZP}_{i,i,m,t} + \mathrm{YP}_{k,m,t} \le 1 \quad \forall i \epsilon I(m), \ \forall k \epsilon I(m), \quad i \neq k, \ \forall m, \ \forall t \tag{44}$$

$$\mathrm{ZP}_{i,i,m,t} \ge \mathrm{YP}_{i,m,t} - \sum_{k \neq i, k \epsilon I(m)} \mathrm{YP}_{k,m,t} \quad \forall i \epsilon I(m), \ \forall m, \ \forall t \tag{45}$$

Constraint (46) defines the total transition time within each period, $\mathrm{TRNP}_{m,t}$, as the summation of the transition times corresponding to each existing pair ($\mathrm{ZP}_{i,k,m,t}$) minus the transition time corresponding to the link that is broken from the sequence ($\mathrm{ZZP}_{i,k,m,t}$) (see example in Fig. 11):

$$\mathrm{TRNP}_{m,t} = \sum_{i \in I_m} \sum_{k \in I_m} \tau_{i,k} * \mathrm{ZP}_{i,k,m,t} - \sum_{i \in I_m} \sum_{k \in I_m} \tau_{i,k} * \mathrm{ZZP}_{i,k,m,t} \quad \forall m, t \tag{46}$$

Transition times across adjacent periods can be handled easily by taking into account the transition times between the last

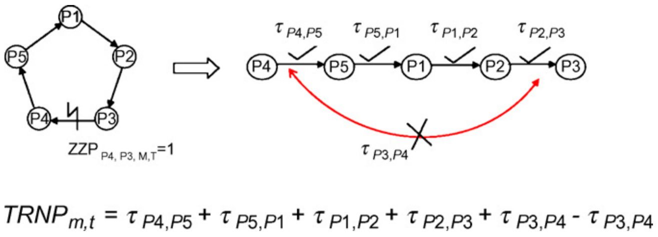
P1
P2
P3
P4
P5
ZZP P4, P3, M, T=1
τ P4,P5
τ P5,P1
τ P1,P2
τ P2,P3
τ P3,P4
TRNPm,t = τ P4,P5 + τ P5,P1 + τ P1,P2 + τ P2,P3 + τ P3,P4 - τ P3,P4

Fig. 11. Total transition time within each week.

product of period t and the first product of the subsequent period $(t+1)$. These products correspond to the pair where the cycle is broken to form the sequence. According to their relative position in the cycle, the head of the cycle will correspond to the first product and the tail will correspond to the last product.

In order to account for the transition times across adjacent periods, we introduce the following binary variables:

- $\mathrm{XF}_{i,m,t}$: becomes 1 if product i is in the first position of the sequence for unit m during time period t.
- $\mathrm{XL}_{i,m,t}$: becomes 1 if product i is in the last position of the sequence for unit m during time period t.
- $\mathrm{ZZZ}_{i,k,m,t}$: becomes 1 if if product i at time t is followed by product k at time $t+1$ in unit m.

$$\mathrm{XF}_{k,m,t} \geq \sum_{i\epsilon I(m)} \mathrm{ZZP}_{i,k,m,t} \quad \forall k\epsilon I(m),\ \forall m,\ \forall t \qquad (47)$$

$$\mathrm{XL}_{i,m,t} \geq \sum_{k\epsilon I(m)} \mathrm{ZZP}_{i,k,m,t} \quad \forall i\epsilon I(m),\ \forall m,\ \forall t \qquad (48)$$

Constraint (47) states that if at least one of the links that point from any product i to product k is broken, then product i becomes the first product in the optimal sequence obtained for unit m during time period t. Similarly, according to constraint (48) if at least one of the links pointing from product i to any product k is broken, then product i becomes the last product in the optimal sequence obtained for unit m during time period t:

$$\sum_{i} \mathrm{XF}_{i,m,t} = 1 \quad \forall i\epsilon I(m),\ \forall m,\ \forall t \qquad (49)$$

$$\sum_{i} \mathrm{XL}_{i,m,t} = 1 \quad \forall i\epsilon I(m),\ \forall m,\ \forall t \qquad (50)$$

Constraints (49) and (50) ensure that exactly one product must be the first to be processed, and exactly one product must be the last to be processed:

$$\sum_{k\epsilon I(m)} \mathrm{ZZZ}_{i,k,m,t} = \mathrm{XL}_{i,m,t} \quad \forall i\epsilon I(m),\ \forall m,\ \forall t \qquad (51)$$

$$\sum_{i\epsilon I(m)} \mathrm{ZZZ}_{i,k,m,t} = \mathrm{XF}_{k,m,t+1} \quad \forall k\epsilon I(m),\ \forall m,\ \forall t\epsilon T - \{\bar{t}\} \qquad (52)$$

Constraints (51) and (52) define the transition variable that accounts for transitions across adjacent weeks, $\mathrm{ZZZ}_{i,k,m,t}$. In (51) , exactly one transition occurs from product i at the end of time period t in unit m, if and only if i is the last produced last at time period t. Similarly, according to (52), exactly one transition to product i' occurs at the beginning of time period $t+1$ in unit m if and only if product i' is produced the first at time period $t+1$ in unit m.

4.1.5. Time balance

$$\sum_{i\epsilon I(m)} \tilde{\theta}_{i,m,t} + \mathrm{TRNP}_{m,t} + \left[\sum_{i\epsilon I(m)} \sum_{k\epsilon I(m)} \tau_{i,k} * \mathrm{ZZZ}_{i,k,m,t}\right] \leq H_t \quad \forall t \qquad (53)$$

In constraint (53), summation of the processing times of the products assigned to unit m during time period t, plus the total transition time within that time period, plus the transition time to the adjacent time period, cannot exceed the length of that time period. Note that these transitions are generally asymmetric.

4.1.6. Objective function

$$\begin{aligned}\text{Profit} = {} & \sum_{i}\sum_{t} \mathrm{CP}_{i,t} * S_{i,t} - \sum_{i}\sum_{t} \mathrm{CINV}_{i,t} * \mathrm{Area}_{i,t} \\ & - \sum_{t}\sum_{i} \mathrm{COP}_{i,t} * \tilde{X}_{i,t} - \sum_{t}\sum_{m}\sum_{i\epsilon I(m)}\sum_{k\epsilon I(m)} \mathrm{CTRANS}_{i,k} \\ & * \left(\mathrm{ZP}_{i,k,m,t}\right) - \sum_{t}\sum_{m}\sum_{i\epsilon I(m)}\sum_{k\epsilon I(m)} \mathrm{CTRANS}_{i,k} \\ & * \left(\mathrm{ZZZ}_{i,k,m,t} - \mathrm{ZZP}_{i,k,mt}\right)\end{aligned} \qquad (54)$$

The objective is to maximize profit in terms of sales minus inventory costs, operating costs and transition costs. The last three terms of the objective function stand for the transition costs: third to last accounts for the transition costs within each cycle, the second to last stands for the transition cost of the link that was broken, and the last term stands for the transition costs of the changeovers that occur across adjacent weeks.

The upper level planning problem (P) is then given by constraints (31)–(54).

4.2. Lower-level problem

In the lower level, the original detailed scheduling model (OP) is solved for a subset of products as determined by the upper level. Specifically, products that are predicted to be produced by the upper level, may or may not be produced by the lower level. However, those products which were predicted as not to be produced by the upper level, are excluded from the lower level in each unit during each time period by setting the corresponding binary variables to zero. Solving the original detailed model (OP) in a reduced space makes it possible to greatly expedite the solution of the corresponding MILP model:

$$\mathrm{YOP}_{i,m,t} \le \mathrm{YP}^{r}_{i,m,t} \quad \forall i \in I_m, m, t \tag{55}$$

The above condition is enforced by the inequality in (55). The lower level problem (OP^r) is then described by constraints (1)–(4), (11), (12), (14)–(30) and (55).

4.3. Integer cuts

After each iteration, if the bounds obtained from the upper level model and the lower level model do not lie within a specified tolerance, we must obtain a new solution from the upper level model. We incorporate into the upper level model integer cuts to generate new solutions in terms of the assignment variable $\mathrm{YP}_{i,m,t}$. The simplest and weakest cut that excludes the previously obtained feasible solutions from the upper level model is as follows (Balas and Jeroslow, 1972):

$$\sum_{(i,t)\epsilon Z_1^r} \mathrm{YP}_{i,m,t} - \sum_{(i,t)\epsilon Z_0^r} \mathrm{YP}_{i,m,t} \le \left|Z_1^r\right| - 1 \tag{56}$$

where $Z_0^r = \left\{i, t \left| \mathrm{YP}^r_{i,m,t} = 0\right.\right\}$ and $Z_1^r = \left\{i, t \left| \mathrm{YP}^r_{i,m,t} = 1\right.\right\}$

Note that Z_0^r and Z_1^r are obtained from the optimal solution at the upper level in terms of the assignment variable in iteration r.

Instead of using the integer cut in (56), we can use the cover cut which is stronger in the sense that it does not only exclude the current assignment r but also any other assignment that is a superset of assignment r. The general form of the cover cuts is as follows:

$$\sum_{(i,t)\epsilon Z_1^r} \mathrm{YP}_{i,m,t} \le \left|Z_1^r\right| - 1 \tag{57}$$

The justification of excluding supersets is due to the fact that any superset implies adding an additional product which as a consequence results in an increase in transition times and costs thereby decrease in the profit. We should also note that the extension of superset and subset cuts presented in Erdirik-Dogan and Grossmann (2006) to the parallel units case is direct. However, we did not include them in this paper since the algorithm often converges in one single iteration as shown in the examples section.

5. Remarks

1. The proposed decomposition scheme is rigorous and guarantees the global optimal solution in a finite number of iterations.
2. For the case when there are no sub-cycles, the solution obtained by the upper level planning model (P) is identical to the one obtained by the original detailed model (OP) since the upper level model (P) rigorously accounts for the sequence-dependent changeover times and costs. Hence, in this case the algorithm converges in one single iteration.
3. When the upper level model (P) exhibits sub-cycles, a valid upper bound on the profit is obtained since constraints (38)–(41) represent a relaxation to the traveling salesman problem.
4. For the case of a single unit, the detailed model proposed for representing the lower level reduces to the detailed scheduling model proposed in Erdirik-Dogan and Grossmann (2006) when the first form of transition constraints (6)–(8) are employed.

6. Examples

The effectiveness of the proposed approach will be illustrated with several examples presented in this section. It should be noted that all the models presented in this paper have been implemented in GAMS 22.3 and solved with CPLEX 10.1 on an 2X Intel Xeon 5150 at 2.66 GHz machine. Note that all the examples presented in this section are solved for a 0% optimality tolerance. The data for Example 3 is given in Section 6.3 and the data for the other examples are omitted but is available upon request to the authors.

6.1. Example 1

As we have mentioned in Section 5, for the case of a single unit the lower level model reduces to the detailed scheduling model proposed in Erdirik-Dogan and Grossmann (2006). To demonstrate this and also to compare the performance of the upper level formulations, we consider Example 1a of Erdirik-Dogan and Grossmann (2006) that consists of five products A–E, and a single reactor over a 4-week horizon.

Fig. 12 shows the Gantt chart that is obtained with the proposed method that has a profit of $43,120.8. This is the same solution as the one obtained by Erdirik-Dogan and Grossmann (2006). Table 1 shows the comparison of the solution times and the corresponding problem sizes and solutions for the two methods. From the results it is clear that the improved upper level formulation shows significant improvements in the solution times. While the method of Erdirik-Dogan and Grossmann (2006) takes 208 s and eight major iterations to solve, the solution with the proposed method takes only 18 s and converges in one single iteration.

The reduction in solution times and total number of iterations arises from the ability of the proposed upper level model to explicitly account for the effects of scheduling making it possible to produce very accurate upper bounds. Since the upper level solution is free of subcycles it corresponds to the global optimum

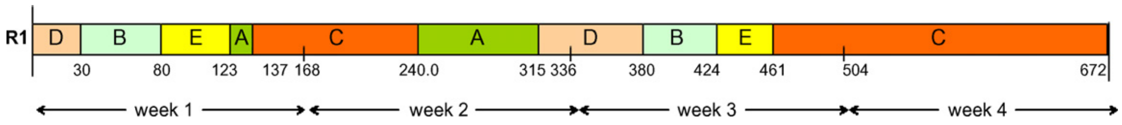

Fig. 12. Gantt chart for Example 1.

Table 1
Comparison of the current formulation with that of Erdirik-Dogan and Grossmann (2006)

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
Proposed algorithm				18.33	43120.80
Problem UB	260	482	453	8.14	43120.80
Problem LB	120	1047	802	10.19	43120.80
Erdirik-Dogan and Grossmann (2006)				207.9	43120.80
Problem UB	20	151	564	2	43013.00
Problem LB	120[a]	996	949	205.9	43120.80

[a] Not the actual size; numbers are prior to further processing and variable fixing.

Table 2
Problem sizes and results for Example 2 for 4 weeks

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
Full space	128	811	737	8.1	105,021
Proposed algorithm				0.85	105,021
Problem UB	216	421	475	0.34	105,021
Problem LB	128[a]	811	737	0.51	105,021

[a] Not the actual size; numbers are prior to further processing and variable fixing.

solution. Also, note that the computationally difficult part of the problem has now moved from the lower level to the upper level. As opposed to Erdirik-Dogan and Grossmann (2006), the solution of the upper level becomes the bottleneck in the bi-level decomposition algorithm.

6.2. Example 2

In this example we consider a manufacturing facility that consists of five products A–E and two reactors R1 and R2. Reactor R1 can process products A–C, whereas reactor R2 can process C–E.

Table 2 shows a comparison of the problem sizes and the corresponding solutions of the proposed algorithm with the full space solution of the scheduling model for a time horizon of 4 weeks. The same objective is obtained by both methods. While it took the full-space method 8 CPUs to find this solution, the proposed method took less than 1 CPUs and converged in one single iteration. Note that, while the size of the upper level model is smaller than the size of the lower level model in terms of number of constraints and number of continuous variables, it is larger in terms of number of binary variables. This is due to the fact that while in the upper level model the detailed timing of operations with continuous variables are neglected, the effects of sequence-dependent changeovers are taken into account by the introduction of additional binary variables.

Fig. 13 shows the optimal sequence obtained by the upper level for each week, while 14 shows the Gantt chart obtained by the lower level. As can be seen from Fig. 13, no subcycles were obtained in the upper level and hence the two levels produced identical solutions making it possible to converge in one single iteration.

Table 3 compares the proposed method with the full-space method where the original scheduling model (OP) is solved directly for horizons varying from 8 to 24 weeks. While for short horizons (i.e. 4 weeks, see Table 2 and 8 weeks) the full-space method yielded the same global optimum solution as the proposed method within a reasonable time, for longer horizons (i.e. 16 and 24 weeks) the full-space method failed to terminate in the specified time limits. Specifically, at 16 weeks 400 CPUs were required to solve the problem with the proposed approach, whereas the full-space method failed to terminate in 6000 CPUs yielding a feasible solution. For the case of 24 weeks, it takes the proposed method about 3 h to obtain the global optimum solution whereas the full-space method can only provide a feasible solution after 4 h. We should also note that in all the cases (8–2–4 weeks) presented in Table 3 no subcycles were obtained in the upper level solution, and hence the proposed method converged in one single

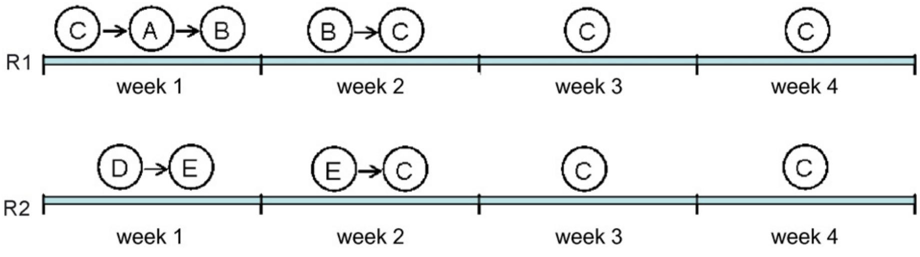

Fig. 13. Optimal schedule obtained by the upper level for Example 2 for 4 weeks.

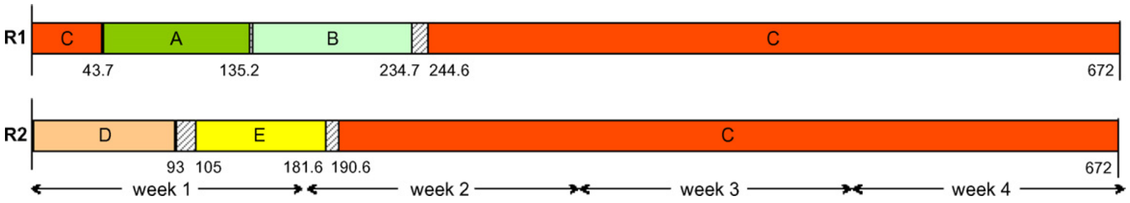

Fig. 14. Gantt chart obtained by the lower level for Example 2 for 4 weeks.

Table 3
Model and solution statistics for Example 2 for 8–24 weeks

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
8 weeks					
Full space	224	1259	1177	13.85	187,518
Proposed algorithm				1.26	187,518
Problem UB	328	686	761	1.09	187,518
Problem LB	224[a]	1259	1177	0.17	187,518
16 weeks					
Full space	448	2515	2361	6000[b]	374,392
Proposed algorithm				398.3	375,933
Problem UB	656	1374	1609	398.3	375,933
Problem LB	448[a]	2515	2361	0.3	375,933
24 weeks					
Full space	672	3771	3545	15,000[b]	570,400
Proposed algorithm				10,610.3	571,525
Problem UB	984	2062	2417	10,609	571,525
Problem LB	672[a]	3771	3545	1.3	571,525

[a] Not the actual size; numbers are prior to further processing and variable fixing.
[b] Search terminated; the best feasible solution is posted.

iteration since the upper and lower level solutions were identical.

6.3. Example 3

In this example we consider eight products A–H and three reactors R1, R2 and R3. Each reactor can process only a subset of the products, namely R1 can process products A–D, R2 can process C–F and finally R3 can process E–H. The data for this example is presented in Table 4 .

Table 5 shows the problem sizes and the corresponding solutions for both the full-space scheduling model and the proposed approach for planning horizons varying from 4 weeks to 24 weeks. Except for the 4 weeks horizon instance, the full-space method failed to terminate in the specified time limits (10,000 CPUs for 8–16 weeks and 50,000 CPUs for 24 weeks).

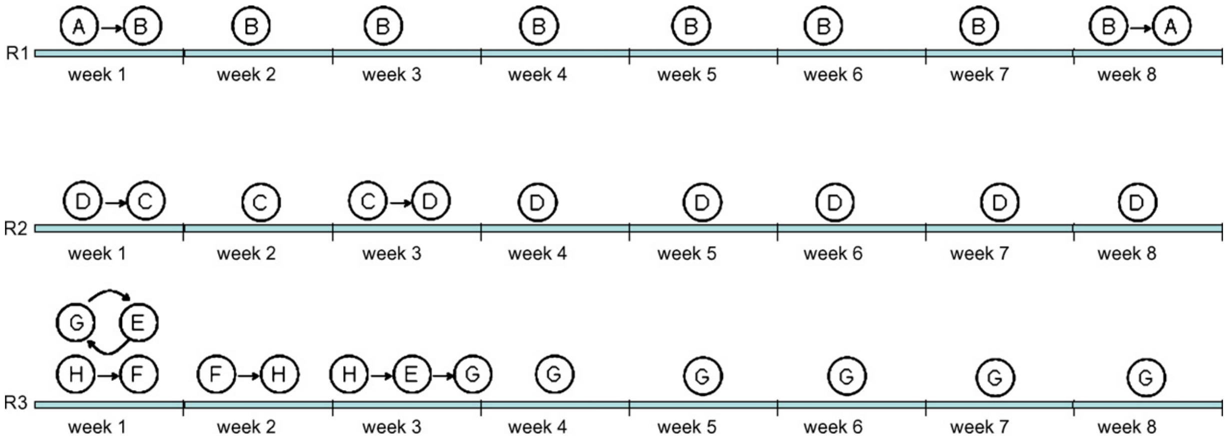

Fig. 15. Optimal schedule obtained by the upper level for Example 3 for 8 weeks.

Table 4
Data for Example 3

	R1			
	A	B	C	D
Transition times (h)				
R1				
A	0	22	20	25
B	10	0	5	6
C	4	12	0	15
D	8	8	12	0

	R2			
	C	D	E	F
Transition times (h)				
R2				
C	0	24	28	30
D	6	0	12	12
E	4	8	0	8
F	10	12	8	0

	R3			
	E	F	G	H
Transition times (h)				
R3				
E	0	4	6	8
F	10	0	15	15
G	25	28	0	30
H	8	6	12	0

	R1			
	A	B	C	D
Transition costs ($)				
R1				
A	0	220	200	250
B	100	0	50	60
C	40	120	0	150
D	80	80	120	0

	R2			
	C	D	E	F
Transition costs ($)				
R2				
C	0	240	280	300
D	60	0	120	120
E	40	80	0	80
F	100	120	80	0

	R3			
	E	F	G	H
Transition costs ($)				
R3				
E	0	40	60	80
F	100	0	150	150
G	250	280	0	300
H	80	60	120	0

Table 4 (*Continued*)

	T1	T2	T3	T4	T5	T6	T7	T8
Lower bounds for demands (kg)								
A	10000	20000	30000	10000	10000	10000	30000	20000
B	25000	20000	15000	25000	15000	15000	15000	15000
C	30000	40000	50000	30000	30000	30000	10000	10000
D	30000	20000	13000	30000	20000	20000	10000	10000
E	30000	20000	12000	30000	20000	20000	20000	10000
F	10000	10000	30000	30000	10000	10000	10000	10000
G	20000	20000	10000	10000	20000	20000	20000	20000
H	10000	10000	10000	10000	20000	20000	10000	10000

	R1		R2		R3
Production rates (kg/h)					
A	1000	C	800	E	1000
B	2000	D	1500	F	1000
C	800	E	1000	G	1200
D	1500	F	1000	H	1200
Selling price ($/kg)					
A	0.95	C	0.90	E	0.85
B	0.99	D	1.00	F	1.00
C	0.90	E	0.85	G	0.99
D	1.00	F	1.00	H	0.95
Operating cost ($/kg)					
A	0.35	C	0.36	E	0.30
B	0.34	D	0.37	F	0.32
C	0.36	E	0.30	G	0.35
D	0.37	F	0.32	H	0.34
Inventory cost ($/(kg h))					
0.0000306					

Therefore, these values correspond to feasible solutions that are suboptimal. For the case of up to 16 weeks, the proposed approach yields up to one or more orders of magnitude reduction in the solution times compared to the full-space method. For the case of 24 weeks the proposed method also requires a large amount of time but is able to find the global optimum after about 4 h. In all the cases, except for the 8 weeks horizon instance, no subcycles were obtained at the upper level and hence the upper level and the lower level produced identical solutions.

Fig. 15 shows the optimal schedule obtained by the upper level for the 8-week planning horizon instance where the solution exhibited one subcycle (see R3 in week 1). Since the solution consists of a subcycle, the solution obtained by the upper level does not correspond to a feasible schedule. However, it does yield a valid upper bound on the profit. Since the difference between the upper bound (3,595,626) and the lower bound (3,594,083) is less than the specified tolerance (0.04%), the algorithm converges in one single iteration (see Table 5).

Fig. 16 shows the actual schedule obtained by the lower level. As can be seen, the only inconsistency between the schedules of the upper level and the lower level is in the first week of R3. Table 6 compares the objective function items of the upper level with that of the lower level. The discrepancy between the objective function values is due to the fact that by yielding a solution with a subcycle (see week 1 of R3 in Fig. 15) instead of the sequence 'G–E–H–F', the upper level led to an underestimation

Table 5
Model and solution statistics for Example 3 for 4–24 weeks

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
4 Weeks					
Full space	336	2,543	1608	485	1,738,291
Proposed algorithm				15.6	1,738,291
Problem UB	528	947	951	15	1,738,291
Problem LB	336[a]	2,543	1608	0.6	1,738,291
8 Weeks					
Full space	672	5,083	3252	10,000[b]	3,590,795
Proposed algorithm				69	3,594,083
Problem UB	1056	1,903	1923	67	3,595,626
Problem LB	672[a]	5,083	3252	2	3,594,083
16 Weeks					
Full space	1433	10,163	6540	10,000[b]	7,259,075
Proposed algorithm				439.0	7,282,340
Problem UB	2112	3,815	3867	438	7,282,340
Problem LB	1433[a]	10,163	6540	0.95	7,282,340
24 Weeks					
Full space	2016	15,243	9828	50,000[b]	10,933,522
Proposed algorithm				14,022.9	10,951,000
Problem UB	3168	5,727	5811	14,021	10,951,000
Problem LB	2016[a]	15,243	9828	1.97	10,951,000

[a] Not the actual size; numbers are prior to further processing and variable fixing.
[b] Search terminated; the best feasible solution is posted.

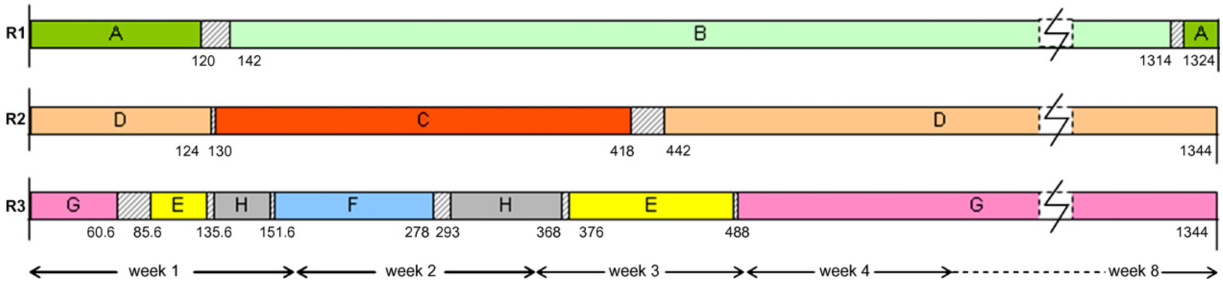

Fig. 16. Gantt chart obtained by the lower level for Example 3 for 8 weeks.

Table 6
Comparison of upper and lower levels in terms of objective function items for Example 3

$	Upper level	Lower level
Sales	5,637,258	5,634,882
Operating costs	2,002,787	2,001,947
Inventory costs	37,563	37,551
Transition costs	1280	1300

of the total transition time by 2 h making it possible to produce an extra 2400 kg of product G.

6.4. Example 4

This example is much larger in size compared to the previous examples. We consider a facility that consists of 10 products (A–J) and 5 reactors (R1–R5). R1 can process products A–D and F; R2 can process products B, H, and I; R3 can process A, D, G, and J; R4 can process B, E, and I; finally R5 can process C, E, H, and J. Note that since the number of products that can be processed by each unit is different, the total number of slots postulated for each unit differs from each other. Specifically, five slots are postulated for each time period of R1, three slots for R2, four slots for R4, and finally four slots for R5.

Table 7 shows the comparison of the proposed method with the full-space scheduling model for planning horizons varying from 4 weeks to 24 weeks. We can see that the trends in the computational results obtained in the previous examples hold for this large problem. The full-space method failed to terminate within the specified time limit for all the instances of this example, and hence, the reported solutions correspond to feasible solutions that are suboptimal. As can be seen from Table 7, solution times are significantly reduced with the proposed approach although computational time increases dramatically at 24 weeks of plan-

Table 7
Results for Example 4 for 4–24 weeks of planning horizons

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
4 weeks					
Full space	624	4,075	2,886	5000[a]	3,550,009
Proposed algorithm				17.7	3,555,060
Problem UB	828	1,444	1,545	15.5	3,555,060
Problem LB	624[b]	4,075	2,886	2.2	3,555,060
8 Weeks					
Full space	1248	8,147	5,830	5000[a]	7,235,455
Proposed algorithm				128.7	7,266,038
Problem UB	1656	2,904	3,125	127.3	7,266,038
Problem LB	1248[b]	8,147	5,830	1.4	7,266,038
16 Weeks					
Full space	2496	16,291	11,718	10,000[a]	14,443,198
Proposed algorithm				4612.6	14,685,603
Problem UB	3312	5,824	6,285	4611	14,685,603
Problem LB	2496[b]	16,291	11,718	1.6	14,685,603
24 weeks					
Full space	3744	24,435	17,606	40,000[a]	21,897,639
Proposed algorithm				40,003.6[a]	22,073,007
Problem UB	4968	8,744	9,445	40,000[a]	22,073,007
Problem LB	3744[b]	24,435	17,606	3.6	22,073,007

[a] Search terminated; the best feasible solution is posted.

[b] Not the actual size; numbers are prior to further processing and variable fixing.

ning horizon. In fact, the proposed approach failed to find the optimal solution within the specified time limit (40,000 s). We should note that, although the reported sizes of the full space model and the lower level model are the same, solving the lower level model results in about three orders of magnitude reduction in the solution times. This is due to the fact that many of the binary variables (assignment and transition) are fixed in the lower level depending on the results as obtained by the upper level. In other words, the lower level model is solved in a reduced space of binary variables. Note also that once again the solution of the upper level became the bottleneck.The effects of tolerance on the upper level is shown in Table 8.

Table 8
Effects of tolerance on the upper level for Example 4 (24 weeks)

Tolerance (%)	Time (CPUs)	Solution ($)
0.03	20,089	22,072,573
0.05	5,108	22,064,573
1	753	21,998,169
2	253	21,829,271

Table 9
Original demand values for Example 2

	T1	T2	T3	T4
Lower bounds for demands (kg)				
A	10,000	20,000	30,000	10,000
B	25,000	20,000	15,000	25,000
C	30,000	40,000	50,000	30,000
D	30,000	20,000	13,000	30,000
E	30,000	20,000	12,000	30,000

No subcycles were obtained in any of the instances, and therefore the solutions obtained by the upper level were identical to those obtained by the lower level as shown in Fig. 17 and 18. Hence, the convergence was achieved in a single iteration for all the instances of this example.

6.5. *Example 5*

The purpose of this example is to address the infeasibility issues that might arise for the case of high demands as discussed in Section 3.5. In order to illustrate such a case, we have resolved the problem presented in Example 2 for the case of 4 weeks by substantially increasing the demands from the values shown in Table 9 to the ones shown in Table 10. Solving this problem using the original model (OP) described by constraints (1)–(4), (11), (12), (14)–(24), (26)–(30) leads to an infeasible solution since the specified demands cannot be sat-

Table 10
Increased demand values for Example 5

	T1	T2	T3	T4
Lower bounds for demands (kg)				
A	50,000	90,000	60,000	50,000
B	65,000	80,000	80,000	80,000
C	70,000	90,000	80,000	100,000
D	40,000	90,000	60,000	40,000
E	60,000	70,000	60,000	60,000

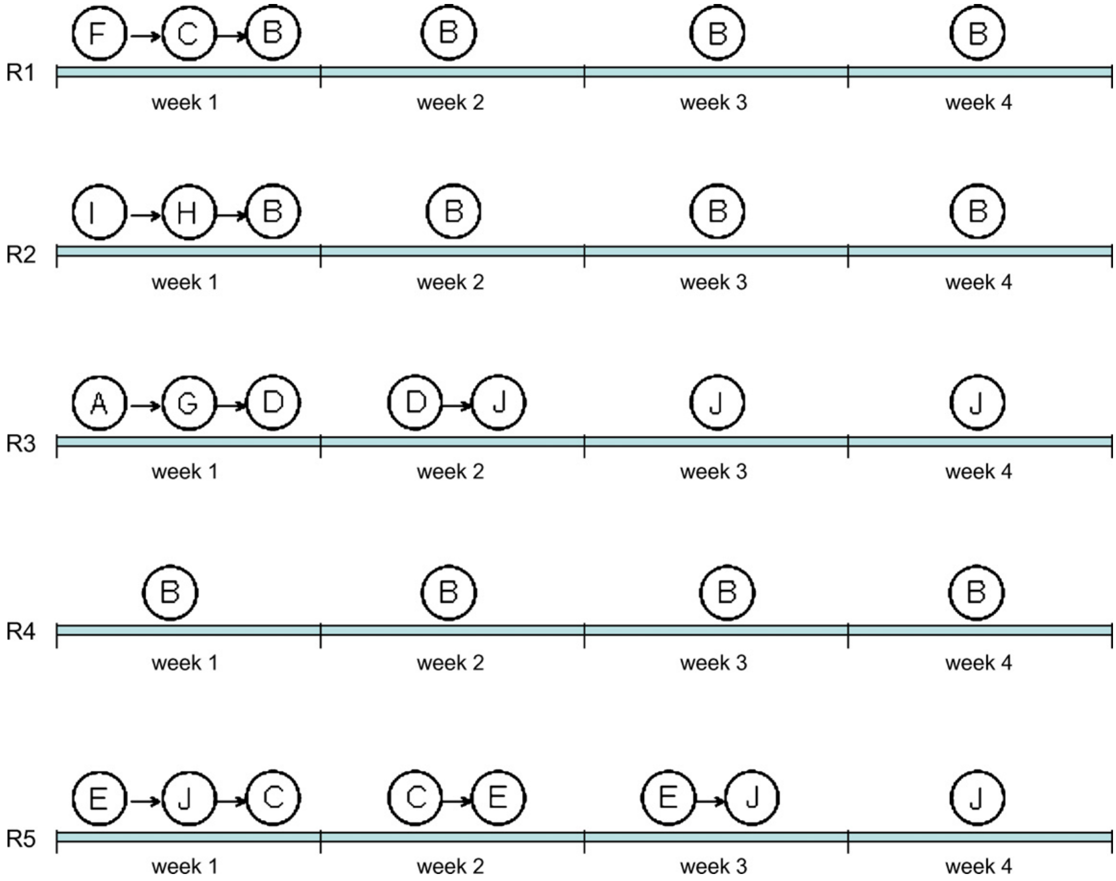

Fig. 17. Optimal sequence obtained by the upper level for 4 weeks of time horizon for Example 4.

isfied at the specified due dates. In order to deduce a feasible schedule, we resolved the problem by replacing constraint (24) by (25) and by subtracting term ($\sum_i \sum_t \mathrm{Pen}_{i,t} \times \Delta_{i,t}$) from the objective function (26) where the value of $\mathrm{Pen}_{i,t}$ is selected as 0.1$/kg for all products and all time periods. This value can be interpreted as a price for loss of customer satisfaction.

Model and solution statistics for this example are shown in Table 11. Note that compared to Table 2, there is an increase in the number of continuous variables, number of constraints and the solution time. Note also that the profit has dropped from $105,021 to $67,469. This decrease in the profit is due to the fact that, some of the specified demands were not met on the specified due dates and the objective function was penalized

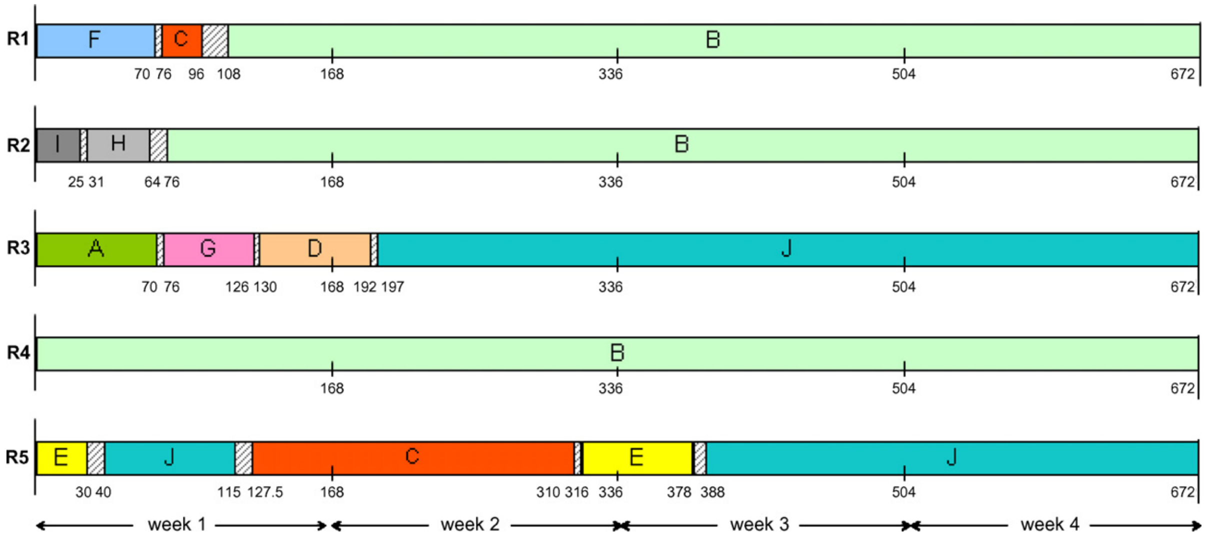

Fig. 18. Gantt chart obtained by the lower level for 4 weeks of time horizon for Example 4.

Table 11
Model and solution statistics for Example 5 for 4 weeks

Method	Number of binary variables	Number of continuous variables	Number of equations	Time (CPUs)	Solution ($)
Proposed algorithm				3.22	67,469
Problem UB	216	441	495	2.36	67,469
Problem LB	128[a]	831	757	0.86	67,469

[a] Not the actual size; numbers are prior to further processing and variable fixing.

Table 12
Unsatisfied demands for Example 5

	T1	T2	T3	T4
A	15,244	90,000	10,222	0
B	0	0	0	0
C	0	0	0	0
D	40,000	0	0	0
E	0	0	0	0

Table 13
Unsatisfied demands within the horizon

	T1	T2	T3	T4
Original solution				
A	15,244	90,000	10,222	0
B	0	0	0	0
C	0	0	0	0
D	40,000	0	0	0
E	0	0	0	0
		T2	T3	T4
Iteration 1				
A		62,844	711	0
B		80,000	0	
C		0	0	0
D		32,333	60,000	40,000
E		0	0	0
			T3	T4
Iteration 2				
A			67,108	0
B			80,000	0
C			0	0
D			14,333	40,000
E			60,000	0
				T4
Iteration 3				
A				65,643
B				80,000
C				41,000
D				68,666
E				0

for these unsatisfied demands. Table 12 illustrates the unsatisfied demands which amount to a total of 155,466 kg. While the specified demands have been satisfied for products B, C and E throughout the time horizon, 15,244 kg of A in the first week, 90,000 of A in the second week, 10,222 of A in the third week and finally 40,000 of D in the first week have been left unsatisfied.

Fig. 19 shows the Gantt chart for the schedule obtained in this example when we have the shortfalls as shown in Table 12. If we are to compare this Gantt chart with that of Example 2 (see Fig. 14), we can see that in Fig. 14, once the model satisfies the minimum specified demands for all the products, it allocates the rest of the available time for the production of Product C, whereas in Fig. 19, the model tries to minimize the demand violations. Note that, for implementation of this schedule one simple option would be to process the unsatisfied demands at the end of the 4-week horizon.

However, for the cases where the concern is to deliver the unsatisfied demands so as to reduce the tardiness, we outline the following procedure. The basic idea is to recursively accept the schedules until the time period, t, where unsatisfied demands are observed. Then, we resolve the problem where unsatisfied demands from time period t are carried over to the subsequent time period $t+1$. In other words, time period $t+1$ is solved for the original demands plus the unsatisfied demands from period t, while periods greater than $t+1$ are solved for the original demand values. While doing so, we also impose the constraint that the shortfalls from period t should be satisfied at period $t+1$. However, enforcing this hard demand constraint may result in an infeasible solution. If that is the case, we implement constraint (25) for the shortfalls but with a higher penalty. The total slacks obtained at time period $t+1$ are then carried on to the next period and the same procedure is applied until the last time period is reached.

Table 13 shows the total unsatisfied demands for each product for each time period at each iteration. While the profit obtained with the schedule with unsatisfied demands was $67,469, it drops to $38,482 in the first iteration, it further drops to $13,253 in the second iteration and finally drops to−$6525 in the third iteration.

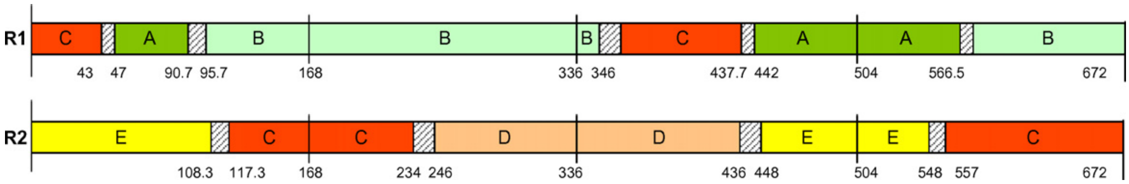

Fig. 19. Gantt chart for Example 5 for 4 weeks.

Note that this is due to the fact that the total unfulfilled demand increases from 155,466 kg (Table 12) where we minimize total violation, to 255,309 kg (Table 13) where we try to minimize tardiness.

7. Conclusions

We have addressed the simultaneous planning and scheduling of single-stage continuous plants with parallel units. We have presented a slot based MILP model, which is the extension of the work by Erdirik-Dogan and Grossmann (2006) for single units to the case of multiple parallel units. While this model is effective for small sized problems and short time horizons, the MILP model becomes computationally expensive to solve for large problems with long time horizons. In order to address this problem we propose a bi-level decomposition scheme. While the proposed decomposition approach is an extension of the work of Erdirik-Dogan and Grossmann (2006), the difference lies on how the upper level model is defined. This model is not only concerned with the assignments of products to units and time periods, but also with the effects of sequencing. Accounting for the scheduling explicitly at the upper level made it possible to successfully integrate planning and scheduling by obtaining very tight upper bounds.

The results show that the proposed upper level model yields significant improvements over the previous upper level model. Global optimal solutions are obtained by the upper level when there are no subcycles. It is important to note that among all the instances we have solved so far, only one instance exhibited a subcycle. Application of the proposed decomposition approach has proved to be very effective requiring only a single iteration between the master and the subproblem for all the instances we have solved so far. Therefore, compared to the full-space method, the proposed approach decreased the solution times by more than an order of magnitude. However, we should point out that the upper level model is now the bottleneck in computation due to the sequencing decisions incorporated at that level, which results in a rapid increase in the computational time as increasing number of time periods. Therefore, the trade-off between decreasing the number of iterations by obtaining tight upper bounds and increasing the computational effort of solving the problem should be investigated in future work.

Acknowledgments

The authors would like to acknowledge financial support from the Pennsylvania Infrastructure Technology Alliance, Institute of Complex Engineered Systems, and from the National Science Foundation under Grant no. DMI-0556090. We would like to thank Dr. Jeffrey Kelly from Honeywell for bringing constraints (11) and (12) to our attention.

Appendix A. On the Cardinality of Constraints (6), (11) and (12)

Constraint (6) has $|I| \times |I| \times |M| \times |L-1| \times |T| - |I| \times |M| \times |L-1| \times |T|$ number of constraints. Constraints (11) and (12) on the other hand lead to $|I| \times |M| \times |L| \times |T| + |I| \times |M| \times |L-1| \times |T|$constraints. The difference between the size of constraint (6) and constraints (11) and (12) is equal to $|I| \times |M| \times |T| \times [|I| \times |L-1| - 2 \times |L-1| - |L|]$. If the number of postulated slots are set equal to the number of products, $|I| = |L|$, than we can replace $|L|$ with $|I|$ which leads to $|I| \times |M| \times |T| \times [|I| \times |I-1| - 2 \times |I-1| - |I|]$ which is equal to $|I| \times |M| \times |T| \times [|I|^2 - 4 \times |I| + 2]$. Therefore, constraint (6) will lead to larger number of constraints provided that the cardinality of the product set, $|I|$ is greater than or equal to 4.

Appendix B. Proof of Property 1

Sum constraint (6) over i to obtain $\sum_i Z_{i,k,m,l,t} \geq \sum_i W_{i,m,l,t} + \sum_i W_{k,m,l+1,t} - \sum_i 1$. By substituting (1), this becomes

$$\sum_i Z_{i,k,m,l,t} \geq 1 + \sum_i W_{k,m,l+1,t} - N_m \tag{B.1}$$

Similarly, sum constraint (6) over k and replace $\sum_k W_{k,m,l+1,t}$ by 1 using constraint (1) to obtain

$$\sum_k Z_{i,k,m,l,t} \geq 1 + \sum_k W_{i,m,l,t} - N_m \tag{B.2}$$

Since $\sum_i W_{k,m,l+1,t}$ in (B.1) is independent of i, it can be replaced by $N_m \times W_{k,m,l+1,t}$, and similarly in (B.2) we can replace $\sum_k W_{i,m,l,t}$ by $N_m \times W_{i,m,l,t}$ to obtain

$$\sum_i Z_{i,k,m,l,t} \geq N_m.(W_{k,m,l+1,t} - 1) + 1 \tag{B.3}$$

$$\sum_k Z_{i,k,m,l,t} \geq N_m.(W_{i,m,l,t} - 1) + 1 \tag{B.4}$$

If $W_{k,m,l+1,t} = 1$ then the right hand side of (B.3) becomes 1. If $W_{k,m,l+1,t} = 0$ then the right hand side of (B.3) becomes $1 - N_m$. However, since $Z_{i,k,m,l,t} \geq 0$, the right hand side of (B.3) cannot be negative. Therefore, (B.3) does not depend on N_m, and hence N_m can be reduced to 1.

Same argument follows for (B.4).

Hence, (B.3) and (B.4) become

$$\sum_i Z_{i,k,m,l,t} \geq W_{k,m,l+1,t} \tag{B.5}$$

and

$$\sum_k Z_{i,k,m,l,t} \geq W_{k,m,l,t} \tag{B.6}$$

By constraint (8), if $W_{k,m,l+1,t}$ is zero then $Z_{i,k,m,l,t}$ must be *zero*. Also, by (B.5) and (8), if $W_{k,m,l+1,t}$ is 1, then $Z_{i,k,m,l,t}$ must be 1 since its upper bound is 1. Hence, the inequality in (B.5) can be replaced by equality in (12). Following the same argument, we can replace the inequality in (B.6) by the equality in (11).

Thus, it follows that (6)-(8) and (11)–(12) are equivalent since the latter can be derived from (6)–(8) as has been proved. To prove that (11)–(12) are at least as tight, we set $W_{i,m,l,t} = 1/|I(m)|$ and $W_{k,m,l,t} = 1/|I(m)|$ so that (1) is satisfied provided

$|I(m)| \geq 2$. Setting $Z_{i,k,m,l,t} = 0$, (6)–(8) are satisfied. However, (11)–(12) are violated. Thus, the feasible space of (11)–(12) is contained in the feasible space of (6)–(8). Also, since every point satisfied by (11)–(12) must be feasible for (6)-(8) by the above derivation from (B.1) to (B.6), it follows that (11)–(12) is at least as tight as (6)–(8).

References

Alle, A., Papageorgiou, L. G., & Pinto, J. M. (2004). A mathematical programming approach for cyclic production and cleaning scheduling of multistage continuous plants. *Computers and Chemical Engineering*, *28*, 3–15.

Balas, E., & Jeroslow, R. (1972). Canonical cuts on the unit hypercube. *SIAM J. Appl. Math.*, *23*, 61–79.

Bassett, M. H., Pekny, J. F., & Reklaitis, G. V. (1996). Decomposition techniques for the solution of large-scale scheduling problems. *American Institute of Chemical Engineering Journal*, *42*, 3373–3384.

Birewar, D. B., & Grossmann, I. E. (1990). Simultaneous production planning and scheduling in multiproduct batch plants. *Industrial Engineering and Chemical Research*, *29*, 570–580.

Dimitriadis, A. D., Shah, N., & Pantelides, C. C. (1997). RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers and Chemical Engineering*, *21*, 1061–1066.

Erdirik-Dogan, M., & Grossmann, I. E. (2007). Optimal production planning models for parallel batch reactors with sequence-dependent changeovers. *American Institute of Chemical Engineering Journal*, *53*, 2284–2300.

Erdirik-Dogan, M., & Grossmann, I. E. (2006). A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Industrial Engineering and Chemical Research*, *45*, 299–315.

Jain, V., & Grossmann, I. E. (1998). Cyclic scheduling of continuous parallel process units with decaying performance. *American Institute of Chemical Engineering Journal*, *44*, 1623–1636.

Jia, Z., & Ierapetritou, M. (2004). Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Computers and Chemical Engineering*, *20*, 1001–1019.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short term scheduling of batch operations. 1. Milp formulation. *Computers and Chemical Engineering*, *17*, 211–227.

Lee, H., Pinto, J. M., Grossmann, I. E., & Park, S. (1996). Mixed-integer linear programming model for refinery short-term scheduling of crude-oil unloading with inventory management. *Industrial Engineering and Chemical Research*, *35*, 1639.

Mendez, C. A., & Cerda, J. (2002). An efficient milp continuous-time formulation for short-term scheduling of multiproduct continuous facilities. *Computers and Chemical Engineering*, *26*, 687–695.

Miller, D. L., & Pekny, J. F. (1991). Exact solution of large asymmetric traveling salesman problems. *Science*, *251*, 754–761.

Nemhauser, G., & Wolsey, L. (1998). *Integer and Combinatorial Optimization*. John Wiley and Sons.

Pantelides, C. C. (1994). Unified frameworks for the optimal process planning and scheduling. In *Proceedings of the second conference on foundations of computer-aided process operations* (Vol. 1, pp. 253–274). Austin, TX: CACHE.

Papageorgiou, L. G., & Pantelides, C. C. (1996). Optimal campaign planning/scheduling of multipurpose batch/semi-continuous plants. 2. A mathematical decomposition approach. *Industrial Engineering and Chemical Research*, *35*, 510.

Pinto, J. M., & Grossmann, I. E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants. *Computers and Chemical Engineering*, *18*, 797–816.

Sahinidis, N. V., & Grossmann, I. E. (1991). MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Computers and Chemical Engineering*, *15*, 85–1036.

Shah, N. (1996). Mathematical programming techniques for crude-oil scheduling. *Computers and Chemical Engineering*, *20S*, 1227.

Sung, C., & Maravelias, C. T. (2007). An attainable region approach for production planning of multiproduct processes. *American Institute of Chemical Engineering Journal*, *53*, 1298–1315.

Wilkinson, S. J., Shah, N., & Pantelides, C. C. (1996). Aggregate modeling of multipurpose plant operation. *Computers and Chemical Engineering*, *19*(Suppl. 1,11), 583–588.