A CP APPROACH TO THE SCHEDULING OF RESOURCE-CONSTRAINED MULTIPRODUCT CONTINUOUS FACILITIES

L.J. ZEBALLOS and G.P. HENNING

INTEC, Universidad Nacional del Litoral - CONICET
Güemes 3450, Santa Fe, Argentina.
{zeballos,ghenning}@intec.unl.edu.ar

Abstract— **This contribution presents a new Constraint Programming (CP) formulation to address the optimal operation of multiproduct continuous processes. In particular, facilities having continuous stages, which manufacture intermediate or final products and a limited number of intermediate storage tanks, are tackled. The problem involves finding (i) the optimal sequence of production runs and (ii) the optimal agenda of usage of each storage tank. The proposed formulation is able to handle different features found in industrial environments such as dissimilar machines, sequence-dependent changeover times, forbidden campaign-machine assignments, production runs´ length constraints, etc. Moreover, one of the main contributions of this paper is the handling of constrained resources of various types, such as intermediate storage tanks of limited capacity and qualified manpower required at some stages.**

Keywords— **Multiproduct facilities, Renewable and non-renewable resources, Constraint Programming.**

I. INTRODUCTION

The short-tem scheduling problem of multiproduct continuous plants has gained increasing attention in recent years. This is because many industrial plants usually involve continuous stages. Also, an additional motivation arises due to the increasing need for moving towards continuous flexible multiproduct plants that quickly respond to demand changes and to the need of processing a variety of products.

Most researchers have tackled this problem by formulating either, a mixed-integer linear mathematical model or a mixed-integer nonlinear one (Zhang and Sargent, 1998, Ierapetritou and Floudas, 1998, Méndez and Cerdá, 2002). However, these formulations have several disadvantages. On the one hand, the consideration of industrial problems leads to large-size models, which are generally hard to solve. On the other hand, certain approaches introduce hard assumptions about the problem. For instance, they assume that the production rate of intermediate products is greater than their associated consumption rate, which is not always true in practice.

This contribution presents a novel Constraint Programming (CP) approach to the previous problem. Constraint Satisfaction (CS) (Brailsford and Potts, 1999) is a relatively new approach, originated in the computer science and artificial intelligence communities. However, it has already been adopted in the Process Systems Engineering field (Maravelias and Grossmann, 2004). A constraint satisfaction problem (CSP) can be interpreted as the problem of selecting, from finite domains, values to be assigned to each variable so that every element of the set of constraints is satisfied. The implementation of special-purpose algorithms that are widely employed to solve CSPs is called Constraint Programming (CP). It uses tree search combined with domain reduction (DR) and constraint propagation (CPG) procedures to solve CSPs. CP implements systematic search procedures that, by fixing the order in which the variables should be chosen and a way to select a value from a variable domain, supply a proper assignment of values to the problem variables. When the search fixes a variable value, DR and CPG algorithms play an active role. Thus, if the domain of a variable participating in a given constraint is modified, the DR algorithm changes the domains of all the other variables in that constraint. Additionally, when a variable's domain is altered, the effects of this change are then spread by CPG to the whole set of constraints that interact with that variable. A wide variety of algorithms have been developed to efficiently propagate constraints and reduce domains so as to increase the global performance of CP. These algorithms are nowadays included in commercial CP systems such as ILOG solver and ECLiPSe or in free software such as OZ.

The goal of this paper is to propose and assess the quality of a new constraint programming (CP) formulation addressing the scheduling problem stated before. The proposed formulation is based on the constraint programming language (OPL) supported by ILOG Solver (2003a) and employs some specific scheduling constructs available in ILOG Scheduler (2003b).

II. PROBLEM STATEMENT

This work considers a generic resource-constrained multiproduct plant with two continuous stages producing intermediate and final products and storage tanks for temporary inventory of intermediates. Since the production and consumption rates of intermediate products are generally dissimilar, tanks are almost always required. However, in most of the cases there is a reduced number of tanks having a limited capacity. In addition, it is considered that certain tasks require manpower, but the number of available people is lower than the requested one.

In order to build the CP model, the following assumptions are considered: i) Requirements of final products are previously defined. ii) The number of campaigns for each intermediate and final product is predefined and taken as problem data. iii) Each intermediate and final product is produced in only one campaign (production run). iv) Intermediate and final product campaigns can only be carried out in some of the processing units. v) A single campaign manufacturing an intermediate product supplies it to several campaigns demanding such product. vi) Production rates only depend on the material to be produced. vii) A set of tanks of limited capacity for temporary storage of intermediates is available. viii) Any intermediate product obtained in a given campaign must be stored in just one tank. (Campaign split is not allowed). ix) Intermediate products can be stored in storage tanks or directly transferred to the following stage. x) Changeover-times to be considered in processing units and storage tanks are sequence dependent. xi) The initial inventory of any intermediate product is equal to zero. xii) The available storage capacity for final products is unlimited. xiii) Production rates of intermediate products can take any arbitrary value in relation to the consumption rates of such intermediate products by means of a set of final product campaigns.

The model aims at establishing (i) the assignment of campaigns to processing units and (ii) the proper sequence of campaigns assigned to the same unit, as well as (iii) the assignment of intermediate products to tanks and the sequence of those allocated to the same tank. Finally, the timing of all manufacturing and storage operations is another output of the model.

III. CONSTRAINT PROGRAMMING MODEL

A. Nomenclature

Sets

P^I Intermediate products

P^F Final products

$P^F_{p'}$ Final products requiring intermediate product *p'* ($P^F_{p'} \subseteq P^F$)

J Set of processing units

$FJIP_{p'}$ Set of forbidden unit-intermediate product (*p'*) pairs

$FJFP_p$ Set of forbidden unit-final product (*p*) pairs

In turn, the model handles the following ILOG sets

JI Represents as alternative units the equipment to be employed in the manufacture of intermediate products. Each element *j* belonging to the set *JI* is an ILOG element defined as a unary resource (cannot be shared by two processing activities at a given time)

JF Represents as alternative units the equipment to be employed in the manufacture of final products. Each element *j* belonging to set *JF* is another element defining a unary resource

T Considers the tanks that are used for the storage of intermediate products as alternative units. Therefore, each element *t* belonging to the set *T* is considered a unary resource

Mp Represents a set of qualified people as a discrete resource. A certain amount of this resource is consumed at the start time of an activity demanding it and the same amount is released at its end time.

Parameters

r_p production rate of final product *p*

$r^{\min}_{p'}$ minimum production rate of intermediate product *p'*

$r^{\max}_{p'}$ maximum production rate of intermediate product *p'*

$\rho_{p'p}$ amount of intermediate product $p' \in P^I$ required per unit of final product $p \in P^F$

d_p requirement of final product $p \in P^F$

ru_j ready time of unit *j*

rc_p release time of the campaign manufacturing product *p*

v_t maximum capacity of tank $t \in T$

m amount of manpower that is available during the scheduling horizon

$\tau_{pp'}$ changeover time between campaigns manufacturing products *p* and *p'*, which is independent of the chosen unit

$\sigma_{pp'}$ changeover time between the storage of products *p* and *p'* in any available tank

Model variables

PC_{pq} binary variable indicating that the campaign that manufactures a final product *p* from intermediate *p'* is being processed at the time a different campaign manufacturing another final product *q*, that requires the same intermediate, ends

FC_{pq} binary variable denoting that the campaign that manufactures a final product *p* from intermediate *p'* ends before another campaign manufacturing another final product *q*, that requires the same intermediate, has finished.

Task-timing related variables

The main variables employed in the CP model are special modelling elements of the ILOG OPL studio. These special elements are characterized by means of starting, duration and ending time variables. $TaskI_{p'}$ and $TaskF_p$ are two types of activities to be included in the model to describe production campaigns of intermediate and final products *p'* and *p*, respectively. $TaskST_{p'}$ is another activity used in the model to represent the storage task of intermediate product *p'*.

In addition, the activity $CO_{p'p}$ is included to describe the time during which the intermediate product *p'*, required to manufacture the final product *p*, is consumed by the campaign manufacturing product *p*, while *p'* is still being produced by $TaskI_{p'}$. Fig. 1 shows a typical situation in which $CO_{p'p}$ appears because the $TaskF_p$ consumes the intermediate product *p'* while it is still being produced by $TaskI_{p'}$. As seen in the figure, the duration associated to the variable $CO_{p'p}$ is determined by the start time of production run $TaskF_p$ and the completion time of campaign $TaskI_{p'}$.

Aggregate performance measure

Mk Makespan

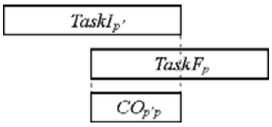

Fig. 1. Conceptual interpretation of variable $CO_{p'p}$

B. Problem Constraints

Assignment constraints

Constraints (1)-(3) are assignment relations that prescribe that each task must be assigned to one unit. These constraints resort to the *requires* special construct, provided by ILOG Scheduler (ILOG, 2003b) to enforce the assignment of just a single unit belonging to a given set of resources. Constraint (1) prescribes that each production campaign $TaskI_{p'}$ of intermediate product *p'* must be assigned to a unit *j* belonging to the set of alternative resources *JI*. Constraint (2) assigns each storage task corresponding to intermediate product *p'* at most to one tank *t* belonging to the set of alternative resources *T*. Constraint (3) enforces the assignment of every campaign $TaskF_p$ manufacturing a final product *p* to a unit *j* belonging to the set of alternative resources *JF*.

Constraint (4) is another assignment constraint that represents the requirement of qualified manpower *Mp*. Each campaign manufacturing a final product *p* requires *n* qualified persons belonging to the set of discrete resources *Mp*.

$$TaskI_{p'}\ requires(1)\ JI \quad \forall\ p' \in P^I \tag{1}$$

$$TaskST_{p'}\ requires(1)\ T \quad \forall\ p' \in P^I \tag{2}$$

$$TaskF_p\ requires(1)\ JF \quad \forall\ p \in P^F \tag{3}$$

$$TaskF_p\ requires(n)\ Mp \quad \forall\ p \in P^F \tag{4}$$

Handling of sequence-dependent changeover times

These changeover times may apply to both processing units and storage tanks. They will be modeled by means of special constructs of the adopted constraint language. In order to handle transition times that depend on products, tasks and processing units need to be declared in a special way by using the state and transition type constructs. Thus, by means of the declaration $Task_p$ *TransitionType* $state_p$, any $Task_p$ (production or storage task associated to product *p*) is linked to the enumerated value $state_p$. Moreover, resources (equipment, tanks) where the task takes place are considered state resources. This means that they have alternative states, having, in this case, as many as the different products that could be processed/stored in the resource. Each of these resources is associated with a transition matrix containing the transition times corresponding to the different *p-p'* pairs. For instance, to consider the changeover time associated to the tank cleaning task that takes place between the storage of two consecutive products, the CP model will include the following declaration: $TaskST_{p'}$ *TransitionType* $stateST_{p'}$. By means of this statement each storage task is associated to the enumerated value $stateST_{p'}$, that has been declared as of transition type. In addition, tanks where storage tasks take place are considered state resources and consequently, they are associated with a transition matrix σ. Thus, given two transition types *p* and *p'*, representing the predecessor and successor products, the $\sigma_{pp'}$ transition time is obtained. This time will be automatically included between the time allocated to the storage of products *p* and *p'*.

Forbidden assignments related constraints

Constraints (5) and (6) handle forbidden assignments by resorting to the special construct *activityHasSelectedResource*. This construct models the assignment of alternatives resources by acting as a predicate that evaluates to true when a certain processing task has been assigned to one unit belonging to a given set of alternative equipment. If the construct is negated, the whole constraint acts as a predicate that evaluates to true when the processing campaign has not been assigned to the specified unit. Constraint (5) prohibits the assignment of $TaskI_{p'}$, a campaign producing intermediate *p'*, to forbidden units *j* belonging to the set $FJIP_{p'}$. On the other hand, constraint (6) prevents the assignment of $TaskF_p$, a campaign manufacturing product *p*, to prohibited units *j* belonging to the set $FJFP_p$.

$$\begin{aligned}&not\ activityHasSelectedResource(TaskI_{p'}, JI, j)\\ &\forall\ j \in FJIP_{p'}, \forall\ p' \in P^I\end{aligned} \tag{5}$$

$$\begin{aligned}&not\ activityHasSelectedResource(TaskF_p, JF, j)\\ &\forall\ j \in FJFP_p, \forall\ p \in P^F\end{aligned} \tag{6}$$

Constraints setting production run lengths

Constraints (7) and (8) associate the length of each activity $TaskI_{p'}$, a campaign manufacturing the intermediate product *p'*, to the demand of those final products *p* that require *p'* as an intermediate product. In addition, these constraints determine the length of campaign $TaskI_{p'}$ by considering the maximum and minimum production rates of the product *p'*.

$$TaskI_{p'}.duration * r_{p'}^{\max} \geq \sum_{p \in P_{p'}^F} d_p * \rho_{p'p} \quad \forall\ p' \in P^I \tag{7}$$

$$TaskI_{p'}.duration * r_{p'}^{\min} \leq \sum_{p \in P_{p'}^F} d_p * \rho_{p'p} \quad \forall\ p' \in P^I \tag{8}$$

Timing constraints

Equations (9) and (10) compute the start time of production campaigns which are executed at the beginning of the scheduling horizon, taking into account both the final products′ release times and equipment ready times.

$$\sum_{j\in JI} activityHasSelectedResource(TaskI_{p'}, JI, j) * \left[Max(ru_j, rc_{p'})\right] \le TaskI_{p'}.start \quad \forall\ p' \in P^I \tag{9}$$

$$\sum_{j\in JF} activityHasSelectedResource(TaskF_p, JF, j) * \left[Max(ru_j, rc_p)\right] \le TaskF_p.start \quad \forall\ p \in P^F \tag{10}$$

Given two production campaigns, if one acts as a source of material and the other one as a sink, there must be a timing relation between them. Constraint (11) indicates that $TaskF_p$, requiring a product *p'* that is manufactured by campaign $TaskI_{p'}$, should not begin before than the start time of such production run.

$$TaskI_{p'}.start \le TaskF_p.start \quad \forall\ p \in P^F_{p'},\ \forall\ p' \in P^I \tag{11}$$

Constraints (12)-(17) are timing constraints that prescribe the temporal relationships associated to the storage task of any intermediate product *p'*. Constraint (12) establishes that the storage task of product *p'* should begin just at the same time the campaign that manufactures product *p′* ($TaskI_{p'}$) starts. Constraint (13) models the temporal relationship that must exist between the length of campaign $TaskI_{p'}$ producing an intermediate product *p'* and the duration of its storage task. Constraint (14) prescribes that a campaign $TaskF_p$ consuming an intermediate product *p'* must finish, just at the time or before the end time of the storage task corresponding to the intermediate product *p'*. Constraints (13) and (14) are enforced if and only if the duration of the storage task is greater than zero. It is important to remark that when a consuming campaign is directly fed by an intermediate product campaign, the storage task associated to this last one becomes a dummy task, having a duration equal to zero. When no storage is required, constraint (15) prescribes that the consuming campaign $TaskF_p$ must finish, just at the time or before, the ending time of campaign $TaskI_{p'}$, which supplies the *p'* material. Finally, constraint (16) indicates that the storage task of product *p'* does not need a tank *t* belonging to the set of alternative resources *T* if and only if the storage task does not exist.

$$TaskI_{p'}.start = TaskST_{p'}.start \quad \forall\ p' \in P^I \tag{12}$$

$$TaskST_{p'}.duration \ge TaskI_{p'}.duration <=> TaskST_{p'}.duration > 0 \quad \forall\ p' \in P^I \tag{13}$$

$$TaskST_{p'}.end \ge TaskF_p.end <=> TaskST_{p'}.duration > 0 \quad \forall\ p \in P^F_{p'}, \forall\ p' \in P^I \tag{14}$$

$$TaskST_{p'}.duration = 0 \ => TaskI_{p'}.end \ge TaskF_p.end \quad \forall\ p \in P^F_{p'},\ \forall\ p' \in P^I \tag{15}$$

$$TaskST_{p'}.duration = 0 <=> TaskST_{p'}\ requires(0)\ T \quad \forall\ p' \in P^I \tag{16}$$

$$\sum_{t\in T} activityHasSelectedResource(TaskST_{p'}, T, t) \le TaskST_{p'}.duration \quad \forall\ p' \in P^I \tag{17}$$

Equation (17) establishes the relationship between two characteristics of the storage task, its assignment and duration. When the storage task of intermediate *p'* is assigned to one of the alternative tanks, its duration must be greater than zero. In fact, this constraint is not strictly necessary but it is included in the model because it helps the solver to speed-up the search for a solution.

Constraints (18)-(20) prescribe the duration of the $CO_{p'p}$ activity, depending on the relationship between campaigns $TaskF_p$ and $TaskI_{p'}$, which require and manufacture product *p'*, respectively. Constraint (18) limits the maximum duration of activity $CO_{p'p}$. This maximum duration cannot exceed the length of the campaign producing *p*. In turn, constraint (19) is active when the campaign $TaskF_p$ is being executed while the production run $TaskI_{p'}$ ends. Thus, the time during which the intermediate product *p'* is consumed by the campaign $TaskF_p$ is greater than zero, but equal to the time elapsed between the start time of the consuming campaign $TaskF_p$ and the completion time of $TaskI_{p'}$. On the other hand, when the production run $TaskF_p$ starts after $TaskI_{p'}$ ends, there is no overlapping between them. Consequently, constraint (20) is active and the duration of activity $CO_{p'p}$ is equal to zero.

$$CO_{p'p}.duration <= d_p / r_p \quad \forall\ p \in P^F_{p'},\ \forall\ p' \in P^I \tag{18}$$

$$TaskI_{p'}.end \ge TaskF_p.start\ \&\ TaskI_{p'}.end \le TaskF_p.end \ => CO_{p'p}.duration \le TaskI_{p'}.end - TaskF_p.start \ \ \& \ \ CO_{p'p}.duration > 0 \quad \forall\ p \in P^F_{p'}, \forall\ p' \in P^I \tag{19}$$

$$TaskI_{p'}.end \le TaskF_p.start <=> CO_{p'p}.duration = 0 \quad \forall\ p \in P^F_{p'},\ \forall\ p' \in P^I \tag{20}$$

Material balance constraints

These constraints are included in the model to ensure that tank capacities are not exceeded as well as to guarantee that final product campaigns have sufficient amounts of intermediate products in order to be executed. These balance constraints will be posed at critical time points such as the end of the intermediate product campaigns as well as the start and end times of the final product ones.

Constraint (21) enforces the amount of material stored in each tank, for each product that is kept, to be lower than or equal to the tank capacity v_t. Thus, the left hand side of constraint (21) considers the capacity of the tank in which the campaign $TaskI_{p'}$ manufacturing *p'* is assigned and the right hand side takes into account the amount of such product that actually needs to be stored, which is computed as follows. The first term on the right hand side considers the total amount of product *p*' manufactured by the production run $TaskI_{p'}$ and the second term represents the consumption of product *p'* up to the completion time of $TaskI_{p'}$. The difference between these two terms is the amount that actually needs to be stored in the tank.

$$\sum_{t\in T} activityHasSelectedResource(TaskST_{p'},T,t) * v_t \geq \sum_{q\in P^F_{p'}} d_q * \rho_{p'q} - \sum_{p\in P^F_{p'}} CO_{p'p}.duration * r_p * \rho_{p'p} \quad \forall\, p'\in P^I \tag{21}$$

Before presenting the remaining balance equations some extra timing constraints will be introduced. They allow defining binary variables FC_{pq} and PC_{pq}. The first variable assumes a value equal to one when the campaign that manufactures a final product p ends before another campaign that manufactures the final product q has finished. In turn, the second variable takes a value equal to one to indicate that the campaign manufacturing a final product p is still being processed at the time another production run manufacturing a different final product q ends. In both cases, campaigns manufacturing p and q demand the same intermediate product p'.

Thus, constraints (22)-(25) determine the temporal relations existing between every pair of final product campaigns demanding the same intermediate product p'. Constraint (22) enforces the value of variables FC_{pq} and PC_{pq} to be equal to one and zero, respectively, when the production run $TaskF_p$ finishes before the completion time of $TaskF_q$. On the other hand, constraint (23) prescribes that the variable FC_{pq} must be equal to zero, when the completion time of $TaskF_p$ is greater than or equal to the finish time of the production campaign $TaskF_q$. Constraint (24) enforces variables FC_{pq} and PC_{pq} to take the values zero and one, respectively, if and only if, the completion time of $TaskF_p$ is greater than or equal to the finish time of campaign $TaskF_q$ and $TaskF_p$ starts before the completion time of the production run $TaskF_q$. In addition, constraint (25) prescribes that the variables FC_{pq} and PC_{pq} must be equal to zero, if and only if, the start time of $TaskF_p$ is greater than or equal to the completion time of campaign $TaskF_q$.

$$TaskF_p.end < TaskF_q.end \;\Leftrightarrow\; FC_{pq}=1 \;\&\; PC_{pq}=0 \quad p\neq q, \forall\, p,q\in P^F_{p'},\; \forall\, p'\in P^I \tag{22}$$

$$TaskF_p.end \geq TaskF_q.end \;\Rightarrow\; FC_{pq}=0 \quad p\neq q, \forall\, p,q\in P^F_{p'},\; \forall\, p'\in P^I \tag{23}$$

$$TaskF_p.end \geq TaskF_q.end \;\&\; TaskF_p.start < TaskF_q.end \;\Leftrightarrow\; FC_{pq}=0 \;\&\; PC_{pq}=1 \quad p\neq q, \forall\, p,q\in P^F_{p'},\; \forall\, p'\in P^I \tag{24}$$

$$TaskF_p.start \geq TaskF_q.end \;\Leftrightarrow\; FC_{pq}=0 \;\&\; PC_{pq}=0 \quad p\neq q, \forall\, p,q\in P^F_{p'},\; \forall\, p'\in P^I \tag{25}$$

Continuing with tank capacity constraints, constraint (26) enforces the amount of intermediate product stored in a given tank, at the start time of each campaign $TaskF_p$ that consumes such product, to be lower than or equal to the associated tank capacity v_t. The left hand side of constraint (26) considers the capacity of the tank in which the production run that manufactured product p' is assigned. The first term on the right hand side represents the amount of intermediate product p' manufactured by the $TaskI_{p'}$ up to the start time of campaign $TaskF_p$. If $TaskF_p$ starts after the completion time of $TaskI_{p'}$, the amount of product p' that was produced is calculated by taking into account the whole duration of $TaskI_{p'}$. When $TaskF_p$ begins before the completion time of $TaskI_{p'}$, the amount of product p' is calculated by considering the time elapsed between the start times of both campaigns, $TaskI_{p'}$ and $TaskF_p$. The second term represents the amount of intermediate product consumed by those campaigns that require product p' and finish before the completion time of $TaskF_p$ ($FC_{rp}=1$). In this case, three situations can be found. First, if $TaskF_r$ ends before the start time of $TaskF_p$, the amount of product p' that is consumed is calculated by taking into account the total duration of $TaskF_r$. Second, when $TaskF_r$ begins before the start time of $TaskF_p$ and finishes after it, the required amount of product p' is calculated by considering the time elapsed between the start times of production runs $TaskF_r$ and $TaskF_p$. Third, if $TaskF_r$ starts after the beginning of $TaskF_p$, $TaskF_r$ has not consumed product p' up to the point the balance is made. The third term on the right hand side of constraint (26) considers the amount of intermediate product consumed by those campaigns $TaskF_u$ that require product p' and are still being processed at the completion time of $TaskF_p$ ($PC_{up}=1$). In this case, two situations can arise. The first one occurs when $TaskF_u$ starts before the beginning of $TaskF_p$, the amount of product p' that is required is calculated by considering the time elapsed between the start times of production runs $TaskF_u$ and $TaskF_p$. The second situation takes place if $TaskF_u$ begins after the start time of campaign $TaskF_p$; Then, $TaskF_u$ has not consumed product p' up to the point the balance is made.

$$\begin{aligned}
&\sum_{t\in T} activityHasSelectedResource(TaskST_{p'},T,t) * v_t \geq \\
&\min(TaskI_{p'}.duration\;, (TaskF_p.start - TaskI_{p'}.start)) * \\
&\qquad (\sum_{q\in P^F_{p'}} d_q * \rho_{p'q} / TaskI_{p'}.duration) - \\
&\qquad \sum_{r\in P^F_{p'}\; r\neq p} \min(TaskF_r.duration, \\
&\max((TaskF_p.start - TaskF_r.start), 0)) * r_r * \rho_{p'r} * FC_{rp} - \\
&\sum_{u\in P^F_{p'}\; u\neq p} \max((TaskF_p.start - TaskF_u.start), 0) * r_u * \rho_{p'u} * PC_{up} \\
&\qquad \forall\, p\in P^F_{p'},\; \forall\, p'\in P^I
\end{aligned} \tag{26}$$

Constraint (27) guarantees that the amount of intermediate product p' manufactured by $TaskI_{p'}$ before the completion time of a given campaign $TaskF_p$, is to be greater than or equal to the amount of this intermediate product consumed by all production runs requiring it up to such point. This constraint must be included in the model due to the assumption that states that the production rates of intermediates products can be lower than their associated consumption rates. The left hand side of constraint (27) represents the amount of intermediate product p' manufactured by $TaskI_{p'}$ before the completion time of the production run $TaskF_p$. When $TaskF_p$ finishes after the end of $TaskI_{p'}$, the amount of *p'* that was produced is calculated by considering the whole duration of $TaskI_{p'}$. If $TaskF_p$ finishes before the end of $TaskI_{p'}$, the amount of *p'* that is considered takes into account the time elapsed between the start time of $TaskI_{p'}$ and the completion time of $TaskF_p$. The first term on the right hand side represents the amount of intermediate product consumed by the campaign $TaskF_p$ (production run for which the material balance is computed). The second term reflects the amount of intermediate product consumed by all the production runs $TaskF_r$ that also require product *p'* and finish before the completion time of campaign $TaskF_p$. The last term on the right hand side considers the consumption of those campaigns $TaskF_u$ that are still being processed at the completion time of $TaskF_p$. In this case, the amount of product *p'* that is demanded is calculated by considering the time elapsed between the start time of $TaskF_u$ and the end time of campaign $TaskF_p$.

$$\begin{aligned} &\min(TaskI_{p'}.duration\ ,(TaskF_p.end - TaskI_{p'}.start))* \\ &\quad (\sum_{q \in P^F_{p'}} d_q * \rho_{p'q} / TaskI_{p'}.duration) \geq \\ &\quad TaskF_p.duration * r_p * \rho_{p'p} + \\ &\sum_{r \in P^F_{p'}\ r \neq p} TaskF_r.duration * r_r * \rho_{p'r} * FC_{rp} + \\ &\sum_{u \in P^F_{p'}\ u \neq p} (TaskF_p.end - TaskF_u.start) * r_u * \rho_{p'u} * PC_{up} \\ &\qquad \forall\ p \in P^F_{p'},\ \forall\ p' \in P^I \end{aligned} \tag{27}$$

C. Objective Function

The problem objective is to minimize the time needed to complete the production requirements. Therefore, inequality (28) allows defining the makespan *MK* to be included as the objective function of the problem.

$$TaskF_p.end \leq Mk \qquad \forall\ p \in P^F \tag{28}$$

IV. COMPUTATIONAL RESULTS

In order to test the proposed CP model, different case studies based on an industrial fast moving consumer goods manufacturing plant (Schilling and Pantelides, 1996) were solved. The plant manufactures seven intermediate products from three distinct available base materials, which in turn are packed in different ways to produce fifteen final products. The manufacturing sequence associated to each final product is specified in Fig. 2. The first stage manufactures intermediate products in three mixers operating in a continuous mode. The second is a packing stage which is composed of five parallel packing lines also operating in continuous mode. The plant has three tanks of limited capacity to store the intermediate products. The adopted scheduling criterion is to minimize the makespan (*Mk*).

The performance of the model was evaluated by means of several test problem instances. One test problem was reported by Zhang and Sargent (1998). Seven new examples were obtained from this one by changing the requirements of final products, production rates, changeover times and forbidden campaign-machine assignments, and by also including manpower requirements. In addition, the new problem instances consider that some production rates of certain intermediate products are lower than or equal to their associated consumption rates, whereas previous authors assumed that intermediate products´ manufacturing rates are always higher than the rates of consumption of final product campaigns.

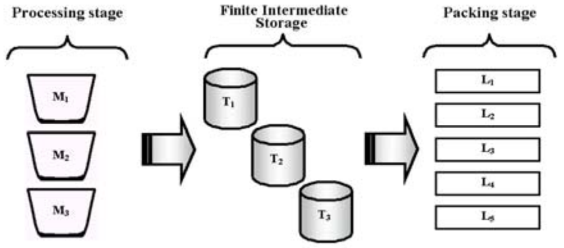

Fig. 2. Schematic representation of the process

As mentioned before, the first problem instance (#1) considered in this work corresponds to the case reported by Zhang and Sargent (1998). Examples #2 to #4 are new problem instances obtained from #1. Problem instances #5 to #8 are similar to #1 to #4, respectively, but the former ones also take into account manpower requirements. Due to lack of space, only one example and its associated solution are reported in this paper.

Tables 1 and 2 present data of the problem instance #6. Table 1 shows for each intermediate product its maximum and minimum production rates, the allowable units and the set of final products which require it. Moreover, Table 2 specifies for each final product its associated production rate, demand, suitable lines and changeover times. In addition, in this example, tanks have 140 tons of storage capacity. The case study also includes manpower limitations. It is considered that only four people are able to work in the last stage, in which one person per line is required.

Table 1. Intermediate products´ data

Intermediate product	Production rates (max-min) (ton/h)	Allowable units	Associated final products
I1	3-7	M1, M3	P2, P3
I2	6-12	M1, M3	P1, P4, P5
I3	5-10	M2, M3	P6, P7
I4	5-9	M3	P13, P14
I5	4-8	M1	P8, P10
I6	7-13	M1, M2	P12, P15
I7	5-9	M1, M2	P9, P11

The solution was found by using ILOG OPL Studio 3.7 with the embedded Scheduler Optimizer 6.0 release (ILOG, 2003a and 2003b). The computations were carried out on a Pentium IV PC (2.8 GHz) with 1 Gbyte of RAM memory. The proposed problem formulation includes 1268 constraints and 759 variables and the optimal schedule was obtained in 28.1 seconds of CPU time. The value of the objective function corresponding to this case is 99.5 h.

The optimal schedule for this case study is shown in Fig 3. This figure depicts the units' workload and, in its lower part, the inventory levels as well as the requirements of qualified manpower. Also, it shows the sequence of intermediate products assigned to each storage tank and the temporal variation of the intermediates' inventory during the scheduling horizon.

Table 2. Final products´ data

Final product	Production rate (ton/h)	Demand (ton)	Allowable lines	Changeover times
P1	2	151	L3	{P6}-1h {P9,P13}-2h
P2	2	105	L1, L5	{P10}-2h {P13,P15}-1h
P3	6	110	L2	{P8,P11}-2h {P12,P15}-1h
P4	4	116	L4	{P7,P11}-1h
P5	7	7	L4	{P7,P11}-1h
P6	12	47	L3	{P1,P13,P14}-1h
P7	6	144	L1, L4	{P4,P5,P10, P12,P13}-1h {P11}-2h
P8	4	80.5	L2	{P3}-2h
P9	3	200.5	L3	{P1}-2h {P6}-1h
P10	5	42.5	L1, L5	{P7}-1h
P11	4	114.5	L2, L4	{P3,P7}-2h {P4,P5,P8}-1h
P12	6	16.5	L2, L4	{P3,P7,P8}-1h
P13	4	170.5	L1, L3	{P1}-2h {P2,P3, P15}-1h
P14	4	103.5	L3, L5	{P6}-1h
P15	6	53	L1, L2	{P3,P7, P11.P13}-2h {P8,P10}-1h

As shown in Fig. 3, the tank capacities are never exceeded. For instance, the capacity of T3 is almost reached when the intermediate product I1 is stored in it. The maximum amount of product I1 stocked in T3 is equal to 139.6 tons. Final products P2 and P3 require intermediate product I1. As seen in the figure, the duration of the storage task of product I1 is determined by the length of the campaign manufacturing final product P2. It is important to note that in order to get the timing of final products P2 and P3 and to not violate the capacity of tank T3, intermediate I1 is produced at a rate (≈ 5 ton/h) that is lower than the maximum permitted (7 ton/h). It can also be observed that the manufacture of final product P3, requiring product I1, starts just at the time the maximum tank capacity is almost reached.

After the storage of I1 is finished, intermediate product I3 is stored in tank T3. The maximum amount of product I3 stocked in this tank is 67.8 tons. In this case, the final product P7, requiring intermediate product I3, determines the length of the I3 product storage task. It is worth to note that the production rate of the final product P6, requiring product I3, is greater than the maximum production rate of intermediate product I3. Thus, at a certain point product I3 is stocked just to have enough material to feed the campaign producing P6.

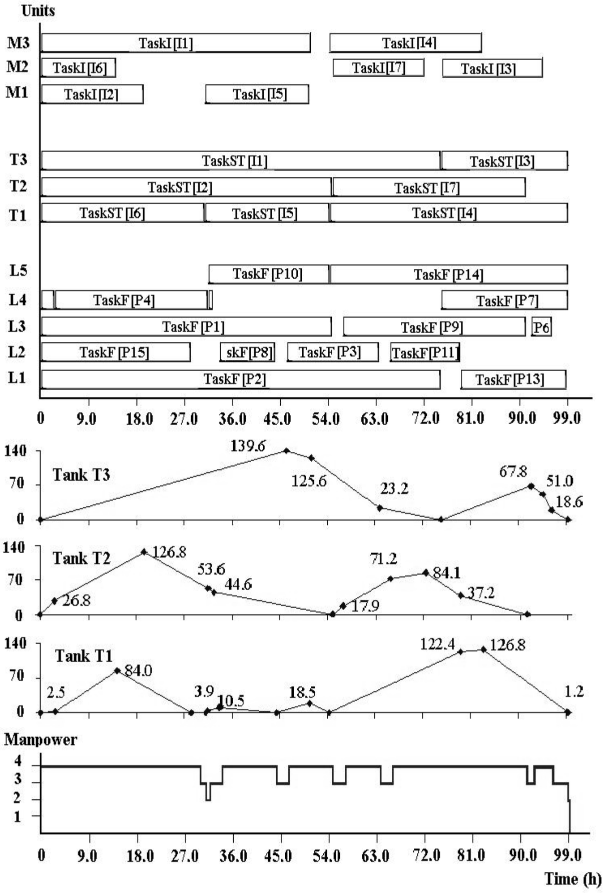

Fig. 3. Diagram showing the optimal solution obtained for problem instance #6.

Changeover times are also shown in Fig. 3. Since P1 and P9 are manufactured in the same line (L3), and the campaign producing P1 is executed before the production run associated to product P9, the campaign which produces P9 must wait two hours to start. Finally, the lower part of Fig. 3 shows that most of the time the four people who are able to work in the packing stage are required. Thus, a few short periods of idleness of one or two workers are registered.

Table 3 presents detailed information about the optimal schedule by including the sequence of products that are manufactured in each unit/line, the campaigns´ agenda as well as the actual intermediate products' manufacturing rate. Furthermore, Table 4 shows computational results for examples #1 to #4 and Table 5 reports the corresponding results for problems #5 to #8. These last examples consider the availability of just four people to work in the last stage, where five are required.

Table 3. Optimal solution of problem instance #6

Unit	Product	Start time (h)	Completion time (h)	Production rate (ton/h)
M1	I1	0	51.1	≈ 5
	I5	31.1	50.8	≈ 8
	I7	55.0	72.4	≈ 9
M2	I3	75.5	94.6	≈ 10
	I6	0	14.4	≈ 13
M3	I2	0	19.5	≈ 12
	I4	54.5	83.5	≈ 9
L1	P2	0	75.5	
	P13	79.1	99.2	
L2	P3	46.4	63.9	
	P8	33.8	44.4	
	P15	0	28.4	
L3	P1	0	55.0	
	P6	92.5	96.4	
	P9	57.0	91.5	
L4	P4	2.7	31.7	
	P5	31.7	32.7	
	P7	75.5	99.5	
	P11	65.9	79.1	
	P12	0	2.7	
L5	P10	31.6	54.5	
	P14	54.5	99.5	

Table 4. Computational results corresponding to problem instances #1 - #4. No manpower requirements are considered

Problem Instance	Constraints	Variables	Objective Function	CPU Time §
# 1	1253	759	69.3	44.2
# 2	1253	759	99.5	127.3
# 3	1253	759	106.1	65.2
# 4	1253	759	112.7	75.4

§ Seconds on a Pentium IV PC (2.8 GHz) with 1 Gb of RAM memory

Table 5. Computational results corresponding to problem instances #5 - #8. Manpower requirements are taken into account

Problem Instance	Constraints	Variables	Objective Function	CPU Time §
# 5	1268	759	69.3	50.6
# 6	1268	759	99.5	28.1
# 7	1268	759	106.1	223.1
# 8	1268	759	128.7*	2000

§ Seconds on a Pentium IV PC (2.8 GHz) with 1 Gb of RAM memory
* Suboptimal solution

It is worth to point out that the number of variables that is reported in Tables 4 and 5 just take into account variables $TaskI_{p'}$, $TaskSt_{p'}$, $TaskF_p$ and $CO_{p'p}$ as well as the binary variables PC_{pq} and FC_{pq}. Assignment variables which are internally handled by the ILOG Scheduler are not included.

An analysis of Tables 4 and 5 reveals that the computational performance is quite dependent on the problem data. As seen, examples with the same number of variables and constraints have associated CPU times that may differ in two orders of magnitude.

V. CONCLUSIONS

A novel CP model addressing the scheduling of a resource-constrained multiproduct plant having continuous stages has been proposed. The model has various relevant features. It can handle many aspects usually found in multiproduct plants, such as finite release times, forbidden campaign-unit assignments, sequence-dependent changeover times, etc.

One key feature of this approach is the handling of limited resources such as storage capacity and manpower. The model has been tested with various case studies and has rendered good computational results for medium-size examples.

ACKNOWLEDGMENTS

Authors acknowledge financial support from ANPCyT under Grant 11-14717, from CONICET under PIP 5915 and “Universidad Nacional del Litoral” under CAI+D 5-30.

REFERENCES

Brailsford, S. C. and C.N. Potts, “Constraint Satisfaction Problems: Algorithms and Applications”, *European J. Operational Research*, **119**, 557-581 (1999).

Ierapetritou, M.G. and C. Floudas, “Effective continuous-time for the short-term scheduling. 2. Continuous and semicontinuous processes”, *Industrial and Engineering Chemistry Research*, **37**, 4360-4374 (1998).

ILOG, OPL Studio 3.7, User’s Manual (2003a).

ILOG, ILOG Scheduler 6.0, User’s Manual (2003b).

Maravelias, C.T and I.E. Grossmann, “A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants”, *Computers & Chem.l Engineering,* **28**, 1921-1949 (2004).

Méndez, C. and J. Cerdá, “An Efficient MILP Continuous-time Formulation for Short-term Scheduling of Multiproduct Continuous Facilities”, *Computers and Chemical Engineering*, **26**, 687-695 (2002).

Schilling, G. and C.C. Pantelides, “A Hybrid Branch and Bound Algorithm for Continuous-time Process Scheduling Formulation”, *AICHE Annual Meeting,* paper no. 171d (1996).

Zhang, X. and R. W. H. Sargent, “The Optimal Operation of Mixed Production Facility – Extensions and Improvements”, *Computers and Chemical Engineering*, **22**, 1287-1295 (1998).

Received: December 20, 2005.

Accepted for publication: June 20, 2006.

Recommended by Editor A. Bandoni.