

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

**MILP Model for Scheduling and Design
of a Special Class of Multipurpose Batch Plants**

V.T. Voudouris & I.E. Grossman

EDRC 06-163-94

**MILP model for Scheduling and Design
of a Special Class of Multipurpose Batch Plants**

Vasilios T. Voudouris * and Ignacio E. Grossmann**

Department of Chemical Engineering

Carnegie Mellon University

Pittsburgh, PA 15213

November, 1993

**** Author to whom correspondence should be addressed**

'Current address: DowElanco, R&D/D2,9410 Zionsville Road, Indianapolis, IN 46268

E-mail: * woudouris@dowelanco.com

****grossmann@cmu.edu**

Abstract.

In this paper we propose a method for the integrated scheduling and design for a special class of multipurpose batch processes. The type of plants considered are the ones where not all the products use the same processing stages, and manufacturing of the products can be characterized through production routes. A novel representation for cyclic schedules is proposed that has the effect of aggregating the number of batches for each product. It is shown that the no-wait characteristics of subtrains can be exploited with a reduction scheme that has the effect of greatly decreasing the dimensionality of the problem. This reduction scheme can be complemented with a tight formulation of the underlying disjunctions in the MILP to reduce the computational expense. The proposed MILP model for scheduling is extended to design problems in which the potential existence of intermediate storage in the production paths is also considered. In addition to the rigorous scheduling of the process, the sizes of the equipment constituting the various production stages are determined. By using exact linearization schemes it is shown that the problem can be reformulated as an MILP model and solved rigorously to global optimality. Application of the proposed model is illustrated with several example problems.

Introduction

Batch processes consist of a collection of processing equipment where batches of the various products are produced by executing a set of processing tasks or operations like reaction, mixing or distillation. Every processing equipment can perform particular operations. Thus, it is possible to recognize production paths consisting of processing equipment which indicate potential routes a batch might follow. A batch of a particular product might follow alternative paths through the process. Processing equipment that can perform the same operations can be grouped in a production stage.

The major classification of batch processes is based on the consideration of the production paths required for the products. If all the products follow the same production path then significant simplifications of the preliminary design problem can be achieved, allowing the creation of a separate class of plants called the multiproduct batch plants (Figure 1). The plants which do not belong to this category are generally classified as multipurpose batch plants. Since the multipurpose class is a superset of the multiproduct class, all the design methods proposed for multipurpose plants are applicable to the multiproduct case. In conjunction to this, the design methods for multipurpose plants are significantly more difficult compared to the methods for multiproduct plants. In this work we introduce a special classification for the multipurpose batch plants. More specifically we divide the multipurpose plants in *sequential plants* and *nonsequential plants* (Figure 1). In a sequential plant it is possible to recognize a specific direction in the plant floor that is followed by the production paths of all the products. Nonsequential plants are all the remaining cases. Clearly, every multiproduct plant is a sequential multipurpose plant, but the converse is not true. Also, according to Reklaitis (1990), multiproduct plants are used when the products exhibit a chemical similarity to each other. As the similarities decrease, the plant becomes a multipurpose batch plant. Among these, sequential multipurpose plants are common in industry and hence of practical importance.

A second major classification of batch processes is due to the transfer policies between the production equipment. At one extreme lie the no-wait plants where no intermediate storage is considered and all intermediates have to be immediately processed by the downstream equipment. At the other extreme lies the case where unlimited intermediate storage is available between every processing equipment that do not belong to the same production stage. In between lies a spectrum of alternative policies (see Ku and Karimi, 1990). A general class under this classification are the plants with mixed intermediate storage (MIS) policy where the process consists of no-wait subtrains separated by an appropriate number of intermediate storage vessels.

Literature review

Because of its relative simplicity, the preliminary design of multiproduct plants has been in focus by many researchers. Rippin (1993) reviews most of the work in this area in recent years. In this review the need for a comprehensive algorithm that will automatically consider and select from all structural possibilities considered simultaneously is recognized. Voudouris and Grossmann (1993) developed a comprehensive MILP model for multiproduct batch plants that considers all the structural possibilities, and even further, considers final product inventories within a periodic scheduling approach.

For multipurpose plants the mixed integer approaches for design and scheduling can be categorized in three broad areas. The main difference between these approaches is the way with which the scheduling subproblem is dealt with. The first approach is based on a simplified campaign planning scheme as for instance with the work by Vaselenak et al (1987). In this approach a central issue is the production campaign formation. Namely, during a production campaign which consists of batches of the same product, two products are allowed to be produced in the same campaign only if their production paths do not share any processing equipment. Faqir and Karimi (1990) generalized this approach by allowing more than one path for the production of a particular product. The model they developed was a nonconvex MINLP which was later reformulated as an MILP by Voudouris and Grossmann (1992). Papageorgaki and Reklaitis (1990) also developed a nonconvex MINLP model which incorporated many additional aspects like flexible task-to-equipment allocation, but still was based on a campaign planning mode. A variant of this campaign approach is proposed by Shah and Pantelides (1992) where the assumption of simultaneously utilizing production paths with noncommon equipment for the formation of production campaigns is applied to production stages instead of production paths. The main problem with these approaches is that the scheduling problem is solved based on a simplifying assumption, thus allowing underutilization of time, the generation of relatively large idle times for the processing equipment, and significant overdesign of the plant when the design subproblem is integrated.

The second approach tries to tackle the problem of time underutilization. For this reason it is recognized that a rigorous scheduling of production paths has first to be performed and to serve as a lower level subproblem to the capacity allocation problem. The work by Wellons and Reklaitis (1991) is representative of this approach. Unfortunately the resulting models are highly intractable mainly because of the nonlinearities that are involved. Furthermore, an arbitrary selection of the total number of batches that are considered may lead to suboptimal solutions.

The third approach is based on solving the scheduling problem by discretizing the time domain in uniform time intervals (Kondili et al, 1993). The major advantage here is the capability of considering complex task networks and handling resource constraints. The major problem with this approach is the large size of the MILP model and the problem of mapping the discretization points with the actual points in time when the events take place. Even though it might seem that the problem can be alleviated by assuming nonuniform time intervals, the identification of these discretization points in the context of the preliminary design is still an unresolved problem. For these reasons approaches based on nonuniform and uniform discretizations are, thus far, considered only for the short term scheduling subproblem and not as a scheduling subproblem inside a larger preliminary design framework.

In this work we address the problem of integrating scheduling in the design of sequential multipurpose plants. As shown in Figure 1 these are plants where all batches follow the same sequence throughout the stages although some of these might be skipped. In the scheduling subproblem an exact model is developed for the sequential batch plant under MIS policy. Starting from the generic machine scheduling formulation, the structure implicit in sequential multipurpose plants is exploited and a reduction scheme is proposed which significantly decreases the dimensionality of the problem. To address the integration of the scheduling subproblem with the design problem, an aggregation scheme is proposed which allows to solve the problem in the space of products, rather than in the space of individual batches. This aggregation scheme is based on a periodic scheduling approach. In this way it is possible to optimize the Production Cycle time, during which the optimal schedule is repeated, and in the design problem, to incorporate costs for final product inventories. The scheduling subproblem is considered for two different cases. In the first case all potential production paths are given and fixed, whereas in the second case the selection of the actual production paths is an optimization variable. The difference between these two cases when the design problem is considered, is the following. In the first case all the equipment in the plant are utilized and the actual decisions are only in terms of sizing the equipment and the scheduling. In the second case the equipment that will actually be used are selected from a set of potential units to synthesize the optimal plant configuration.

Problem definitions.

The general design approach followed in this paper is described in a previous paper (Voudouris and Grossmann, 1993). One of the most important steps in this approach is to identify the space of alternatives (see Figure 2). This space of alternatives

can be redefined if the optimal solution involves undesirable operational conditions. The verification step can be performed using a discrete event simulator. In this section we identify the issues that will define our space of alternatives. We should note that some of the set notation used in this work is not standard mathematical notation, but it has a one to one correspondence with the one in the GAMS modeling language (Brooke et al, 1988).

Consider that a set of products $P=\{p\}$ is given with deterministic demand specifications Q_p that have to be satisfied during a design horizon $3Z$. The production of those products involves the processing of a set of tasks $I=\{i\}$ in a set of processing equipment $K=\{k\}$. Let j be an alias for the index of tasks L . Every product p is associated with a number of processing tasks i . This association is expressed with the set of dyads $A=\{ (p, i) : \text{task } i \text{ is associated with the production of product } p \}$. Every processing equipment can perform only a restricted number of tasks. This is expressed with the set of dyads $B=\{ (k, i) : \text{task } i \text{ can be performed on equipment } k \}$. Because of the above definitions, it is possible to identify a set of *production paths* $H=\{h\}$ in the plant floor (see Figure 3). Every production path is associated with a number of processing equipment which is indicated with the set $M=\{ (h, k) : \text{equipment } k \text{ belongs to path } h \}$. In Figure 3 the set M is defined as, $M = \{ (1,1), (1,4), (2,1), (2,5), (3,2), (3, 4), \dots \}$. Furthermore, every path h will be dedicated to the production of a particular product p which is indicated by the set $C=\{ (h, p) : \forall (k, i) \in B, (p, i) \in A, (h, k) \in M \}$. Again in Figure 3, set C is defined as, $C = \{ (1,A), (2,A), (3,A), (4,A), (5,A), (6,A), (7,B), (8,B) \}$. The specific production recipes for every product are expressed with a particular precedence among the operations. This precedence can be expressed with the set $G=\{ (i, j) \cdot \forall p, (p, i) \in A, (p, j) \in A, (j \text{ has to be executed immediately after } i) \}$ or with the set of triads $D=\{ (h, k, k') : \forall (i, j) \in G, (k, i) \in B, (k, j) \in B, (h, k) \in M, (h, k') \in M \}$. For the example shown in Figure 3 we get $D=\{ (1,1,4), (2,1,5), (3,2,4), (4,2,5), (5,3,4), (6,3,4), (7,4,6), (8,5,6) \}$. Scheduling of the operations consists, in this work, identifying a sequence of operations in the equipment while ensuring that potential clashes will not occur. The potential pairs of clashes are expressed with the set $E=\{ (h, h \setminus k) : (h, k) \in M \text{ and } (h \setminus k) \in M \}$ which for our example is defined as $E=\{ (1,2,1), (2,1,1), (3,4,2), (4,3,2), (1,3,4), (3,1,4), (1,4,4), (4,1,4), (1,7,4) \dots \}$.

The sequential multipurpose plant is a restricted version of the multiple directions plant. More specifically, when the processing equipment are specified to belong to a sequence, then the set D includes elements which have the property that for every path h , k^* has a higher sequence number than k . This can easily be verified for the instance of Figure 3. Graphically this means that it is possible to identify a sequence of the

processing equipment such that all production paths flow in a single direction. This property can also be shown with a Gantt chart as in Figure 4. We have to emphasize that this is a simplified version of multipurpose batch plants but is a more general class than the multiproduct plants. In Figure 4 it can be seen that in the possible sequence given, the operation of product A is after the operation of product B in equipment 1. The reverse is true however for equipment 3. It is therefore not correct to consider a zero processing time of batch C at equipment 2 and treat the plant as multiproduct because this assumption allows only the same sequence in every equipment

Some other interesting characteristics of sequential multipurpose plants is that they can be used as a representation of sequential plants with parallel equipment. Consider for example the plant shown in Figure 3 which involves 3 parallel equipment at stage 1 and 2 parallel equipment in stage 2. By defining the corresponding production paths it is easy to show that the process is a sequential multipurpose batch plant As was mentioned before, the execution of the operations in a production path is performed in a no-wait fashion. This means that a subsequent operation has to be started as soon as the previous operation in a path is finished. Intermediate storage can be treated by the proper definition of the production paths. The existence of an intermediate storage vessel at some point in a production path (e.g. between equipment 2 and 3 in Figure 1.b) can be considered by decomposing the paths in two independent no-wait subpaths (or subtrains) as is shown in Figure 5.

We address the scheduling problem in two phases. First, it is required that a given number of batches for every product np_p has to be produced in the minimum amount of time. As a major second step we present aggregated models for cyclic scheduling in which the number of batches appear as parameters that can be relaxed as variables in design problems. Furthermore, we address the scheduling subproblem for two cases: a) the production paths in the processing network are given and fixed; b) the production paths have to be selected. This hierarchy of models is proposed because there is a trade-off between computational efficiency and generality of the models.

The second part of the paper deals with the optimal design of a process by simultaneously considering the production schedule. In this phase the inputs are the demand specifications for every product during a design horizon. A selection of the sizes and layout of the equipment has to be made in such a manner that the profitability of the process expressed by the Net Present Value (NPV) is maximized. Major assumptions for developing the models include processing times that are independent from batch size, and semicontinuous units are not considered. The batch plant is assumed to consist of no-wait subtrains which are separated by intermediate storage vessels. The location of the

intermediate storage vessels is given, and sizing for these vessels is not considered. During a production cycle a production path is utilized only once to produce the optimal number of batches. Synthesis decisions regarding task to equipment allocation are not considered in this work, although they could have been treated by proper definition of the production paths. However, note that the vessel sizes are considered in standard values. This allows the application of the linearization transformations proposed in our previous work (Voudouris and Grossmann , 1992,1993; Grossmann et al 1992). Next we present the mathematical programming models proposed for the above problems.

Scheduling of sequential multipurpose plants under MIS

Suppose a processing network is given by means of the sets defined above. In general, a production path can be utilized several times to produce identical batches. In this section, however, we assume for simplicity that every batch follows every individual path only once. In this section, the MIS policy is assumed to be a combination of the no-wait and the unlimited intermediate storage policy (UIS), whereas in the later sections it is assumed to be a combination of the no-intermediate storage (NIS) and UIS policies. Consider that t_{hk} represents the starting time of the proper operation of path h performed on equipment k and d_{hk} is the processing time of the same operation. We define the following binary variable,

$$y_{h^i k - j^f k} = \begin{cases} 1 & \text{if path } h^i \text{ is before path } h^f \text{ in machine } k \\ 0 & \text{otherwise} \end{cases}$$

The generic machine scheduling model can be treated with the following well known formulation (e.g. see Balas, 1985a).

$$\begin{aligned} & \min Ms && \text{(P.I)} \\ \text{s.t} & Ms \geq t_{hk} + d_{hk} && \forall (h, k) \in M \\ & t_{hk} \geq 0 && \forall (h, k) \in M \\ & t_{hk} - t_{h'k} \geq d_{hk} - d_{h'k} && \forall (h, k, k') \in D \\ & t_{h'k} - t_{hk} - (d_{hk} - d_{h'k}) y_{hh'k} \geq L_{hh'k} && \forall (h, h' | k) \in E^+ \end{aligned}$$

$$-thk + thk + (d_{hk} \bullet L_{hhk} \wedge y_{hhk} \wedge d_{hk} \quad v_{*h, h', k}) \in E^+$$

$$y_{hh'k} \in f^{0,1*} \quad V(h, h|k) \in E^+$$

where $M_s = \text{Makespan of the schedule}$, $E = E^+ \cup E'$ with $(h, h|k) \in E^+$ if and only if $(h|k) \in E'$. A very simple way to calculate E^+ is to consider only those indices h and h^f which satisfy the condition $\text{ordinality}(h) < \text{ordinality}(h^f)$. Note that $L_{hh'k} = L_{hk} - U_{hk}$ where L_{hk} is the lower bound on the starting time of the operation of path h^f performed on machine k and U_{hk} is the upper bound on the starting time of the operation of path h performed on machine k .

Although a general formulation like the one in (P.I) is relatively simple, it is notoriously difficult to solve (see also Raman and Grossmann, 1992). Significant improvements can be obtained, however, by exploiting the structure of a particular problem. In our case we start by exploiting the fact that our network consists of no-wait subtrains. In other words, the third constraint in (P.I) is defined as an equality. Furthermore, we exploit the fact that the plant is sequential which means that we can identify an equipment which we can characterize as being first in the sequence. Defining the starting times \bar{t}_h for every path in the first equipment, the starting times thk at each machine k are given by,

$$thk = \bar{t}_h + \sum_{k'=1}^{k-1} d_{hk'} \quad V(h, k) \in M \quad (1)$$

Note that the sum is defined over all the equipment. In case an equipment does not belong to path h then d_{hk} is zero. Note that equation (1) is equivalent to the third constraint in (P.I) when it is an equality, and thus this constraint is replaced by (1).

The fixed paths case

Consider that all production paths in a process have been identified and all of them will be used in the schedule. The existence of intermediate storage vessels as well as parallel equipment in a production stage is treated by the path decomposition method illustrated in Figure 5. By assuming no-wait transfer, i.e. the third constraint in (P.I) is an equality, we can substitute the starting times thk in (P-I) with the definition in (1). Thus, we get the following model,

$$\begin{aligned}
& \min Ms && \text{(P.2)} \\
\text{S.L.} & MS \leq \bar{t}_h + \sum_{k=1}^{Iid} d_{hk} && \forall h \\
& \bar{t}_h \geq 0 && \forall h \\
& \bar{t}_{h'} - \bar{t}_h + W(1 - y_{hh'k}) \geq \left(\sum_{k'=1}^k d_{hk'} - \sum_{k'=1}^{k-1} d_{h'k'} \right) && \forall (h, h', k) \in E^+ \\
& \bar{t}_h - \bar{t}_{h'} + W y_{hh'k} \geq \left(\sum_{k'=1}^k d_{hk'} - \sum_{k'=1}^{k-1} d_{h'k'} \right) && \forall (h, h' \setminus k) \in E^+ \\
& y_{hh'k} \in \{0, 1\} && \forall (h, h' \setminus k) \in E^+
\end{aligned}$$

Note that the calculation of bounds for the disjunctions is not required here due to the no-wait transfer assumption. Also, note that it is possible to consider the existence of intermediate storage vessels by a proper definition of the paths. The production paths have to be classified into two classes. First there are the paths that are producing intermediate products called the *intermediate paths*. Then there are the paths that produce final products called *market paths*. The set $FT = \{h\}$ is a subset of H and has as entries all the market paths in the process. For every path producing an intermediate, there is exactly one path (either market or intermediate path) that is considered to be the downstream path after the intermediate storage vessel. The correspondence between upstream and downstream paths is indicated with the set of dyads $F = \{(h, h^f) : \text{path } h \text{ is the upstream path for the downstream path } h^f\}$. Note that for every market path there can be one or more intermediate paths related to it. One of these is the immediate predecessor of the market path. The following constraint ensures that the downstream path starts operating after the upstream path has produced a batch,

$$\bar{t}_h + \sum_{k^f=1}^{IK} d_{h^f k^f} \leq \bar{t}_{h^f} \quad \forall (h, h^f) \in F \quad \text{(Ia)}$$

Although models (P.2) and (P.I) are mathematically equivalent under the no-wait transfer assumption, model (P.2) has significantly smaller number of continuous variables, the same number of binary variables and a relatively smaller number of

constraints. It is thus a formulation of model (P.1) in a reduced continuous space. Since the lower bounds in model (P.1) are implicitly considered in model (P.2), the LP relaxation of model (P.2) is the same as model (P.1) when equality of the third constraint is enforced. This means that model (P.2) requires less computational effort to solve a particular instance of the same problem. The main problem, however, that model (P.1) exhibits, is also present in model (P.2). More specifically, it has been proved (Balas, 1985a) that the disjunctive constraints defined over the set E , do not have any constraining power when the corresponding binary variable is relaxed. In order to alleviate this problem many researchers have dedicated significant efforts to devise strong cutting planes for particular cases of model (P.1). Since model (P.2) is a particular case of model (P.1) which is obtained by applying well defined mathematical steps, it is possible to modify some of the most efficient of these cutting planes and to apply them in model (P.2). Some efficient cutting planes have been proposed by Balas (1985b). From our experience the most effective cutting planes have been initially proposed by Dyer and Wolsey (1990) for the one machine scheduling problem with release times and due dates, and later modified for model (P.1) by Applegate and Cook (1991). The form of these cutting planes for model (P.2) is given in Appendix I. By also exploiting the no-wait character of model (P.2), we propose in the next section a reduction scheme which leads to an equivalent model (P.3) that has a significantly lower dimensionality in the binary space.

Reduction scheme

The dimensionality of the binary space in model (P.2) is equivalent to the cardinality of set E^+ , since the binary variable y_{hh^*} is defined over that set. A geometric interpretation of the principles where the reduction scheme is based, are given in the Gantt chart of Figure 6. In the first case of this figure, $y_{AB1} = 1$ and $y_{AB2} = 1$ are implied when $y_{AB3} = 1$ and thus they are redundant. In the second case however the instance $y_{AC3} = 1$ is not redundant when $y_{AC1} = 1$ because, as shown in the third case, it is possible to get a 'reversal' where $y_{AC1} = 1$ and $y_{AC3} = 0$ (or $y_{AC3} = 1$). Based on these ideas we devise the following reduction scheme which significantly reduces the cardinality of the domain of the sequencing binary variables without compromising the optimality of the model.

Let us first define the variables D_{hh^*} , V_{hh^*} and S_{khh^*} with the following equations,

$$D_{hh^*} = \sum_{k'=1}^{k-1} 4IY - \sum_{k'=1}^k dh_{kk^*} \quad \forall (h, h', k) \in E^+ \quad (2)$$

$$V_{hh'} = \min_{k: (h, h') \in E^*} [D_{hh'} + O] \quad \forall h, h' \quad (3)$$

$$Slk_{hh'k} = D_{hh'} - V_{hh'} \quad \forall (h, h') \in E^+ \quad (4)$$

The slacks $Slk_{hh'k}$ represent the forced idle time imposed on machine k when path h is followed by h' in a no-wait multipurpose plant. A geometrical interpretation can be seen in the Gantt chart of Figure 7. In this figure only two pairs of paths are given and the corresponding slacks are shown in Table I.

We further define $Pl_{hh'k}$ and $Ch_{hh'k}$ as follows following equations,

$$Pl_{hh'k} = d_{hk} + d_{h'k} + Slk_{hh'k} \quad \forall (h, h') \in E^+ \quad (5)$$

$$Ch_{hh'k} = \max[0, Slk_{hh'k} - Pl_{hh'k}] \quad \forall (h, h') \in E^+ \text{ and } (h, h', k) \in E^+ \quad (6)$$

or

$$Ch_{hh'k} = \begin{cases} \text{small positive number} & \text{if } Slk_{hh'k} = Pl_{hh'k} \\ 0 & \text{if } Slk_{hh'k} < Pl_{hh'k} \end{cases} \quad \forall (h, h') \in E^+ \text{ and } (h, h', k) \in E^+ \quad (7)$$

Finally, the sets Q and R are defined as follows,

$$Q = \left\{ (h, h') : (h, h') \in E^+ \text{ and } \sum_{k=1}^{|K|} Ch_{hh'k} > 0 \right\} \quad (8)$$

$$R = \left\{ (h, h') : (h, h', k) \in E^+ \text{ and } (Slk_{hh'k} = 0 \vee Slk_{hh'k} = 0) \right. \\ \left. \vee (h, h') \in Q \vee (h, h', k) \in Q \right\} \quad (9)$$

Note that all triads belonging to set R belong to set E^+ , but they have to also satisfy one of the four conditions in (9). Thus depending on these conditions, the cardinality of set R is smaller or at most equal to the cardinality of the set E^+ . Next we prove the following proposition.

Proposition 1. Problem (P.2) is equivalent to a reduced problem (RP.2), in which the disjunctions are defined only over the set R rather than set E^+ ,

Proof. See Appendix n.

Due to the restrictive nature of the conditions imposed when set R is defined, the number of binary variables and disjunctive constraints is significantly reduced. The effectiveness of this reduction scheme and of the cutting planes is illustrated in the next example.

Example 1

Consider the case where 4 products will be produced in a multipurpose batch plant. For product A three batches will be produced, for product B also three and for products C and D two batches for each. In Figure 8 the production paths are shown for the batches of each product. The objective is to find the schedule that minimizes the makespan. Although this problem with 4 products and 10 batches, seems to be relatively small, the computational effort required to solve this instance is surprisingly large. The optimal schedule is shown in the Gantt chart of Figure 9. The optimal makespan is 52 hours.

In order to study the impact of the reduction scheme and the cutting planes, 4 particular models have been tested. All the models were generated with GAMS 2.25 (Brooke et al, 1988) and the MILP solver was OSL (OSL, 1991). The computer platform was an IBM/R6000/Power 530 workstation. Note that a custom fit options file was used to optimally set the optimization parameters on OSL. More details on the settings of these parameters are given in the section of computational considerations. In the first version model (P.2) was used with no cutting planes and no reduction scheme. The model involved 182 constraints and 97 variables of which 86 were binary. After more than 2 CPU hours and 144,063 nodes enumerated in the branch and bound tree, the solution had still a relaxation gap of 47%. This solution was 53 hours which is not the optimal solution. In the second version the same model was used but now only the reduction scheme was applied. The model involved 120 constraints and 66 variables of which 55 were binary. Again in this case the model failed to report the optimal solution in 2 CPU hours. Instead a solution of 53 was the best reported. The relaxation gap was 20.9% and 137,532 nodes were enumerated. In the third case only the cutting planes were used. The optimal solution of 52 hours was found after 42 CPU minutes and 51,232 nodes. The LP relaxation of this model was 52 hours so the relaxation gap was 0% and the tree was enumerated to identify an integer solution with the same makespan as the relaxation. It should be noted that in the first two versions the relaxation had a solution of 18 and thus the relaxation gap was 65%. Finally, in the fourth case both the cutting planes and the reduction scheme were used. This model involved 130 constraints and 66 variables of which 55 were binary. The optimal solution was now obtained in 601 CPU seconds and after 16,092 nodes. Note that the reduction scheme reduces the number of binary

variables from 86 to 55. For the final version we also solved the model with GAMS 2.25/Sciconic 2.11 and the solution was obtained in 293 CPU seconds and 13,177 nodes were enumerated. Sciconic (SCICONIC, 1991) however did not always perform better than OSL.

To also clarify the difference between the multiproduct and sequential multipurpose plants the same example was solved as if it were a multiproduct plant with zero processing time on the stages that are skipped. The optimal solution in this case was 71 hours which is 40% higher than the optimal makespan! This schedule is shown in Figure 10. The model used for the rigorous scheduling was the one proposed by Birewar and Grossmann (1989).

From the computational results that are reported here it is apparent that the combination of the cutting planes and the reduction scheme improves significantly the computational performance of model (P.2). These methods however are not sufficient to address the intractability of the model. For this reason we propose in the next section an aggregation scheme in which the objective is the minimization of the cycle time.

Aggregated model

In this section we further exploit a significant characteristic of multipurpose plants to reduce the computational demands and thus make possible the solution of larger problem instances before the computational "exponential wall*" is reached. One of the main problems in the reduced model (RP.2) is the fact that when the number of paths (or total number of batches) is large, the number of disjunctive constraints is increased quadratically to the number of batches. The main idea in the aggregated model is to employ a periodic scheduling approach in which a smaller nested scheduling subproblem is solved optimally. It is assumed that this elementary scheduling subproblem (production wheel) is formed with single product campaigns and that the actual schedule is obtained by repeating the elementary schedule a number of times. This number of repetitions has to be determined optimally. This approach has successfully been implemented in the case of multiproduct batch plants (see Voudouris and Grossmann, 1993). The key decisions in the elementary schedule is the optimal sequence of the products, the number of batches produced and the length of the elementary schedule. For the overall schedule it will be decided how many times the elementary schedule will be repeated. As an example, consider the scheduling problem given in Figure 11. A total of 12 batches will be produced where 3 batches are of product A, 6 of B and 3 of C. One possible realization of the periodic scheduling approach is to repeat three times an elementary schedule which introduces the production of 1 batch each of A and C and 2 batches of B. A restriction of

the above approach would seem to be that the total number of batches will be a multiple of the number of repetitions. This restriction is, however, not so important because the total number of batches considered in the schedule will be the next higher multiple of the number of repetitions compared to the total number of batches required. In other words, we assume the schedules to be periodic even if this is not always optimal. Also, we will allow for possible overproduction of batches to introduce more freedom in the selection of a periodic schedule.

Timing of the elementary schedule is based on the recognition of the bottleneck stage. The notion of a bottleneck stage is well understood in the case where batches of the same product are considered. In this case the stage with the largest processing time is considered to be the bottleneck stage. The processing time of the bottleneck stage represents the period of repetition. This period is widely known as the *cycle time* of that product. When a larger number of products with various production paths are involved, the timing pattern in every stage becomes relatively complicated. It is possible, however, to recognize a bottleneck stage whose operation defines the period of repetition for the whole elementary schedule. This period of repetition will be denoted in this paper as the *production cycle time*.

Given that at least np_p batches have to be produced, consider nb_h batches that are produced in path h for product p $((h, p) \in C)$ in each cycle. In any particular equipment k the starting time of the corresponding operation of path h is defined as t_{hk} and the finish time is defined as f_{hk} . As shown in Appendix III, the finish time f_{hk} is given by the following equation,

$$f_{hk} = t_{hk} + d_{hk} + (nb_h - 1)Tl_h \quad \forall (h, k) \in M \quad (10)$$

where Tl_h is the cycle time of path h and is equal to the processing time of the operation in the path with the largest duration,

$$Tl_h = \max_{k: (h, k) \in M} \{d_{hk}\} \quad \forall h \quad (11)$$

Since an explicit expression of the finishing time of single product campaign within an elementary schedule is given with equation (10), the disjunctions in model (P.2) can be written to arbitrate clashes among campaigns. The modified disjunctive constraints are,

$$\bar{t}_h + (nb_h - 1) T_h + \sum_{k'=1}^k dhk' \leq \bar{t}_h + \sum_{k'=1}^k d_h Y + W d - y t o) \quad V(h, h', k) \in R \quad (12)$$

$$\bar{t}_h + (nb_h - 1) T_h + \sum_{k'=1}^k d_{hv} \leq \bar{t}_h + \sum_{k'=1}^{k-1} dhk' + W y^{\wedge} \quad V(h, h \setminus k) \in R \quad (13)$$

Note that the disjunctions are again defined over the set R. This is because the reduction scheme is still valid with the aggregated disjunctive constraints.

Proposition 2 : The production cycle (or cycle time of the elementary schedule) can be defined rigorously by the following equation,

$$P = \max_{(h, h \setminus k) \in R} \left[\bar{t}_h + (nb_h - 1) T_h + \sum_{k'=1}^k d^{\wedge} - (\bar{f}c + \sum_{k'=1}^k <W) \right] \quad (14)$$

Proof: See appendix in.

In terms of inequalities in a cycle time minimization problem (14) is given by,

$$P \geq \bar{t}_h + (nb_h - 1) T_h + \sum_{k'=1}^k d_{hv} \leq (\bar{t}_h + \sum_{k'=1}^{k-1} d_{hv}) \quad V(h, h \setminus k) \in R \quad (15)$$

As mentioned in the multiproduct case (Voudouris and Grossmann, 1993), when a periodic scheduling approach is considered, and particularly for cases of small number of elementary schedule repetitions, it is imperative to devise constraints that ensure the integrality of the ratio,

$$\sum_{i=1}^I \xi_i = Nr \quad (15a)$$

where T_c = Total time required.
 Nr = Number of elementary schedule repetitions.
 P = Production cycle time.

Consider the following binary variable defined over the set $SV = \{sv\}$ whose entries represent the number of repetitions of the elementary schedule,

$$r_{sv} = \begin{cases} 1 & \text{if } sv \text{ repetitions of the elementary schedule are considered} \\ 0 & \text{otherwise} \end{cases}$$

Then the following constraints ensure the integrality of the ratio T_c / P .

$$\sum_{sv=1}^{|sv|} r_{sv} = 1 \tag{16}$$

$$\sum_{sv=1}^{|sv|} a_{p_{sv}} = P \tag{17}$$

$$a_{p_{sv}} \leq U r_{sv} \quad \forall sv \tag{18}$$

$$\sum_{sv=1}^{|sv|} \text{ord}(sv) a_{p_{sv}} = T_c \tag{19}$$

The total number of batches produced by the market paths has to be greater or equal to the required number of batches for each product. This is expressed by the constraint,

$$\sum_{h: (h, p) \in C, h \in H^*} n_{bh} \geq n_{pp} \quad \forall p \tag{20}$$

where the inequality sign has been specified to allow an overproduction of batches for a periodic schedule as discussed previously. When strict equality is enforced, the optimal solution will be equal or worse to the optimal solution obtained when only the inequality is considered as will be illustrated in example 2. The nonlinear constraint in (20) can be linearized using a case 2 linearization scheme (Grossmann et al, 1992). The equivalent linear set of constraints is,

$$\sum_{h: (h, p) \in C, h \in H^*} \sum_{sv=1}^{|sv|} \text{ord}(sv) a_{bhsv} \geq n_{pp} \quad \forall p \tag{21}$$

$$\sum_h a_{bhsv} \leq U r_{sv} \quad \forall sv \tag{22}$$

$$\sum_{sv} amb_{hsv} = nbh \quad \forall h \quad (23)$$

The following constraints have to be satisfied in order to ensure proper operation of the intermediate storage vessels,

$$\bar{t}_h + \sum_{k'=1}^K dhk' \leq \bar{t}_{h'} \quad \forall (h, h') \in F \quad (24)$$

$$nbh \leq Zhh \leq nbh' \quad \forall (h, h') \in F \quad (25)$$

Constraint (25) ensures the proper time coordination between the upstream and downstream paths. Zhh^f is a fixed rational number whose value is indicated by the ratio of the cycle times of the upstream and downstream subtrains. This ratio is such that the productivities of the two subtrains are equal (see Karimi and Reklaitis, 1985; Modi and Karimi, 1989). When fixing the value of Zhh^f care should be given to the fact that even though the ratio of the cycle times is unrestricted, the ratio of the number of batches is a ratio of integral numbers.

The final MELP model is given by,

$$\min Tc \quad (P.3)$$

$$\text{s.t.} \quad (12), (13), (15), (16)-(19), (21) - (25)$$

$$\bar{t}_h \geq 0 \quad \forall h$$

$$P \geq 0$$

$$ap_{sv} \geq 0 \quad sv = 1 \dots |SV|$$

$$amb_{fev} \geq 0 \quad \forall h, sv = 1 \dots |SV|$$

$$nbh = \text{integer} \quad \forall h$$

$$y_{hhk}, r_{sv} \in \{0,1\}$$

Example 2

In example 2 we resolve the problem described in example 1 but this time we use the periodic scheduling model (P.3) which involves 301 constraints and 322 variables out of which 12 are discrete. The optimal schedule obtained from model (P.3) is shown in the Gantt chart of Figure 12. The elementary schedule involves one batch of every product. Since model (P.3) was used, the objective was to minimize the cycle time of the schedule and not the makespan as was the case before. The cycle time of the elementary schedule (Production Cycle) is 16 hours and it has to be repeated three times. The total cycle time required is 48 hours and the makespan is 58 hours. In comparison, model (P.2) yields an optimum makespan of 52 hours. Note also that 3 batches of each product will be produced. Hence equations (21) will not all be active since there is an overproduction of one batch of C and one of D. Figure 13 shows the optimal schedule with only one repetition of an elementary schedule involving 3 batches of A, 3 of B, 2 of C and 2 of D. In other words we enforce strict equality on equation (21). In this case the optimal value for the total cycle time is 61 hours. Thus the schedule of Figure 12, even though it involves more batches than required, is more efficient compared to the one in Figure 13. Model (P.3) was solved in 24 CPU seconds using GAMS 2.25/OSL on an IBM/R6000/Power 530. A total of 305 nodes were enumerated in the branch and bound tree. Note that Special Ordered Sets (see Voudouris and Grossmann, 1992) are not considered as discrete variables although they are present. The set of repetitions SV had 10 entries starting from repetition 1 (see tree partitioning scheme on the computational considerations section).

The path selection problem

Many times in a multipurpose plant it is possible to identify more than one production path that a batch of a particular product can follow. The demand for a particular number of batches can then be satisfied from batches produced in every individual path. These paths may produce batches of the same or different size and of the same or different number. As an example, if in an otherwise single-equipment-per-stage subtrain, only one stage has two equipment operating in parallel, then it is possible to identify two production routes dedicated for the product produced in the subtrain. The ability to consider different batch sizes per route allows the consideration of equipment of unequal sizes in a stage operating in parallel. This can have a significant effect in the throughput of the process. Furthermore, the fact that the time relation of the paths is not restricted, allows to consider both in-phase and out-of-phase cases resulting in additional throughput

The models developed thus far in this paper have considered that all identifiable paths for the production of batches of a particular product will actually be used. It is conceivable, however, that in some cases the forced utilization of a production path is unnecessary. Therefore, an important extension of the basic model must consider the nonexistence of a path.

In mathematical terms the nonexistence of a path can be considered in various ways. The main idea of these methods is to make the disjunctive constraints (12) and (13) redundant for the cases where one path in this disjunction is nonexistent. Furthermore, the number of batches produced in such a path has to be forced to zero.

Consider the following binary variables,

$$y_{3h} = \begin{cases} 1 & \text{if path } h \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

Note that two paths h and h' belonging to two different products, may otherwise be exactly the same in terms of direction inside the plant. Since some of the processing parameters associated with these two paths like processing times, may be different, a distinction of these paths has to be made.

One possible way to consider the nonexistence of a path is the following,

$$\bar{t}_h + (nbh - 1) T_{1h} + \sum_{k'=1}^k d_{hk'} \wedge \bar{V} + \sum_{k'=1}^{k-1} d_{hk'} + W(3 - y_{hh} - y_{3h} - y_{3h'}) \quad (26)$$

$$\forall (h, h' \in R)$$

$$\bar{t}_h + (nbh - 1) T_{1h} + \sum_{k'=1}^k d_{hk'} \wedge \bar{t}_h + \sum_{k'=1}^{k-1} d_{hk'} + W(2 + y_{hh} - y_{3h} - y_{3h'}) \quad (27)$$

$$\forall (h, h' \in G \cup R)$$

$$nbh \leq U y_{3h} \quad \forall h \quad (28)$$

Note that there is no need to include any logical constraints to enforce consistency between the binary variables. This is because the only way that both constraints (26) and (27) are non redundant is for the variables y_{3h} and $y_{3h'}$ to be one. In this case, depending on the value of the binary variable y_{hh} only one of these constraints will be nonredundant.

Another alternative to model the above is by introducing the following binary variables,

$$y^{1hhk} = \begin{cases} -1 & \text{if } P^{ah} \text{ before path } h^f \text{ in machine } k \\ 1 & \text{otherwise} \end{cases}$$

which is defined over the set R , and

$$y^{2hhk} = \begin{cases} -1 & \text{if path } h' \text{ is before path } h \text{ in machine } k \\ 1 & \text{otherwise} \end{cases}$$

which is again defined over the set R . Note that because of the difference in defining the variables, when both of them are not zero the following condition must hold,

$$y^{1hhk} + y^{2hhk} = 1 \quad \forall (h, h', k) \in R$$

Therefore, the following constraints can replace constraints (12) and (13),

$$\bar{t}_h + (nb_h - 1) T_h + \sum_{k'=i}^k d_{hk} \wedge \bar{t}_{h'} + \sum_{k'=i}^{k-1} 4^* + W(1 - y^{1hhk}) \quad \forall (h, h', k) \in R \quad (29)$$

$$\bar{t}_h + (nb_h - 1) T_h + \sum_{k'=1}^k \wedge \bar{t}_{h'} + \sum_{k'=1}^{k-1} d_{hk} V + W(1 - y^{2hhk}) \quad \forall (h, h', k) \in R \quad (30)$$

The logical consistency between the binary variables y^{1hhk} , y^{2hhk} and y^3h is enforced with the following constraints (in aggregated form),

$$\sum_{(h',k): (h,h',k) \in E^*} (y^{1hhk} + y^{2hhk}) \wedge y^3h \quad \forall h \quad (31)$$

$$\sum_{(hJ): (hJi|k) \in E^*} (y^{1hhk} + y^{2hhk}) \leq y^3h \quad \forall h^f \quad (32)$$

$$y^{1hhk} + y^{2hhk} \wedge y^3h + y^3h - 1 \quad \forall (h, h', k) \in R \quad (33)$$

The sets of constraints (29)-(33) and (26)-(27) are equivalent to each other in the binary space which means that they are equivalent as far as representation of the problem is concerned. They exhibit, however, significant differences related to computational performance. The advantage of constraint set (26)-(27) is that a smaller number of binary variables and constraints is required. The main disadvantage, however, is that the upper

bound in the disjunctive constraints is 3 times larger compared to the upper bound in constraints (29)-(33), This means that the constraining power of these constraints is significantly reduced when the binary variable is relaxed. This might lead to a larger enumeration of the branch and bound tree (Nemhauser and Wolsey, 1988). The above effect however, is mostly significant when constraints (26)-(27) are a dominant part of the MILP model. In our case this is not the case and the intractability of (26)-(27) is reduced. From our experience the above argument has been justified and constraints (26)-(27) have been chosen. By replacing the disjunctive constraints (12) - (13) in model (P.3) with constraints (26M27) and by adding constraint (28), the following model is obtained,

$$\begin{array}{ll}
 \min T_c & \text{(P.4)} \\
 \text{s.t} & (15),(16)-(19), (21).(28) \\
 & \bar{t}_h \leq 0 \quad \forall h \\
 & P \geq 0 \\
 & a_{p_{sv}} \geq 0 \quad sv = 1 \dots |SV| \\
 & a_{nb_{hs}} \leq 0 \quad \forall h, sv=1 \dots |SV| \\
 & nbh = \text{integer} \quad \forall h
 \end{array}$$

This model addresses the problem of path selection and scheduling of the selected paths in a sequential multipurpose plant, and its application is illustrated later in this paper.

The design problem

In this section we will expand the scheduling models to design by considering the selection of sizes of the various equipment Furthermore, when the underlying scheduling subproblem is the path selection case, this also involves the selection of units.

The number of batches of each path that will be produced during the total design horizon is the product of the number of batches produced during a production cycle multiplied by the number of repetitions. Thus,

$$N_r n_{b_h} = \frac{q_h}{B_h} \quad \forall h \quad (34)$$

where $q^{\wedge} =$ demand for product p satisfied by path h (Note that $(h, p) \in C$).

$B_h =$ Batch size of path h .

Since every vessel must be able to accommodate the batches of the paths that are utilizing that vessel,

$$V_k \geq S_{hk} B_h \quad \forall (h, k) \in M \quad (35)$$

In this problem the total number of batches for each product n_{pp} is not given. Instead the total demands Q_p that have to be satisfied during the design horizon are the input data. For this reason constraint (20) has to be replaced with the following constraint,

$$N_r \sum_{(h,p) \in C, h \in H} q_h \geq Q_p \quad \forall p \quad (36)$$

Because of the periodic scheduling that is assumed, it is possible to incorporate the inventory and operating costs in the design model. The objective in this case is to maximize the profitability of the process as expressed by the Net Present Value. This is defined with the following equation,

$$NPV = -P_c + (R - O_c) (1-tx) (prcoef) + (P_c/N_y) tx (prcoef) \quad (37)$$

where rx is the tax rate, N_y the expected life of the plant in years, R is the total revenue from selling the products which is calculated only for the required amount of products and not for the overproduction. $Prcoef$ is the present value coefficient with which future profits are projected to the present. This coefficient is given by,

$$prcoef = \left\{ \frac{(1+in)^{N_y} - 1}{in (1+in)^{N_y}} \right\} \quad (38)$$

where in represents the interest rate.

The plant cost P_c can be calculated by the following equation,

$$P_c = \sum_k \alpha_k V_k^{\beta_k} \quad (39)$$

which is the capital investment required for the equipment
The cyclic operating costs O_c are calculated by the expression,

$$O_c = \sum_p \left(\mu_p \frac{Q_p}{2} P \right) + \text{mint} N_r \quad (40)$$

where the first summation is the inventory cost and the second term is the setup cost paid every time the optimal schedule is repeated. N_r is the total number of repetitions of cycles, mint is the cost in \$ per repetition and M_p is the inventory cost per unit mass of inventory of product p per unit time. Note that the calculation of the inventory cost is different to the one proposed for the case of multiproduct plants (Voudouris and Grossmann, 1993). This operating policy is indicated in Figure 14. The main reason behind this assumption is the fact that in multipurpose plants it is relatively difficult to identify production times for each of the products. Even further, the consideration of production times generates nonlinear terms that cannot be linearized.

The consideration of intermediate storage can be performed in a similar fashion as in the pure scheduling and operation subproblem. The main difference, however, is **that** constraint (25) has to be replaced by the following constraint,

$$\frac{B_h}{T_h} = \frac{B_{h'}}{T_{h'}} \quad \forall (h, h') \in F \quad (25a)$$

As mentioned before it is possible to consider two distinct cases. In the first case only the simultaneous capacity allocation and scheduling of an existing process with selected production paths, is considered. The nonlinear model for this case is,

$$\begin{aligned} & \max \text{ NPV} && \text{(R5)} \\ \text{s. t.} & && (12M13), (15), (15a), (24), (25a), (34)-(37), (39)-(40) \\ & && \text{Non negativity and integrality constraints} \end{aligned}$$

The second case addresses the potential existence of paths and units in addition to the items in (P.5). This gives rise to a model that partially addresses the issue of flowsheet synthesis. Again the nonlinear model is,

$$\begin{aligned} & \max \text{ NPV} && \text{(P.6)} \\ \text{s.t} & && (15), (15a), (24), (25a), (26)-(28), (34)-(37), (39)-(40) \\ & && \text{Non negativity and integrality constraints} \end{aligned}$$

Linearization of nonlinear design models.

By considering the availability of equipment in standard sizes, it is possible to linearize models (P.5) and (P.6) from nonconvex MINLPs to MttP's and therefore define a model for which the global optimum can be rigorously obtained with the LP based branch and bound method. First the following binary variable is defined,

$$x_{ks} = \begin{cases} 1 & \text{if equipment } k \text{ has size } s \\ 0 & \text{otherwise} \end{cases}$$

Note that the sizes of the equipment are indicated by the set $S^k = \{s\}$. The first entry in that set corresponds to zero size or nonselection of that equipment. We therefore define the modified set $S_k = S^k \setminus \{1\}$ which indicates only nonzero sizes. The sets S_k and S^k are associated with the discrete sizes $V_{S^k} = \{V_{k1}, V_{k2}, \dots, V_{ks}\}$. Thus the size of a vessel is given by,

$$V_k = \sum_{s \in S^k} v_{ks} x_{ks} \quad (41)$$

$$\sum_{s \in S_k} x_{ks} = 1 \quad (42)$$

for the case of selected paths. For the path selection cases the sum in (42) is over set S'_k instead of set S_k for equation (42).

By combining constraints (34) and (35) we get,

$$N r_{nb_h} \geq \frac{S_{hk} q_h}{V_k} \quad \forall (h, k) \in M \quad (43)$$

In order to consider the availability of intermediate storage the following condition must hold,

$$q_h = q_{h'} \quad \forall (h, h') \in F \quad (43b)$$

Because of the multiple choice character of constraint (42), constraint (41) can be written as,

$$\frac{1}{V_k} = \sum_{s \in S_k} \frac{x_{ks}}{v_{ks}} \quad (44)$$

By replacing this into (43) we get,

$$N r n b_h \geq \sum_{s_k} \frac{S_{hk} q_h}{V_{ks}} x_{ks} \quad V(h, k) \in M \quad (45)$$

In the left hand side of this constraint the bilinear term has already been linearized by replacing (20) with (21)-(23). The right hand side has also the bilinear terms $q_h x^h$. By applying a case 2 linearization scheme we get,

$$\sum_{sv \ll 1} \frac{S_{hks} q_h}{V_{ks}} \geq \sum_{s_k} \frac{S_{hks} q_h}{V_{ks}} \quad V(h, k) \in M \quad (46)$$

$$2 \sum_{h} a_{hsv} \wedge U r_{sv} \quad V_{sv} \quad (47)$$

$$2) \sum_{sv} a_{hsv} = n b_h \quad V_h \quad (48)$$

$$s_k \leq U x_{ks} \quad V_{k.s} \quad (49)$$

$$\sum_{s_k} a_{hks} = q_h \quad V(h, k) \in M \quad (50)$$

Similarly, constraint (36) can be linearized and replaced with the following constraints,

$$\sum_{sv} a_{hsv} = Q_p \quad V_p \quad (51)$$

$$\sum_{h: (h,p) \in C} a_{hsv} \wedge Q_p r_{sv} \quad V_{p,sv} \quad (52)$$

$$\sum_{sv} a_{hsv} = q_h \quad V_h \in H \quad (53)$$

In order to consider the proper operation of intermediate storage (43b) has to be combined with the following constraint,

$$n b_h = \frac{T_{h'}}{T_{in}} n b_{h'} \quad V(h, h') \in F \quad (25b)$$

It can easily be proved that (25b) and (43b) are equivalent to (25a) by considering that $q_h = B_h n_{bh}$. Because of the consideration of standard sizes the nonconvex equation (39) can be written as,

$$Pc = \sum_k \sum_{S_k} cst_{ks} x_{ks} \quad (54)$$

where the parameter $cst_{ks} = o_{k} v_j \xi$ is the cost of every equipment k . Finally equation (19) has to be rewritten in terms of the design horizon which is a parameter instead of the total production time required which is a variable,

$$\sum_{sv=1}^{|SV|} ord(sv) ap^{\wedge} = H \quad (19a)$$

For the case of fixed paths the design and scheduling model is,

$$\max NPV \quad (P.7)$$

s. L (12)-(13), (15), (16)-(18), (19a), (24), (25b), (40), (42), (46)-(53), (54)

Non negativity and integrality constraints

For the path selection problem the synthesis, design and scheduling model is,

$$\max NPV \quad (P.8)$$

s. t. (15), (16)-(18), (19a), (24), (25b), (26)-(28), (40), (42), (46)-(53), (54)

Non negativity and integrality constraints.

Computational considerations

As mentioned earlier in the paper, the disjunctive constraints involved in scheduling problems are notorious for the computational difficulties they add to a MHP model. For this reason a number of cutting planes have been proposed in the literature which alleviate this problem. This is achieved by improving the relaxation gap between

the LP subproblems that are solved in the branch and bound tree and the integer solutions of the problem. More details about these cutting planes are given in Appendix I.

Significant differences in the computational performance of the models can be achieved by modifying the default options the OSL solver is using (OSL, 1992). These options have to be properly adjusted for every particular model in order to get the maximum benefit. In our case it was found that the best solutions were achieved by using the following options. First the LP problems solved at the nodes of the branch and bound tree are solved by both dual and primal simplex methods depending on the relative number of rows and constraints of the LP. In the default version OSL uses only the primal simplex method. Scaling of the problem is also performed. We allow OSL to generate 200 cutting planes. OSL generates these cutting planes automatically based on methods for general integer linear models. The branch and bound algorithm is modified in such a way that the utilization of supernodes are allowed. By utilizing supernodes, OSL analyzes many nodes of the branch and bound tree at once. This analysis is based on applying logical tests on the 0-1 structure of the problem using *implication lists* and *probing*. The preprocessor also performs tests to eliminate continuous variables from the LP relaxation and finally the branch and bound tree is kept in core and not in the disk enhancing the processing speed by reducing the amount of time for input/output operations.

The main enhancement in computational performance was obtained by utilizing a tree decomposition scheme similar to the one proposed in one of our previous papers (Voudouris and Grossmann, 1993). The basic characteristic of this decomposition scheme is that the logic inherent in a Mixed Integer Optimization model can be exploited to generate a partial enumeration of the vector of discrete variables that is expressed through a partial Disjunctive Normal Form (DNF). Thus the solution domain is partitioned in a number of subproblems each one of which can be exploited by a smaller instance of the original MILP. For example, in the path selection scheduling problem one potential partitioning scheme is to consider selections of equipment that constitute a feasible flowsheet and to assume existence of the paths that utilize the selected equipment. For every selection, the binary variables for path existence can be fixed accordingly, and the MILP partition will search only the remaining solution space. One such scheme has been employed to enhance the computational performance of the MILPs in our work in design of multipurpose plants with multiple production routes (Voudouris and Grossmann, 1992). In this work the tree partitioning scheme is mainly based on the proper definition of the set SV which indicates the number of repetitions of the elementary schedule. By doing so, in addition to the reduced binary space that has to be searched, it is possible to

define upper and lower bound in the production cycle for every partition. Furthermore, it is possible to identify good values for the parameter W in the disjunctive constraints (26)-(27). Finally efficient upper and lower bounds in the number of batches produced in every path, can be derived for every partition. More specifically, the lower bound in the production cycle is defined by the following equation,

$$p^{lo} = \frac{Z}{cp+1+|SV|} \quad (55)$$

where cp is a parameter which depends on the partition and indicates the starting number of repetitions considered; $|SV|$ is the cardinality of the set of repetitions. The upper bound in the production cycle is defined as,

$$p^{up} = \frac{Z}{cp+1} \quad (56)$$

A reasonable overestimation for the parameter W is to consider this variable equal to the upper bound in the production cycle. The lower bound in the number of batches is given by the equation,

$$nt^* = \lfloor \quad \rfloor \quad (57)$$

whereas the upper bound on the number of batches is,

$$\lceil \quad \rceil = \lceil \tau_h \rceil \quad (58)$$

Finally, there are cases in which the instance that is considered generates models that are particularly large and hard to solve. In these cases it is always possible to solve the models with an ϵ -optimality tolerance instead of the obtaining the globally optimal solution. One of the big advantages of the branch and bound based algorithms is that in every instance it is possible to determine the relaxation gap of the current best integer solution (provided there is a feasible design). This means that if the designer feels comfortable with the current relaxation gap, he/she can terminate the solution procedure and retrieve the currently best integer solution. One characteristic of the branch and bound algorithms is that the globally optimal solution is obtained relatively fast, but in order for the solver to prove optimality by closing the relaxation gap to zero requires significantly more time. This means that in many practical cases working with an ϵ -optimality criterion is a justified alternative.

Numerical Results

Example 3

Here we illustrate the fact that problem (P.3) has the very important characteristic of handling instances with large number of batches belonging to few products. In this example a total of 200 batches of product A, 150 batches of product B, 100 batches of product C and 300 batches of product D must be produced. The rest of the data are the same as in examples 1 and 2. The model used involved 301 constraints, 322 variables of which 12 were discrete. A total of 141 CPU seconds were required to solve the problem to optimality and 2294 tree nodes were enumerated. Again GAMS 2.25/OSL were used to generate and to solve the model on the same computer. The partition used had a lower bound of fifty and an upper bound of 100 on the number of repetition. When the partition with bounds on the number of repetitions of 0 and 50 was used, the solution did not have the overproduction of batches that is reported by the optimal solution, but the optimal time was significantly worse. The optimal schedule and the optimal values of the variables are shown in the Gantt chart of Figure 15. Note that since 200 batches of A are produced, 200 of B, 100 of C and 300 of D, this means that B is overproduced by 50 batches.

Example 4

The data are the same as in example 3. This time, however, it is assumed that unlimited intermediate storage is available between stages 3 and 4 for all the products. The model involved 515 constraints and 531 variables of which 17 were discrete. The optimal schedule is shown in Figure 16. The time required to produce the batches is now 3,600 hours. Thus, when intermediate storage is not utilized, the time required to produce the specified amount of batches is almost 20% higher. In Figure 16 only one repetition of the elementary schedule is shown. The previous and next repetitions will conform with this repetition in such a way that the idle times for the equipment shown in the schedule will be significantly reduced. The time coordination of the subtrains is ensured in the long run. This means for example that the amount of material required for the second batch of product B in the downstream is provided by the first batch of the upstream plus some reminder from the previous repetition. The amount of material of the various products that is kept in the intermediate storage is relatively high in this case. It is however possible to add a constraint in model (P.3) that will for example, constrain the difference between the finish time of the upstream and the start time of the downstream.

GAMS2.25/Sciconic2.11 were used to generate and to solve the model on the same computer and a total of 35 CPU minutes were required. The optimal partition had a lower bound of 20 and an upper bound of 30 repetitions.

Example 5

In this problem the path selection scheduling subproblem is illustrated. We consider an existing process that is indicated in Figure 17. The process involves 7 processing equipment Equipment 2 and 3 are identical and constitute processing stage 2. The same is valid for equipment 5 and 6 which constitute processing stage 4. A total of 3 products will be produced More specifically for product A, 32 batches will be produced; for product B, 21 batches; and for product C, 43 batches. There are two alternative production paths for the batches of product A. These are paths 1 and 2. For product B there are again two alternative production paths 3 and 4, whereas for product C the only production path that the batches can follow, is path 5. The processing times required for every task in every path are indicated in Figure 18. By using model (P.4) we decide which of these production paths do we have to use and how to schedule them in order to produce the specified amount of batches in the least amount of time. Note the since the alternative paths are exactly identical as far as processing times are concerned, we perform a small modification in the objective function in (P.4). This modification consists of adding the sum of the binary variables y_{3h} which denote the path existence, multiplied by a small weight. This weight is sufficiently small so that the optimal solution is not affected. The above modification is necessary in order to identify the solutions which require the least number of equipment but still do not jeopardize optimality. In this particular example the weight we used was 0.1. The model consists of 141 constraints, 106 variables of which 24 were discrete. The optimal partition had a lower bound of 10 and an upper bound of 20 repetitions The optimal schedule is shown in Figure 19. It should be noted that the only path that was not selected is path 3. A total of 16 CPU seconds were required and a total of 573 nodes were enumerated. The matrix generator, solver and computer used are the same as in the previous example.

Example 6

Here we illustrate the application of the MILP model for design and scheduling of sequential multipurpose batch plants. In order to illustrate the difference of perspective between this work with a more typical campaign planning approaches in the literature, we will consider as objective function the minimization of the capital investment instead of the maximization of the NPV since the capital investment is the objective most campaign

approaches utilize. We consider the same process as in example 5 which is shown in Figure 17. The difference now is that we need to decide which equipment will be used and in what size. The processing time data for every path are shown in Figure 18. The size factors, cost data and demand specifications are shown in Table II. A tree partitioning scheme has been used. In this scheme the number of repetitions has been considered in groups of 10 repetitions starting from 0 and ending in 200. For every individual group 4 other partitions have been solved. These partitions have been generated by recognizing that all possible combinations of path and equipment existence are the following. First, only one equipment is used in stages 2 and 4 which means that only paths 2,4, and 5 exist (Or another equivalent alternative that only paths 1,3, and 5 exist). Second, two equipment are used in stage 2 and only 1 in stage 4, which means that only paths 1,2,4, and 5 exist Third, two equipment exist in stage 4 and only 1 in stage 1, which means that only paths 2, 3,4 and 5 exist Finally, 2 equipment exist in both stages 2 and 4, which means that all paths exist. Note that this scheme allows explicit exploitation of structural logic and eliminates degeneracy. A total of 80 MILP subproblems with their corresponding tighter relaxations due to equations (55) - (58) have been solved for the instance. These problems were solved in parallel by utilizing the multitasking capabilities of the IBM/R6000 workstation. The optimum solution was a design of \$247,680. The optimal schedule is shown in Figure 20. This design has been obtained in a large number of partitions. Out of these, the partitions with smaller production cycles have a more efficient utilization of time. The optimal design requires for equipment 1, 3, 4, 6 to be of 6,000 liters, equipment 2 and 5 to be not selected, and equipment 7 to be of 10,000 liters.

The same problem was solved with the campaign approach used by Voudouris and Grossmann (1992) for designing multipurpose batch plants with multiple production routes. The optimal design in this case required the same equipment to be selected but now equipment 6 and 7 had to be of 10,000 liters. The cost of the process in this case is \$280,000 or about 14% higher. This difference is explained by the fact that in this case the campaign mode will dedicate long campaigns to every product and therefore will generate large idle times to the equipment that are not utilized at each campaign. It is interesting to note that when the optimal schedule of example 5 was considered (by properly constraining the problem), the optimal design was \$278,171 which even though is worse than the best solution, is still better than the solution reported with the campaign approach.

The partition which generated the optimal solution had 274 constraints and 229 variables of which 19 were discrete. The same partition required 195 CPU seconds in the

same computer as above to enumerate 3376 nodes. GAMS2.25 was utilized for matrix generation and OSL was the solver. Although the solution of the model is computationally demanding (80 parallel problems of about 4 minutes each), we believe that using a proper partition scheme in a distributed computing environment (Kudva and Pekny, 1993) allows the solution of problems whose dimensionality is restricted only by the number of available processes in the computer network.

Example 7

Note that in example 6 the objective was the minimization of capital investment which is utilized by the campaign approach mentioned in the literature. It has been shown in example 6 that the integration of rigorous scheduling and design offers significant savings in investment. A significant advantage, however, of the formulations proposed in this work is the fact that the operating costs are also incorporated in the design procedure. In this example the profitability of the process as expressed by the Net Present Value will be illustrated.

The data in this example are the same as the ones in example 6. In addition, it is assumed that the market prices of the three products are 0.7 \$/kg for A, 0.4 \$/kg for B and 0.5 \$/kg for C. The cost of keeping the final product in inventory is assumed to be the same for all products and equal to 0.1 \$/ton/hr. The taxation rate is 45%, the interest rate is 10%, the expected life of the plant is 10 years.

The solution reported by the campaign approach used by Voudouris and Grossmann (1992) requires, as mentioned in the previous example, \$280,000 in capital investment. The production plan requires that the three products be produced in 3 campaigns. The first campaign has a length of 2908 hrs and only product A will be produced in it. The second campaign has a length of 1100 hrs and only product B will be produced during the whole campaign. Finally the third campaign has a length of 1472 hrs and only product C will be produced in it. For the calculation of the operating cost zero changeover cost is assumed and the inventory costs are calculated as shown in equation (40) but with the time component corrected to $(P-T_j)$ instead of P . The reason for this correction is the fact that the campaign approach considers a production cycle to be equal to the design horizon. For this case, however, the inventory policy shown in Figure 14 tends to give significant overestimations of the inventory costs since the depletion due to product selling is not considered. The use of the term $(P-T^*)$ instead of P addresses exactly this depletion. More details on the inventory policy that best describes cases of large production cycles is given in Voudouris and Grossmann (1993). By considering all the above, the operating costs are 228,802 \$/yr, and the NPV is \$ 1,203,967.

The optimal design that has been obtained by using model (P.8) is utilizing the schedule that is illustrated in Figure 20 and involves equipment **1,3,4** and 6 with a size of 6,000 liters and equipment 7 with a size of 10,000 liters. The capital investment for this design is \$247,680 and the NPV is \$1,992,901 which means an increase in profitability of 65.5 % compared to the previous case! Note that the operating costs in this case have been calculated with equation (40) and correspond to \$ 2,275 per year.

Again the partition scheme of the previous example has been used. The partition which generated the optimal solution had 277 constraints and 235 variables of which 19 were discrete. The optimal number of repetitions of the elementary schedule was 146. The optimal partition required 164 CPU seconds in the same computer as in the previous examples in order to enumerate 2836 nodes. GAMS2.25 was utilized for matrix generation and OSL was the solver.

Conclusions **and** significance

In this work we have addressed the scheduling and design of sequential multipurpose batch plants. Even though this class is more restrictive than the general nonsequential multipurpose batch plants, it is still significantly more general than the multiproduct case. Furthermore, the mathematical structure of this problem can be exploited to significantly reduce the computational difficulty. More specifically, a reduction scheme was proposed that yields a significant decrease of the binary dimensionality of the models. In addition, an aggregation scheme based on a periodic scheduling was derived that allows the consideration of problems of practical size. The scheduling models were successfully incorporated with the design and synthesis problems making possible a global approach to the preliminary design of batch processes.

By considering the availability of equipment in standard sizes, it was possible to derive MDLP models which can be solved to global optimality. A number of solution techniques were suggested to improve computational performance and permit in this manner, the consideration of larger practical problems. Finally, an example was presented to show that a significantly lower capital investment can be obtained compared to methods that assume production campaigns.

Acknowledgment

We gratefully acknowledge financial support by the National Science Foundation under Grant CTS-9209671.

Nomenclature.

Indices.

- i** Index of individual operations (tasks).
- k** Index of processing equipment.
- p** Index of products.
- h** Index of processing paths.
- s** Index of discrete sizes for processing equipment {1,2,__, nsk}.
- sv** Index of number of production cycles.

Variables/Parameters.

- B_h** Batch size for path h.
- dh^f*k** Cleanup time required in equipment k when path h^f follows path h.
- cp** Parameter indicating smaller number of repetitions considered in each tree partition.
- dhk** Processing time of the operation of path h on equipment k.
- fhk** finishing time of the last batch of path h on equipment k.
- H** Time horizon in which the demand has to be satisfied.
- Ms** Makespan of a schedule.
- nbh** Number of batches produced in path h during a production cycle,
- npp** Number of batches of product p during the horizon.
- Nr** Number of production cycles during the horizon.
- Oc** Operating costs
- P** Length of production cycle.
- Pc** Capital investment
- q_h** Amount of production by path h during one production cycle.
- Q_p** Market demand for product p.
- Sy** Size factor of equipment k for the proper operation of path h.

$S_{k,h,h'}$ Idle time (slack) imposed in equipment k when path h' follows path h.

t_{1k} Start time of the first batch of path h on equipment k.

\bar{t}_h Start time for the first operation of path h.

T_c Total time required to satisfy demands on number of batches.

T_h Cycle time for path h.

V_k Volume of equipment k.

v_{ks} Standard volume of size s for equipment k.

Z rational number that ensures time coordination of neighbouring subtrains.

Its value is a rational number close or equal to the ratio of the cycle times.

Greek Letters.

α_k Cost coefficient for equipment k

β_k Cost exponent for equipment k.

References.

- Applegate D and Cook W. (1991). A Computational Study of the Job-Shop Scheduling Problem. *ORSA Journal on Computing*, Vol.3, No.2, pp 149-156.
- Balas E. (1985a). Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM J. Alg. Disc. Meth.*, Vol.6, No.3, pp 466-486.
- Balas E. (1985b). On the Facial Structure of Scheduling Polyhedra. *Mathematical Programming Study*, Vol.24, pp 179-218.
- Birewar D.B and Grossmann I.E (1989). Efficient Optimization Algorithms for Zero Wait Scheduling of Multiproduct Batch Plants, *Ind. Eng. Chem. Res.*, Vol 28, pp 1333-1345.
- Brooke, A., Kendrick, D. and Meeraus, A. (1988). GAMS: A User's Guide. Scientific Press, Palo Alto.
- Dyer M.E and Wolsey L.A. (1990). Formulating the Single Machine Sequencing Problem with **Release** Dates as a Mixed Integer Program. *Discrete Applied Mathematics*, 26, pp 255-270.
- Faqir N.M, Karimi LA. (1990), Design of Multipurpose Batch Plants with Multiple Production Routes, *Proceedings FOCAPD'89*, Snowmass Village CO, pp 451-468
- Grossmann I.E, Voudouris V.T., Ghattas O. (1992). Mixed-Integer Linear Programming Reformulation for Some Nonlinear Discrete Design Optimization Problems, *Recent Advances in Global Optimization* (eds, Floudas, C.A.. and Pardalos, P.M.), pp.478-512, Princeton University Press.
- Karimi LA. and Reklaitis G.V. (1985). Intermediate Storage in Noncontinuous Processes Involving Stages of Parallel Units. *AIChE Journal*, Vol.31, No. 1, pp 44-52.
- Kondili E. Pantelides C.C. and Sargent R.W.H. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations-I. MILP Formulation. *Comp&ChemEng*, 17, No.2, pp 211-227.
- Ku H.M and Karimi LA. (1990). Completion Time Algorithms for Serial Multiproduct Batch Processes with Shared Storage. *Comp&Chem.Eng.*, Vol 14, No 1, pp 49-69.

Kudva G. and Pekny J.F., (1993). "DCABB: A Distributed Central Architecture for Branch and Bound Calculations", Computer Integrated Process Operations Center, Report CEPAC-93-13, Purdue University.

Modi A.K and Karimi L.A. (1989). Design of Multiproduct Batch Processes with Finite Intermediate Storage. *Comp&Chem.Eng.* 13, No 1/2,127-139.

Nemhauser G.L, Wolsey L.A. Integer and Combinatorial Optimization, Wiley, 1988

OSL, Optimization Subroutine Library, Guide and Reference. Release 2,1991. IBM Corp. USA

Papageorgaki S. and Reklaitis G.V (1990) Optimal Design of Multipurpose Batch Plants-1. Problem Formulation, *Ind&Eng.ChemJ(es.yo)* 29, No. 10, pp 2054-2062

Raman, R and Grossmann I.E. (1992). Modeling and Computational Techniques for Logic Based Mixed Integer Programming. Annual AIChE Meeting, Miami Beach, Florida.

Reklaitis G.V (1990) Progress and Issues in Computer-Aided Batch Process Design, *FOCAPD Proceedings*, Elsevier, NY, pp 241-275

Rippin D.W.T. (1993). Batch Process Systems Engineering: A Retrospective and Prospective Review. *Computers and Chem. Eng.*, 17, Suppl., S1-S13.

SCICONIC/VM 2.11 (1991). Users Guide", Scicon Ltd, U.K.

Shah N. and Pantelides C.C. (1992). Optimal Long-Term Campaign Planning and Design of Batch Operations, *Ind, Eng. Chem. Res.*, Vol 30, No. 10, pp 2308-2321

Vaselenak J.A , Grossmann I.E, Westerberg A.W.(1987). An Embedding Formulation for the Optimal Scheduling and Design of Multipurpose Batch Plants, *Ind. Eng. Chem. Res.*, Vol 26, No.1, pp139-148

Voudouris V.T and Grossmann I.E. (1992). Mixed Integer Linear Programming Reformulations for Batch Process Design with Discrete Equipment Sizes, *Ind. Eng. Chem. Res.*, 31, No.5,1314-1326

Voudouris V.T and Grossmann I.E. (1993). Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations. *Ind. Eng. Chem. Res.* 32, No. 9,1962-1980

Wellons M.C. and Reklaitis G.V. (1991). Scheduling of Multipurpose Batch Chemical Plants-2. Multiple-Product Campaign Formation and Production Planning. *Ind. Eng. Chem. Res.*, 30, pp 688-70S

Appendix I. Cutting planes for model (P.2).

Dyer and Wolsey (1990) proposed a number of strong valid inequalities for the problem of minimizing the weighted sum of completion times for the one-machine scheduling problem with release times and due dates. Applegate and Cook (1991) modified the tightest of these inequalities for the general machine scheduling problem and considered it as cutting planes in model (P. 1)- The equations they proposed are,

$$\sum_{h: (h, k) \in M} d_{hk} t_{hk} \wedge \hat{E}_k \hat{d}_h + \sum_{(h, h'): (h, h \setminus k) \in E^*} < i_{hk} d_{h \cdot k} \quad \forall k \quad (\text{A.1})$$

$$\sum_{h: (h, k) \in M} d_{hk} (M_s - t_{hk}) \geq \hat{F}_k \hat{d}_h + \sum_{h: (h, k) \in M} d_{hk}^2 + \sum_{(h, h'): (h, h \setminus k) \in E^*} d_{hk} d_{h'k} \quad \forall k \quad (\text{A.2})$$

where \hat{E}_k is the minimum of the earliest possible start time of the operation of path h on machine k , and \hat{F}_k is the minimum of the earliest possible completion time of the operation of path h on machine k . The earliest possible start time of the operation of path h on machine k is calculated by adding the processing times of the operations of the same path on all the preceding machines. The earliest possible completion time of the operation of path h on machine k is calculated by adding the processing times of the operations of the same path on all the remaining machines, d_k is the sum of the processing times of the operations of all paths h that utilize machine k . By utilizing equation (1) the above inequalities can be written as,

$$\sum_{h: (h, k) \in M} d_{hk} \bar{0}_{h+} + \sum_{k=1}^k d_{hk} \geq \hat{E}_k \hat{d}_h + \sum_{(h, h'): (h, h \setminus k) \in E^*} d_{hk} d_{h'k} \quad \forall k \quad (\text{A.3})$$

$$\sum_{h: (h, k) \in M} d_{hk} (M_s - (t_h + \sum_{k=1}^k d_{hk})) \geq 2\hat{F}_k \hat{d}_h + \sum_{h: (h, k) \in M} d_{hk}^2 + \sum_{(h, h'): (h, h', k) \in E^*} d_{hk} d_{h'k} \quad \forall k \quad (\text{A.4})$$

A similar modification can be applied to the two-job cutting planes proposed by Balas (1985). For model (P.2) the triangular cutting planes are written as,

$$Y_{hk} + Y_{h'k} - Y_{hh'} \leq 1 \quad \forall (h, h', k), (h, h') \in E^+ \quad (\text{A.5})$$

which in logical terms means that when the operation of path h is before the operation of path h' on machine k , and the operation of path h' is before the operation of path h'' then the operation of path h'' must have place after the operation of path h on the same machine.

All of the above mentioned cutting planes assist in reducing the relaxation gap of model (P.2). From our experience, however, it seems that the strong inequalities (A.3) and (A.5) are dominant to the rest of the cutting planes. Therefore, in the final version of the model we included only constraints (A.3) and (A.5) because the incorporation of the rest only adds to the number of constraints.

Appendix II. Reduction scheme.

Proposition 1. Problem (P.2) is equivalent to a reduced problem (RP.2), in which the disjunctions are defined only over the set R rather than set $E+$.

Proof: We know that problem (P.2) is equivalent to problem (P.I) plus equation (1). Therefore, it suffices to prove that the reduced problem (RP.2) is equivalent to (P.I) plus (1). Consider a triad $(h, h^f, k^*) \in E_{AR}$. We will prove that any disjunctive constraint defined by this triad is redundant. Assume initially that $y_{hh^f k^*} = 1$ (exactly the same procedure applies when $y_{hh^f k^*} = 0$). This means that the fourth constraint in (P.I) is redundant and that the third constraint takes the form,

$$t_{h^f k^*} - t_{hk^*} \geq d_{hk^*} \quad (CB.1)$$

or by introducing a non negative slack a ,

$$t_{h^f k^*} = t_{hk^*} + d_{hk^*} + a \quad (B.I)$$

Since $(h, h^f, k^*) \in R$ it follows from the conditions in (9) that,

$$S_{t_{h^f k^*}} > 0 \quad (B.2)$$

From the definition of $S_{t_{h^f k^*}}$ it is clear that for every pair h and h^f there is at least one triad (h, h^f, k) such that,

$$S_{t_{h^f k}} > 0 \quad (B.3)$$

Consider first the instance in which the disjunction imposed by (h, h^f, k) is arbitrated by $y_{hh^f k} = 1$. This means that the only nonredundant constraint imposed by this triad is,

$$t_{hk} - t_{h^f k} \geq d_{hk} \quad (CB.2)$$

or by introducing a non negative slack p ,

$$t_{hk} = t_{h^f k} + d_{hk} + p \quad (B.4)$$

By subtracting equation (B.4) from (B.I) we get

$$t_h Y - t_h k = t_h k'' - t_h k + d_h k^{\#} - d_{hk} + (a - p) \quad (B.5)$$

From the definition of the slacks it is easy to verify that the following is a valid system of equations,

$$t_h' k^* = t_h k^* + d_h k^* + S l_{hh'} k^*$$

$$t_h' k = t_h k + d_h k$$

By subtracting these equations we get,

$$t_h' k' - t_h k = t_h k^* - t_h k + d_h k'' - d_{hk} + S l_{hh'} k^* \quad (B.6)$$

From (B.6) and (B.5) we get that,

$$\alpha - \beta = S l_{hh'} k^*$$

which from (B.2) yields $a > p$. This in turn implies that constraint (CB.1) is redundant with respect to (CB.2).

We consider next the instance in which the disjunction imposed by $(h, h \setminus k)$ is defined by $y_{hh'} k = 0$. This means that the only nonredundant constraint imposed by this triad is,

$$t_h k - t_h^* k \leq d_h k \quad (CB.3)$$

or by introducing a non negative slack y ,

$$t_h k = t_h^* k + 4k + y \quad (B.7)$$

adding (B.7) to (B.I) we get the following equality,

$$t_h' k' - t_h' k = t_h k^* - t_h^* k + d_h k^* + d_h' k + Y + \alpha \quad (B.8)$$

Adding and subtracting $d_h k$ on the right hand side gets,

$$t_h^k - t_h^k = t_h^k - t_h^k + d_h^k + dh^k + dh_k - dh_k + Y + <^* \quad (\text{B.9})$$

Comparing equation (B.9) and (B.6) yields,

$$SlkhV = dh^k + dh_k + a + y \quad (\text{B.10})$$

Since $(h, h^k) \in R$ it follows from the conditions in (9) that,

$$SlkhV < Plh_k \leq d_{hk} + d_{h^k} \quad \forall k: (h, h^k) \in E^+$$

From the above result and (B.10) it follows that,

$$a + Y < 0$$

which is impossible for non negative a and Y .

Therefore when $yh^k = 1$ the instance in which $yh^k = 0$ is infeasible. The **only** feasible instance is when $yh^k = 1$ in which case the triad (h, h^k) defines only redundant disjunctive constraints.

Using exactly the same procedure we can find that $yh^k = 0$ also defines only redundant constraints. So the triad (h, h^k) can be ignored in (P.2).

In conclusion, the reduced model (RP.2) is equivalent to model (P.2) because all the additional constraints in (P.2) are redundant. As a final point it should be noted that since $R \subseteq E^+$, the number of disjunctions in model (RP.2) is smaller than the one in model (P.2). Actually because of the quite restrictive nature of the conditions imposed when set R was defined, the number of disjunctions is significantly reduced. For example in the problem defined in the first example the number was reduced from 172 to only 110.

Appendix III. Timing of the elementary schedule.

Proposition 2 : The production cycle (or cycle time of the elementary schedule) can be defined rigorously by the following equation,

$$P = \max_{(h, h \setminus k) \in E} \left[t_h + (nb_h - 1) T_{1h} + \sum_{k'=1}^k d_{hk'} + \left(\bar{V} + \sum_{k'=1}^{k-1} d_{hk'} \right) \right]$$

Proof: It is obvious from Figure A.I that for every sequential repetition r and r^f in the production of a batch through path h , the following constraint holds,

$$d_{hk} + S_{1k} = d_{h(k-1)} + S_{1k} + D \quad \forall (h, k) \in M, (h, k-1) \in M \quad (C1)$$

The finish time f_{hk} of the operation of path h on machine k is expressed by the following equation,

$$f_{hk} = t_{hk} + (d_{1k} + S_{1k}) + (d_{2k} + S_{2k}) + \dots - K_{nbh} + S_{1k} \quad \forall (h, k) \in M \quad (C.2)$$

For every path h there is a stage k where the processing time t_{hk} is the maximum for all operations of the path h . This stage is considered as the *bottleneck* stage and its processing time is referred to as *Cycle time* for path h and is noted as T_{1h} . Because the optimization direction is to minimize P and thus to minimize the slacks S_{1k} (in case they were variables), it follows from (C.1) that the slack for the bottleneck stage is zero. In Figure (A.I) for example the third stage is the bottleneck stage for the 3 batches of product A and the cycle time for product A is 4 hours. Thus the following constraint holds,

$$T_{1h} = t_{hk} + S_{1k} \quad \forall (h, k) \in M, (r, r^f) \text{ are sequential batches produced through path } h \quad (C.3)$$

In equation (C.2) the above term exists $nb_h - 1$ times for each path h , where nb_h is the number of batches produced through path h . For the example in Figure (A.I) the term in equation (C.3) for product A exists two times in equation (C.2). For this reason equation (C.2) can be restated as

$$f_{hk} = t_{hk} + (nb_h - 1) T_{1h} \quad \forall (h, k) \in M \quad (C.4)$$

By using equation (1) the above can be restated as,

$$f_{hk} = \bar{t}_h + \sum_{k'=i}^k d_{hk'} + (nbh - 1) T_{in} \quad \forall (h,j) \in M \quad (C.5)$$

The elementary schedule consists of time intervals during which the various processing equipment are utilized. These time intervals have constant relationship to each other. For this reason the optimal production cycle is defined as the time intervals with the maximum duration. This can be stated as,

$$P = \max_{(h,h|k) \in E} \{ f_h^* - t_{hk} \} \quad (C.6)$$

By considering equation (1) and (C.5), the above can be written as,

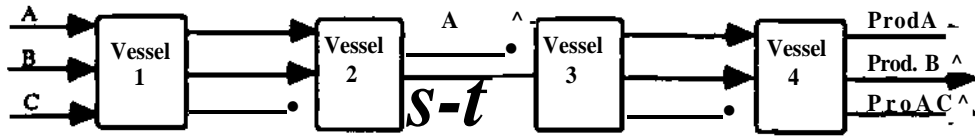
$$P = \max \left\{ \bar{t}_h + \sum_{k'=i}^k d_{hk'} + (nbh - 1) T_{in} - (\bar{t}_h + \sum_{k'=i}^k d_{hk'}) \right\} \quad (C.7)$$

h	A	A	A	A	A	C	C	C	C	C
h ¹	B	B	B	B	B	D	D	D	D	D
k	1	2	3	4	5	1	2	3	4	5
Slack	0	-	-	-	2	-	2	-	-	0

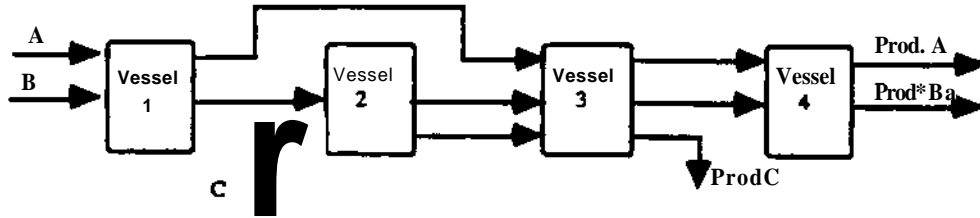
Table I. Slacks for example on Figure 7.

	Size Factors (liters/kg)							Demands (kR/yr)	
	Eq. 1	Ea. 2	Eq.3	Ea.4	Eq. 5	Eq.6	Eq.7		Thousands
Path1	3.5	5.0		4.0				A	350
Path 2	3.5		5.0	4.0					
Path 3				10	55		3.8	B	400
Path 4				2.0		5.5	3.8		
Path 5	4.6			3.7			6.3	C	480
Discrete Sizes S=f (3000,6000, 10000) liters Design Horizon = 5480 hours									

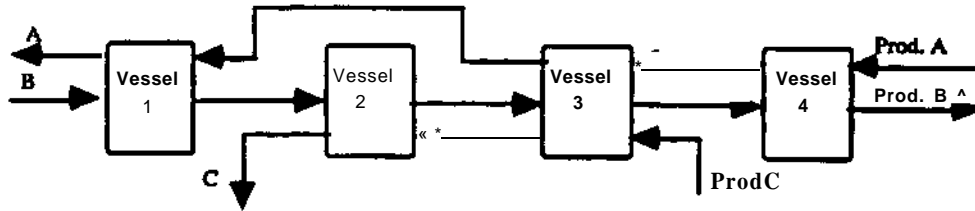
Table II. Data for example 6



a) Multiproduct batch plant



b) Sequential multipurpose batch plant



c) Non sequential multipurpose batch plant

Figure 1. Classifications of batch plants

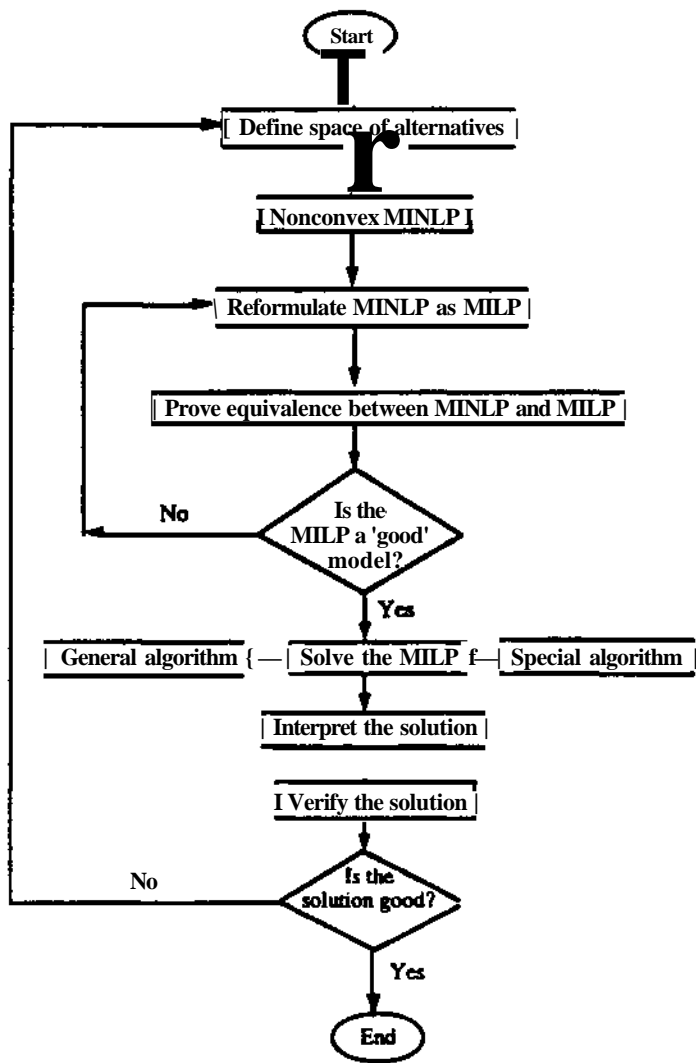


Figure 2. Design approach

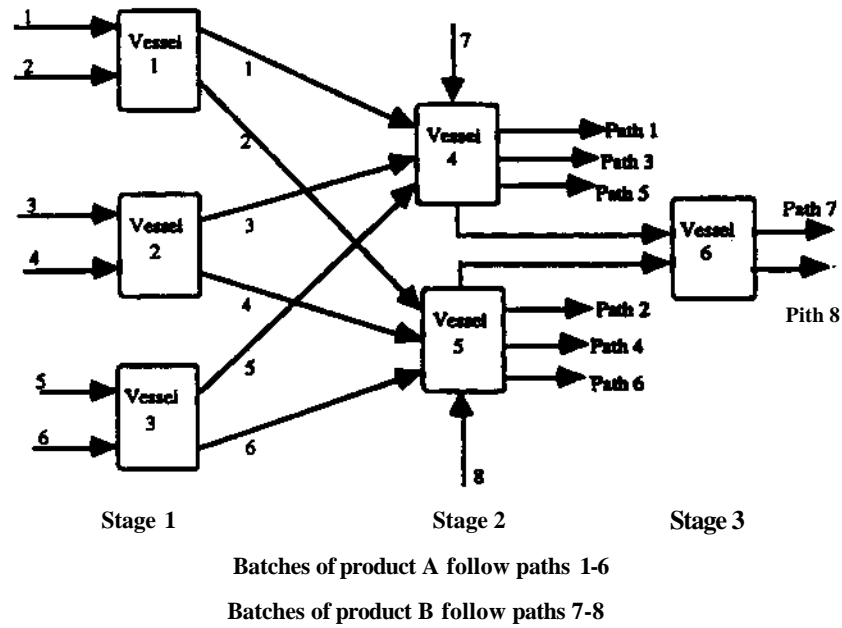


Figure 3. Production paths in a sequential multipurpose plant

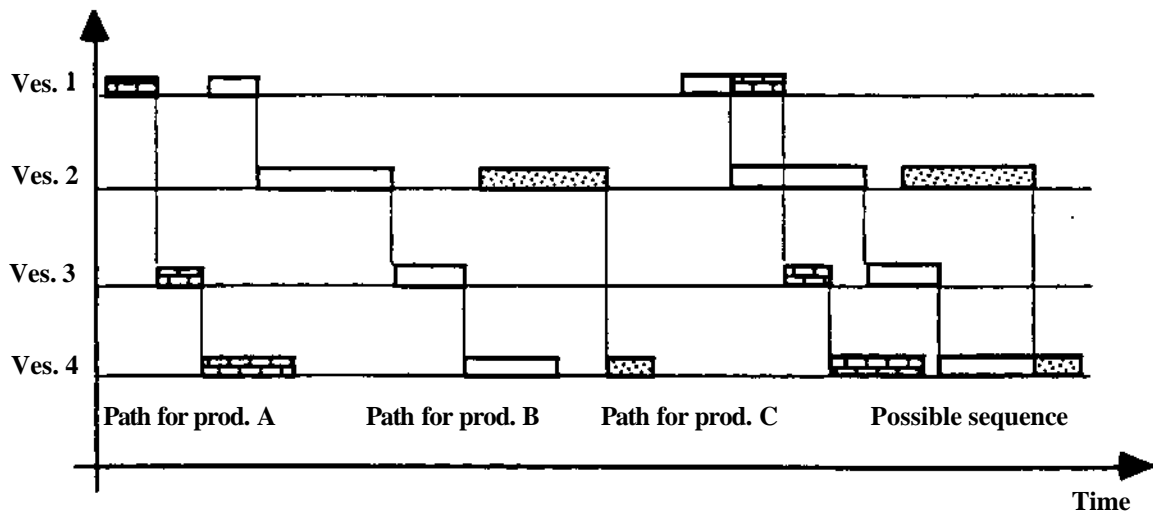


Figure 4. Gantt charts for process in Figure 1.b

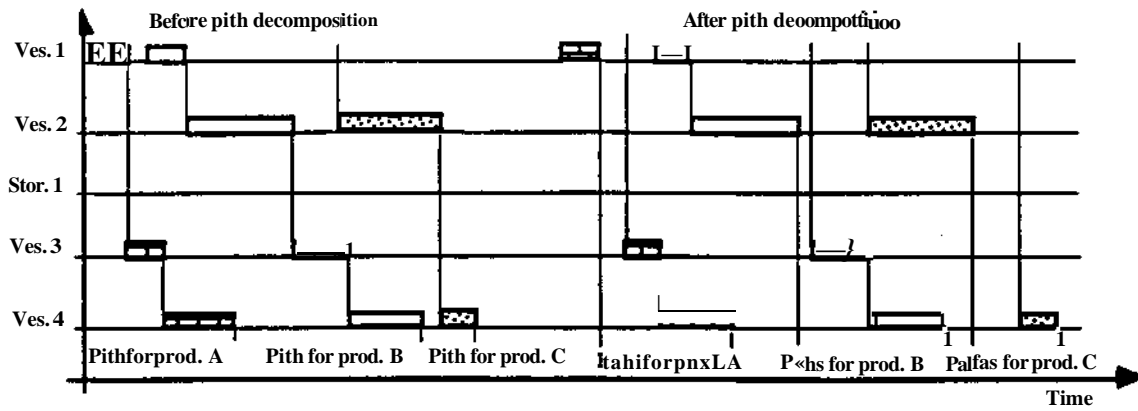


Figure 5. Consideration of intermediate storage vessels

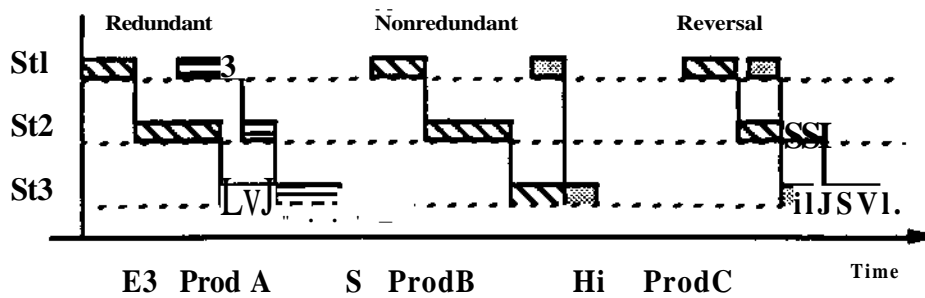


Figure 6. Geometric principles of the reduction scheme

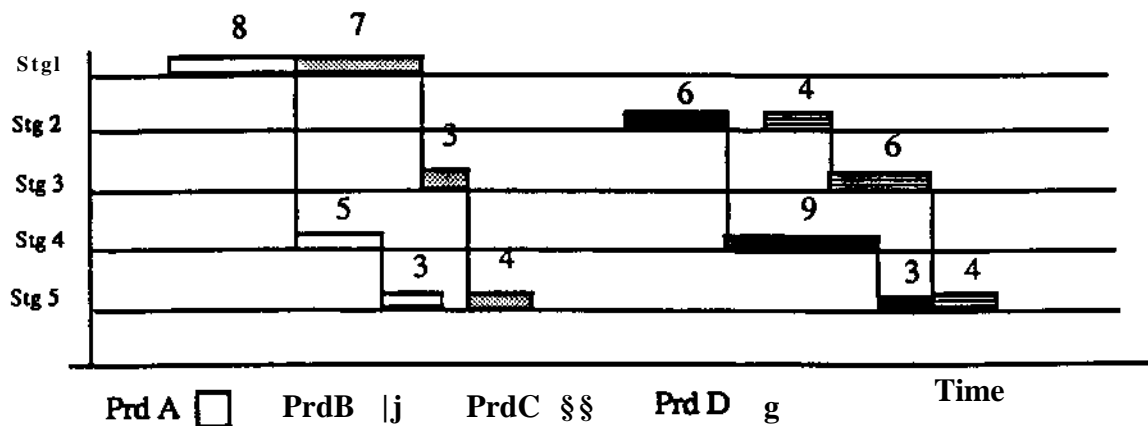


Figure 7. Geometrical interpretation of slacks

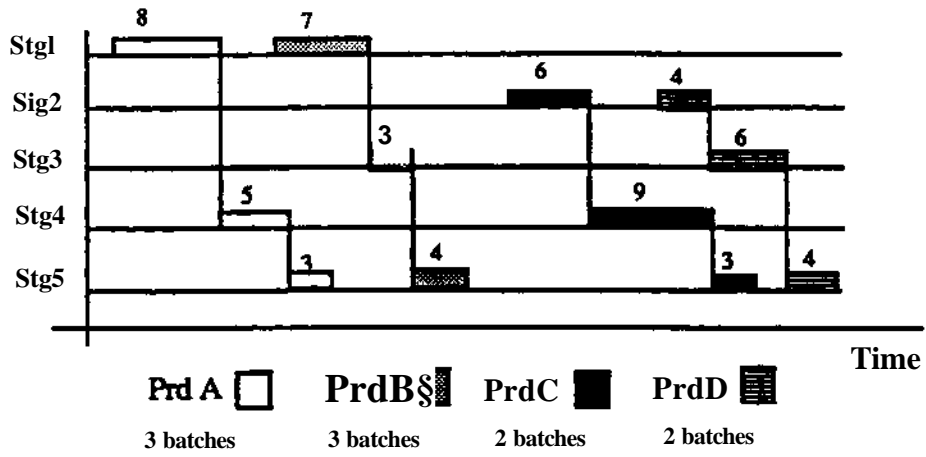


Figure 8. Data for example 1

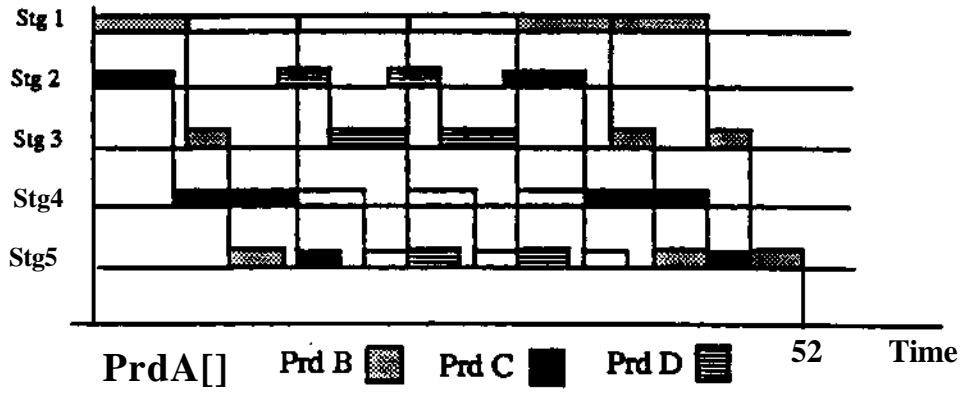


Figure 9. Optimal schedule for example 1

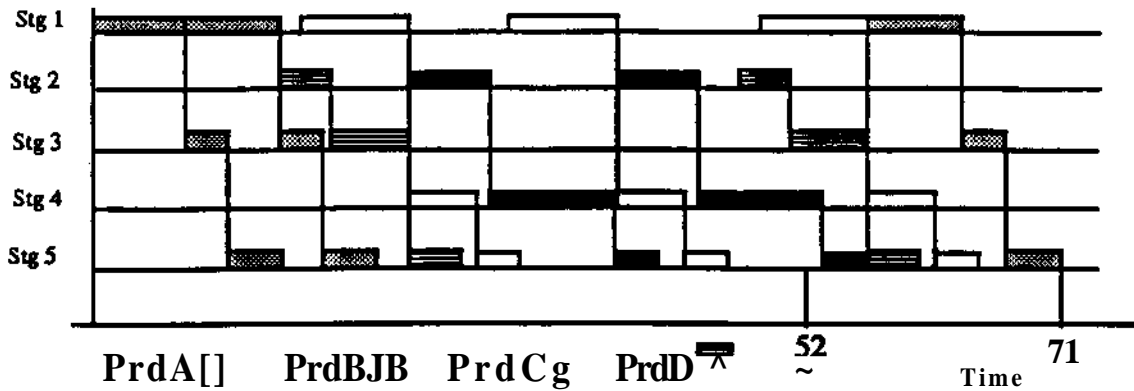


Figure 10. Optimal multiproduct schedule for example 1

Prod A (3 batches)
 Prod B (6 batches)
 Prod C (3 batches)

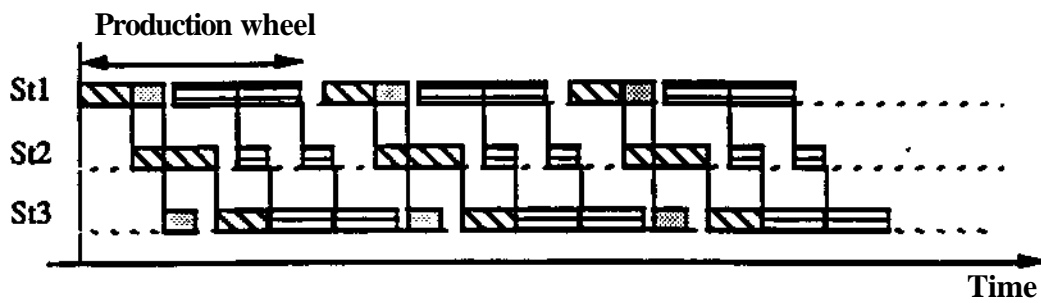


Figure 11. Periodic scheduling approach.

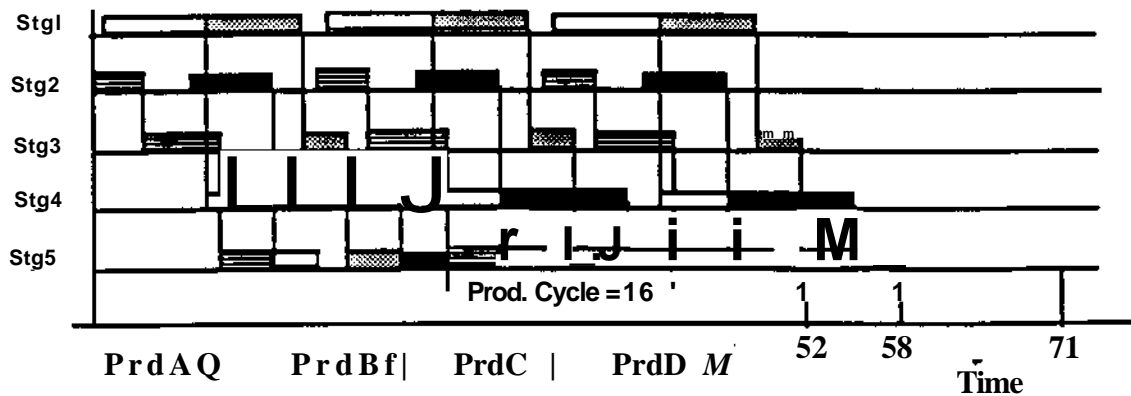


Figure 12. Aggregated schedule for example 1.

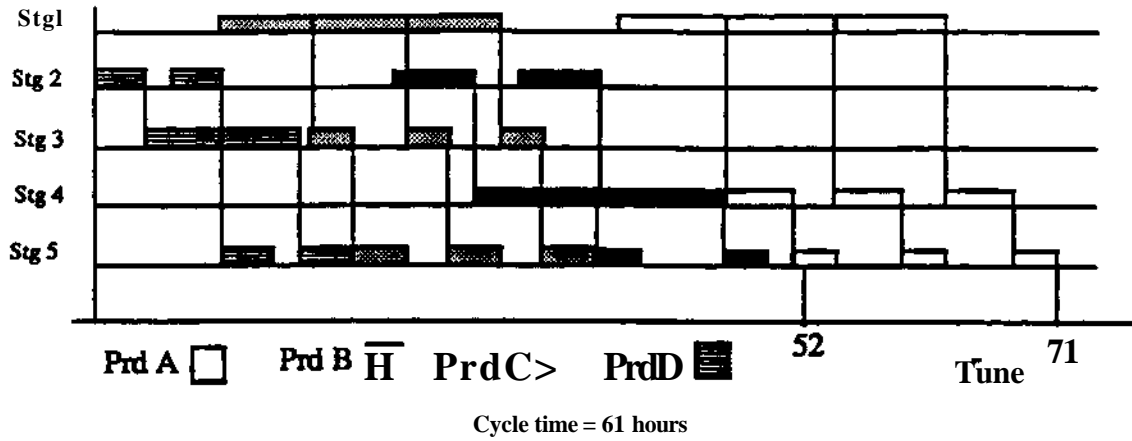


Figure 13. One repetition fixed for example 2.

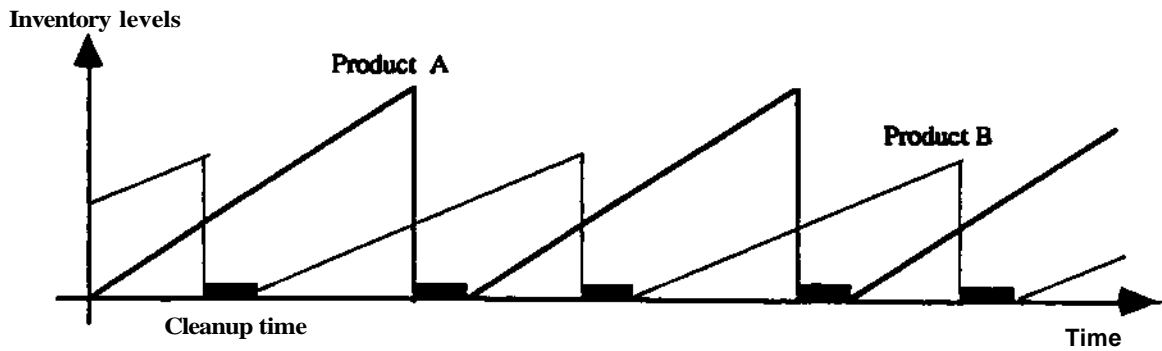


Figure 14. Inventory profiles

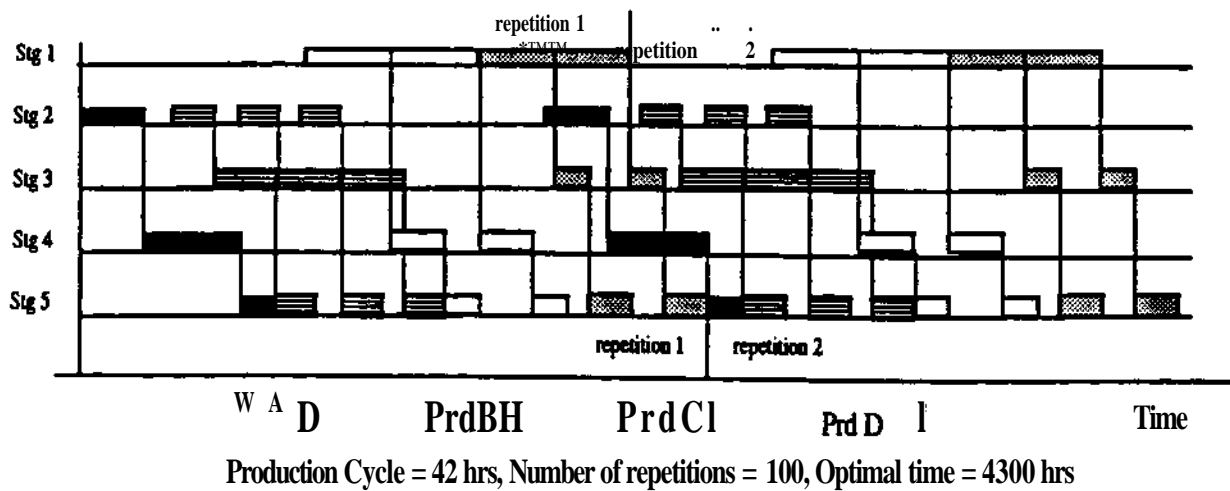
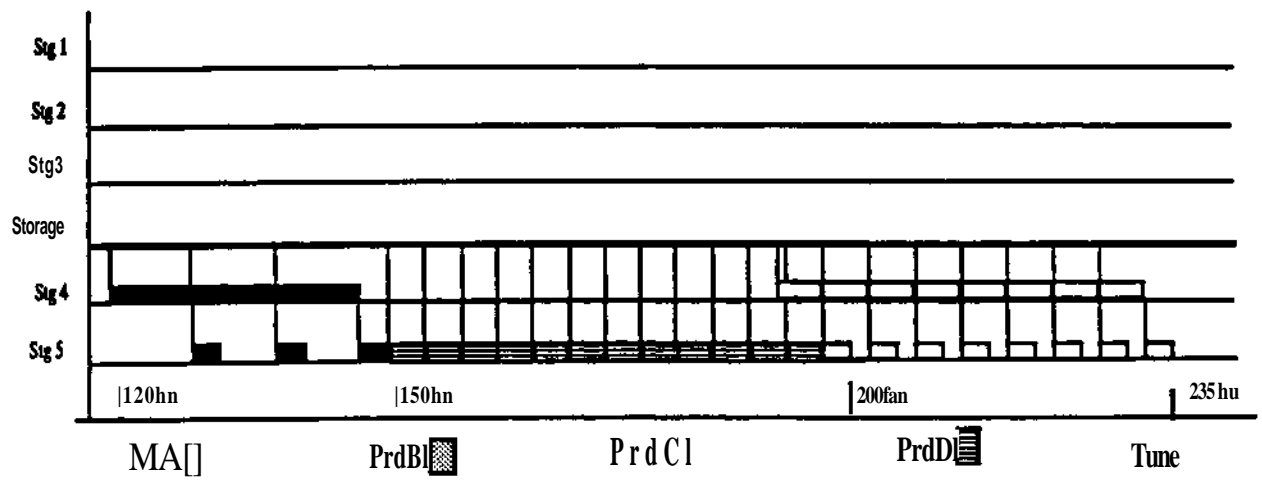
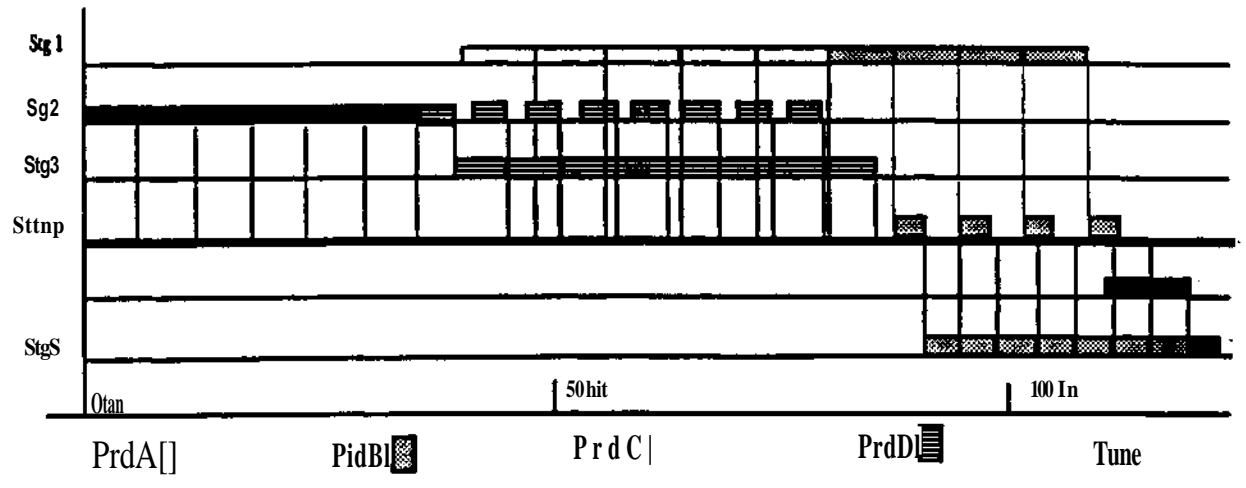
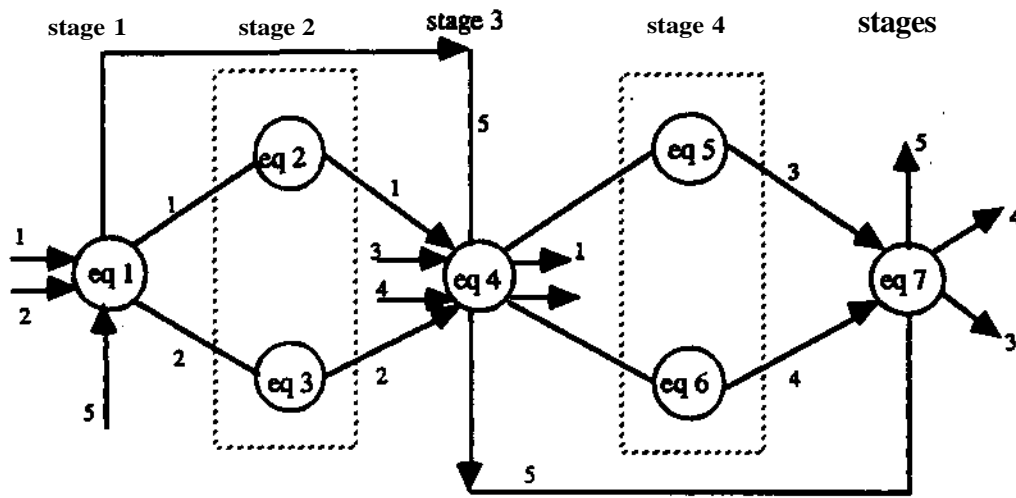


Figure 15. Optimal schedule for example 3



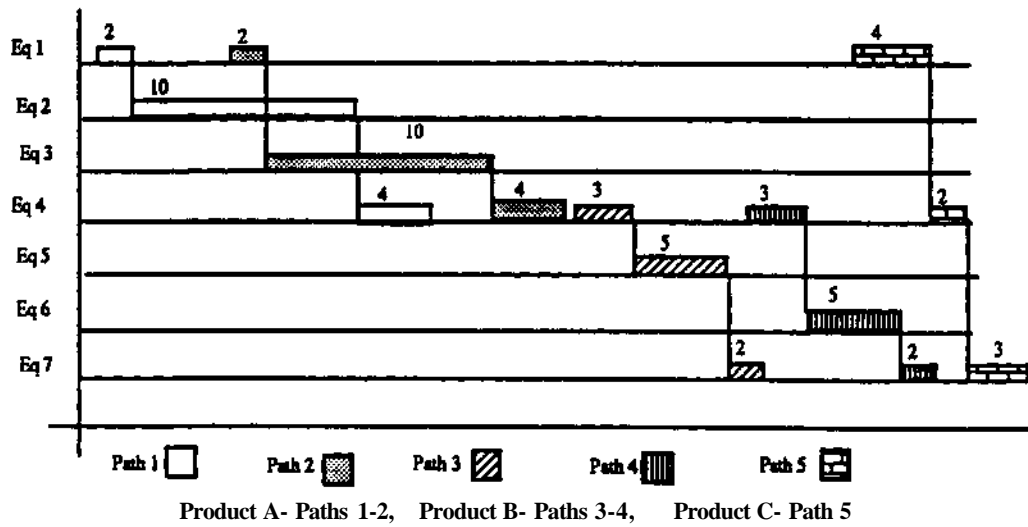
Production Cycle = 144 his, Number of repetitions=25, Optimal time=3600 his

Figure 16. Optimal schedule for example 4.



Product A- Paths 1-2, Product B - Paths 3-4, Product C - Path 5

Rgure 17. Layout of the existing process for example 4.



Product A- Paths 1-2, Product B- Paths 3-4, Product C- Path 5

Figure 18. Processing time data for example 5.

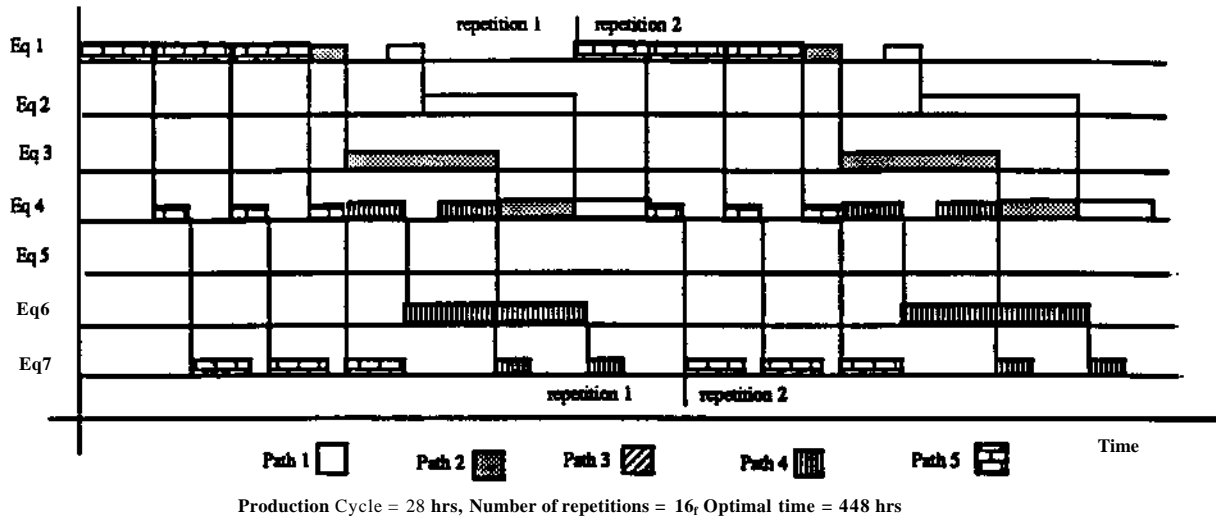


Figure 19. Two repetitions of the optimal elementary schedule for example 5

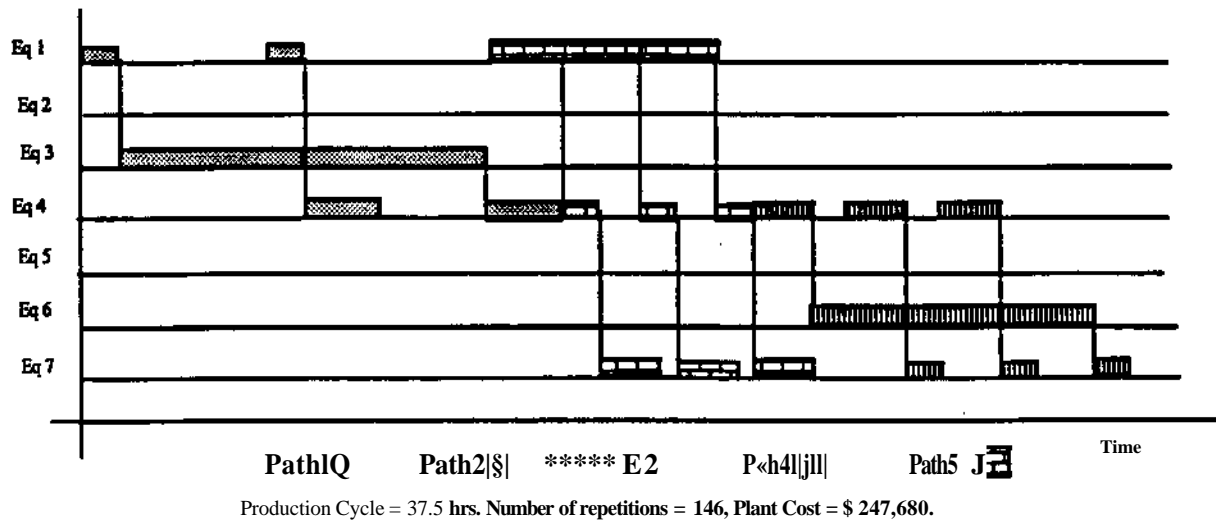


Figure 20. Optimal elementary schedule for examples 6 and 7.

