Optimal Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations

VasiliosT. Voudouris, Ignacio E. Grossmann

EDRC 06-145-92

Optimal Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations

Vasilios T. Voudouris and Ignacio E. Grossmann
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Presented in AIChE 1992 Annual Meeting. November 1992. Paper 135c

Abstract

This paper addresses the problem of determining the optimal configuration and cyclic operation of batch plants in which all the products require the same processing sequence. In particular, the problem can be stated as follows. Given are demands of a number of products, as well as technical information on the processing tasks (size factors, processing times, clean-up times) which are not restricted to a zero-wait policy. Given are also cost data for investment and product inventories, a list of candidate equipment and a list of candidate storage vessels with standard sizes. The problem then consists in determining the following items: number, type and size of equipment, as well as their allocation to one or multiple tasks and possible parallel operation; location and size of intermediate storage vessels; the length of the production cycle including the sequence of production of the products; levels of product inventories. The objective is to maximize the net present value. The major complication of this design problem lies in the many trade-offs that are involved, as for instance the merging of tasks versus its impact on the schedule, and length of production cycle versus inventory levels. By using a novel representation for cyclic schedules and exact linearization schemes, it is shown that this problem can for formulated as a *mixed-integer linear programming problem,* and solved rigorously to global optimality. An efficient computational scheme is proposed for this purpose. Compared to the previous work by Birewar and Grossmann (1990), the proposed model provides a significant extension of the scope of the operational problem, while at the same time yielding an optimization problem that does not involve nonlinearities. Several example problems are presented to illustrate the capability of this method.

Introduction.

Batch processing offers distinct advantages for the production of many specialty chemicals. Of major importance is the flexibility to modify the product envelope and introduce or remove products with short or medium life cycle from the production line. Other advantages include the relative ease with which chemistry intensive processes can be scaled up from the laboratory bench to the production line. Also, in case very stringent quality standards have to be met by the products, batch processing is the preferred mode of operation because potential contamination can be limited only to one batch. Because of these characteristics most of the high value added chemicals are produced through batch processes. In fact batch processing constitutes a significant fraction in the Chemical Process Industries. For example 80% of pharmaceutical and 65% of the food and beverage processes are batch processes (Reeve, 1992). In both of the above types of processes quality is of crucial importance.

Because of the multiproduct nature of batch processes, the logistics of operation, such as scheduling of the products and inventory handling, has to be considered in the design stage making the problem of designing these processes significantly more difficult compared to continuous processes.

Despite the recent development of design models and techniques for batch processes (see Reklaitis, (1990) for a review) there is still a lack of comprehensive design methodologies that can properly address the many aspects involved in batch processes. This work is an attempt to expand the scope of systematic methods for the preliminary design of multiproduct batch processes by integrating the synthesis, design, production planning and scheduling problems.

This paper is a preliminary draft in which the following problem is addressed. Given is a set of N products, the required processing steps for the production of these products, the demands of the products and the time in which these demands have to be satisfied which is referred to as the design horizon H. The overall objective is to find the optimal flowsheet as well as the optimal operation of a batch process that will produce these products. The following decisions are involved in this problem.

1) *Synthesis decisions.*

a) Allocation of tasks to equipment.

b Parallel units of equal size operating either in-phase or out-of-phase.

c) Location of intermediate storage.

2) *Design decisions.*

a) Selection of equipment of standard sizes.

b) Sizing of intermediate storage vessels with standard sizes.

3) *Production planning decisions.*

a) Optimal length of production cycle during which the optimal schedule is executed.

b) Handling of inventory of final product.

4) *Scheduling decisions.*

a) Sequencing of products

The objective is to select the decisions so as to maximize the Net Present Value (NPV) of the process, which is a projection of the profitability of the process during its life span to the present.

Following are the major assumptions that will be made in this problem:

1) All products have identical task networks which are sequential, although some more general cases can be treated.

2) Intermediate storage is considered only as a means to decouple neighboring no-wait subtrains for which only one vessel of intermediate storage is considered.

3) Parallel units in a stage, operating either out-of-phase or in-phase are considered in equal sizes. This assumption is preferred in grassroots design models like the ones presented in this work. In case of retrofit models this assumption can be relaxed.

4) Flexible equipment operation is not considered: A set of parallel equipment is exclusively operated in-phase or out-of-phase for all products. This assumption can easily be relaxed.

5) The same sequence of products in the various subtrains is maintained.

6) Processing times are not dependent on the batch size.

7) Semicontinuous units are not considered.

In this paper an MDLP model will be proposed that can explicitly address the decisions indicated under the above assumptions. The unique feature of this model will be the capability of optimizing the length of the production cycle by accounting for the inventories and the scale of integration among the elements of batch design. The outline of the paper is as follows. A brief review of the literature is first presented. An outline of the synthesis model is then presented in which the major aspects that define the space of alternatives are explained. The importance of anticipating the effect of inventories is highlighted by developing an MILP model for a simplified version of the general problem. Numerical examples are given to backup the claim. Next, a nonconvex MINLP model is given, which incorporates all the issues of the design problem. A novel reformulation scheme is then applied that transforms the nonconvex MINLP to an MELP problem and the equivalence of the two models is established. The solution approach is presented next. Finally the solution of the proposed MILFs is illustrated with several examples and the consistency of these models is verified.

Literature review.

Comprehensive descriptions of the design problem and discussions on the issues that characterize the batch process design problem, have been presented in the literature (Reklaitis, 1990, Rippin, 1993).

Among earlier work, Yeh and Reklaitis (1987) addressed the issues of storage location, allocation of tasks to equipment, parallel units and vessel sizing. They proposed a nonconvex MINLP model but because of computational difficulties a heuristic procedure was used to solve the problem. Modi and Karimi (1989) also developed a heuristic procedure that considers the storage location problem in a multiproduct batch plant. Birewar and Grossmann (1990) proposed a nonconvex MINLP model with which the synthesis, sizing and scheduling issues were integrated in the same model. This model however did not address the issue of intermediate storage sizing and its location, in its general form, inventory handling was not considered, and the equipment sizing problem was solved with continuous instead of standard equipment sizes. Patel et al (1991) proposed a simulated annealing method in which intermediate storage as well as parallel equipment of unequal sizes were considered. Simulated annealing was also the solution method that Tricoire and Malone (1991) used for their proposed model. These authors considered among others the handling of final product inventories which was introduced in the batch process area by Klossner and Rippin (1984). A significant limitation on most previous mathematical programming approaches is the assumption that the sizing problem is solved in a continuous space. This assumption gave rise to nonlinearities and even further to nonconvexities. Voudouris and Grossmann (1992) have shown that the consideration of standard equipment sizes allows the application of novel reformulations schemes and thus remove the need to develop MINLP models in many instances. Instead more robust and efficient MILP models that address the design problem for cases of multiproduct and multipurpose plants have been proposed allowing the scope of mathematical programming models to be expanded. Finally, Shah and Pantelides (1991) and Papageorgaki and Reklaitis (1990) have developed MILP and MINLP models, respectively, which can address particular cases of multipurpose plant design.

Outline of alternatives for **synthesis model.**

Model building for preliminary design is still largely an art depending mainly on the designers intuition and conception of the problem. Under this framework it is not surprising that a variety of models have been proposed for preliminary design of batch processes. This is mainly due to the different concepts attacked by the modelers. In other words the space of alternatives is defined differently by various designers. One example of this is the role of intermediate storage between two no-wait subtrains. One alternative is to use intermediate storage only as a means to decouple the operation of the neighboring subtrains. In this case the storage vessels will accommodate only a limited number of batches and thus they can be of relatively small size. A second alternative is to allow the use of relatively large storage vessels so that it is possible to store all the batches of an intermediate produced in the upstream subtrain during a period of time. This mode of operation allows equipment in the upstream subtrain to be utilized also in the downstream subtrain in a subsequent period. On one hand savings in capital investment are achieved by better utilizing the equipment, but on the other hand this mode also means higher capital investment because larger storage vessels and larger operating costs. Although it might be desirable to exploit both alternatives, to the moment this is impossible since the models generated for both alternatives are radically different to each other. Therefore, a decision has to be made beforehand about the role of the intermediate storage.

Although the ideal model is one that considers all possible alternatives that might influence the design decisions, this model by all likelihood will be impossible to solve. Therefore, a preliminary screening of what is more and what is less important has to be performed, and a clear definition of the restricted space of alternatives has to be given. For this reason, the first step in our design approach is to explicitly state what the space of alternatives is. Overall, the design approach will consist of the steps indicated in Figure 1. The main issues that define the space of alternatives are:

1) Inventory considerations.
2) Allocation of tasks to equipment.
3) Use of intermediate storage.
4) Parallel units per stage.
5) Timing of the production cycle.

To highlight the importance of anticipating the effect of inventories in the design stage, we first address this aspect and illustrate it with an example with fixed topology.

We then discuss the other aspects and derive the comprehensive synthesis model addressed in this paper.

Inventory considerations.

In a batch process two kinds of inventories can be identified. First the inventory held in intermediate storage (also called WIP for work in process) and the final product inventories. WIP is usually relatively small compared to the final product inventories. The level of product inside the inventory vessel can be illustrated by means of the inventory triangles like the one illustrated in Figure 2(b). In a storage vessel there is a incoming stream from the process and an outcoming stream to the market (Figure 2(a)). Even though in batch processes the incoming and outgoing streams are not continuous, when the number of batches is relatively large it can be assumed that they are continuous. It can be seen in Appendix I that the cost of inventory is proportional to the area of the inventory triangle and can be given with a relatively simple equation. When the number of batches is relatively small during a time period P then the assumption of continuity for the incoming and outgoing streams does not hold and the actual inventory cost cannot be linked to the area of the inventory triangle. In this case, though, the cost of inventory is small and the impact it has on the design process is marginal.

Because of the high value of the products of a batch process the way the inventories are handled significantly affects the flowsheet design. This can be understood by means of a small example that is illustrated in Figure 3. Suppose that products A and B are produced. One alternative is to use one single product campaign for each product. This means that in campaign 1 product A is produced and in campaign 2 product B is produced. Another alternative is to produce these 2 product in 4 campaigns. In campaigns 1 and 3 product A is produced, whereas in campaigns 2 and 4 product B is produced. In other words, in the second case the optimal sequence of A-B is repeated twice. We say that the *Production Cycle* in the second case is one half of the design horizon, whereas in the first case it coincides with the design horizon. So the *Production Cycle* is the length of time in which the optimal schedule is executed once. Going back to the example it can be seen that although the inventory levels illustrated by the area of the inventory triangles are lower in the second case, more cleanup time between the products is introduced and thus the actual available time is reduced. In our model this means that the plant has to be overdesigned to meet the demand specifications, leading to increased capital investment. The decision how to resolve this trade-off will be included in our model.

To illustrate the impact of inventories, a small problem given in the literature (Birewar and Grossmann, 1989) is expanded so that inventories are considered in the design.

Incorporating inventories in a batch plant with fixed topology.

Consider the case of a multiproduct batch plant with one equipment per stage. The tasks to be performed in each equipment have already been assigned and no intermediate storage between the equipment is considered. It will be assumed that the plant operates in single product (SPC) campaign mode. The design has to be such that the demands Qi of N products have to be satisfied over a design horizon H. During this time horizon the plant will operate in a cyclic manner such that demands qi = Qi / NC of N products have to be satisfied over a production cycle time P = H / NC , where NC is the number of cycles repeated in the design horizon H. It is clear from the previous equations that,

$$H q_i = P Q_i \qquad \forall i$$

The design of a plant under the new specifications can be treated with the following MINLP model,

$$\max \ NPV \qquad \text{(M1)}$$

s.t.

$$V_j \geq S_{ij} B_i \qquad \forall i, j$$

$$n_i = \frac{q_i}{B_i} \qquad \forall i$$

$$\sum_k NP_{ik} = n_i \qquad \forall i$$

$$\sum_i NP_{ik} = n_k \qquad \forall k$$

$$T_i = (n_i t_{ij} + (\sum_k NP_{ik} SL_{ikj})) \qquad j' = M$$

$$\sum_i (n_i t_{ij} + (\sum_k NP_{ik} SL_{ikj})) \leq P \qquad \forall j$$

$$NP_{ii} = n_i - 1 \qquad \forall i$$

$$H_{qi} = P\ Qi \qquad V\ i$$

$$\frac{H}{P} = NC$$

$$Pc = \sum_j N_j\, \alpha_j\, \hat{V}_j^{\beta_j}$$

$$Oc = X(fii|(P-Ti)) + mintNC$$
i

$$V\}^{\circ} \leq Vj \leq V_j?^{p} \qquad Vj$$

qi, Bi ≥ 0 V i , NP& £ 0 V i, k , P ≥ 0

ni = Integer V i, NC = Integer

The objective variable is the Net Present Value (NPV) which we want to maximize. The Net Present Value is a linear function of the plant cost and the operating costs. It is defined by the following equation,

NPV «-Pc + (R - Oc) (1-tx) (Prcoef)+(Pc/Ny) tx (Prcoef) (1)

where tx is the tax rate, *Ny* the expected life of the plant, *R* is the total revenue from selling the products and is obtained by multiplying the price of every product by the total demand for that product. *Prcoef* is the present value coefficient with which the future profits are projected to the present. This coefficient is defined as

$$Prcoef = <\frac{/(1+in)^{N}y-1}{I\ in\ (1+in)^{N}y}\Big\}$$

where *in* represents the interest rate.
The plant cost Pc can be calculated by the following equation,

$$Pc = \sum_j \alpha_j\, \hat{V}_j^{\beta_j}$$

which is the capital investment required for equipment.

The operating costs Oc are calculated by the expression (See Appendix I),

$$Oc = \sum_i \left(\mu_i \frac{Q_i}{2} (P - T_i) \right) + \text{mint } NC$$

where the first summation is the inventory cost and the second term is the setup cost paid every time the optimal schedule is repeated. NC is the total number of repetitions, *mint* is the cost in $ per repetition and fij is the inventory cost per unit mass of inventory of product i per unit time.

The second and ninth constraints are nonlinear and involve crossproducts, whereas the tenth constraint is nonlinear and nonconvex. The first constraint in model (Ml) assures that the volume Vj of every equipment can accommodate a batch of product i of size Bi where Sy is a proportionality constant called *size factor* through which the simplified mass and energy balances around the process equipment are considered. The second constraint is the definition of number of batches n* produced in a production cycle. The third and fourth constraints are aggregated assignment constraints analogous to the TSP assignment constraints which define a sequence of products (See Birewar and Grossmann, 1989). NPft represents the number of changeovers from product i to product k in the optimal schedule. The fifth constraint is the definition of the time dedicated to the production of product i whereas the sixth constraint is the horizon constraint which makes sure that the production will be satisfied in the production cycle P that is to be optimized. Note that the fifth constraint is defined only for the last stage. In these constraints ty is the processing time of product i in equipment j, and SLay is the forced idle time generated in equipment j when k is produced after i. For the Zero-Wait (ZW) case these slack times can be calculated a priori as was shown by Birewar and Grossmann (1989). The seventh constraint is needed for eliminating single product subcycles. By enforcing the equality in this constraint the SPC policy during a production cycle is considered. The eighth constraint is required to enforce the integrality of NC.

Let us consider that the equipment are available in discrete sizes. The following binary variables are introduced,

$$y_{js} = \begin{cases} 1 & \text{if unit at stage } j \text{ has size } s \\ 0 & \text{otherwise} \end{cases}$$

and

$$r_{sv} = \begin{cases} 1 & \text{if sv cycles are considered} \\ 0 & \text{otherwise} \end{cases}$$

Following the procedure in (Grossmann et al, 1991), the integrality of the number of cycles can be enforced then with the following constraints,

$$H = \sum_{sv} sv\, \hat{P}_{sv}$$

$$\sum_{sv} \hat{P}_{sv} = P \qquad \sum_{sv} r_{sv} = 1$$

$$\hat{P}_{sv} \le H\, r_{sv} \qquad \forall\ sv$$

By combining the first and second constraint in model (Ml) and by reformulating the resulting model as shown in Voudouris and Grossmann (1992), the following model is obtained.

$$\max\ NPV \qquad \textbf{(M2)}$$

$$\text{s.t} \qquad n_i \ge \sum_{s} \left(\frac{q_i S_{ij}}{V_{js}}\right) y_{js} \qquad \forall\ i, j$$

$$H q_i = P Q_i \qquad \forall\ i \qquad (2)$$

$$\sum_{k} NP_{ik} = n_i \qquad \forall\ i \qquad (3)$$

$$\sum_{i} NP_{ik} = n_k \qquad \forall\ k \qquad (4)$$

$$T_j = \left(n_i\, t_{ij} + \left(\sum_{k} NP_{ik}\, SL_{ikj}\right)\right) \qquad \forall\ j = M \qquad (5)$$

$$\sum_{i} \left(n_i\, t_{ij} + \left(\sum_{k} NP_{ik}\, SL_{ikj}\right)\right) \le P \qquad \forall\ j \qquad (6)$$

$$NP_{ii} = n_i - 1 \qquad \forall\ i \qquad (7)$$

$$H = \sum_{sv} sv\, \hat{P}_{sv} \qquad \textbf{(8)}$$

$$\sum_{sv} \hat{P}_{sv} = P \qquad (9)$$

$$\hat{P}_{sv} \le H\, r_{sv} \qquad \forall\, sv \qquad (10)$$

$$Pc = \sum_{j} \sum_{s} \hat{c}_{js} y_{js} \qquad (11)$$

$$Oc = \sum_{i} \left(\mu_i \frac{Q_i}{2} (P - T_i) \right) + \text{mint} \sum_{sv} sv\, r_{sv} \qquad (12)$$

$$\sum_{s} y_{js} = 1 \qquad \forall\, j \qquad (13)$$

$$\sum_{sv} r_{sv} = 1 \qquad (14)$$

qi £ 0 V i , NP& £ 0 V i, k , P £ 0 (15)

rii = Integer V i, y_{js}, r_{sv} = binary (16)

where 6js = aj v?s represents cost of standard vessels and $\hat{v}j_s$ represents standard volume s for equipment j. The only nonlinear constraint is the first one. By substituting qi in the first constraint according to equation (2) and by substituting the crossproduct P yj_s with the nonnegative continuous variable ej_s the following MILP model is obtained,

max NPV (M3)

s.t ⁿi ^ X (^ 7 § ") ᵉjs V i J

$ejs \wedge H\, y_{js}$ V j , s

$$P = \sum_{s} e_{js} \qquad \forall\, j$$

(3)-(16)

Compared to model (Ml), model (M3) has the important feature of explicitly handling discrete equipment sizes and even further, it incorporates linear constraints and a linear objective function. By exploiting the structure of model (M3) we can enhance the computational performance with procedures similar to the ones presented in a previous paper (Voudouris and Grossmann, 1992).

It should be noted that model (M3) addresses a simple case of design of multiproduct plants. The more general case in which parallel units in each stage as well as flexible allocation of tasks to equipment and intermediate storage is considered, will be addressed later in the paper.

From the analysis of the results which are presented in the next section it can be seen that the NPV can be influenced significantly. This shows that inventories have to be explicitly considered in the design stage of a batch process.

Numerical results.

The following numerical example will illustrate the interaction between sizing of vessels, sequencing and product inventory handling as considered in model (M3). The key parameter which is expected to affect the interaction between sizing, sequencing and inventory handling is the cleanup times between products. This parameter is incorporated in the precalculated slacks Slag. If for example, the cleanup times are relatively small, then a schedule with many changeovers between products will lead to a more efficient utilization of the available time horizon. In this case it is expected that the production cycle time will be forced towards small values and the number of production cycles will tend towards higher values. On the other side, if the cleanup times are relatively large then the opposite trend towards large single product campaigns develops in order to get efficient utilization of the design horizon or equivalently to minimize the cost of the plant which is supposed to meet the demand specifications in the preassigned horizon. The last trend though, is in conflict with the minimization of inventory costs. Large single product campaigns mean that high levels of final product inventories should be maintained. To complicate things even more, larger production cycles might be desirable in case the demands are subject to significant uncertainty. In this case the possibility of not being able to deliver a specific order is reduced because the level of inventory is not sufficient to satisfy the need. Building up a significant level of inventory may result to high storage costs but the buffer between market demand and plant production is then larger. Overall it is clear from the previous discussion that all these interactions have to be captured in the proposed model.

Consider initially an instance of a plant consisting of 5 stages operating with ZW policy. The equipment on those stages are available in the following 6 discrete sizes SVj = {5000, 10000, 20000, 40000, 60000 liters }. A total of 4 products are produced. Production data for this example are shown in Table I, the cost data are shown in Table II and the results in Table III. The optimal value for the NPV is $1,451,262. The capital investment is $1,626,900 and the inventory cost is 62,282.58 $/yr. The optimal sequence is A-B-C-D and 5 batches of A, 2 batches of B, 4 batches of C and 8 batches of D have to be produced in each cycle. The length of production cycle is P = 129.03 hrs and overall the optimal schedule has to be repeated 62 times in one year. It is interesting to compare these results with the results obtained if the integrality of the ratio (Horizon/Pc) is not enforced In this case the optimal production cycle time is P = 100.2 hrs, the optimal sequence A-C-D-B and 4 batches of A, 2 of B, 3 of C and 6 of D have to be produced. The NPV is $1,496,337.0 the capital investment is $1,626,900 and the cost of inventory is 48.945 $/yr. These results are not surprising since the second case is a relaxation of the first one. If one were to simply round this selection by forcing the number of cycles to the next higher integer value, that is 79 cycles with P = 101.26 hrs, then the optimal NPV is only $1,249,172. This value is 14% lower than the optimal value calculated if integrality is enforced in model (M3).

In many cases a lower bound in the length of the production cycle may be specified because by enforcing this bound the inventory that is accumulated acts like a safety buffer to uncertain demands. If in the previous instance a lower bound of 400 hrs (maximum of 20 repetitions per year) is considered, then the value of the optimal NPV is $1,026,044 (see second column in Table III). It can be seen that a 29% decrease in the NPV is the price that the designer has to pay for greater flexibility. Although a quantitative treatment of finding the optimal tradeoff is possible, this is not addressed in this work. In the last column of Table in the results obtained when the cleanup times are all increased by 5 hrs, can be seen. As expected the production cycle as well as the volume of the vessels increases. The size of the models and the computational performance are shown in Table IV. It should be noted that in cases 1 and 3 a maximum of 100 repetitions is allowed, and in some cases some extra TSP cycle breaking constraints had to be enforced. The mathematical programs have been modeled with GAMS (Brooke et al, 1988) , and Sciconic (SCICONIC/VM 2.11 , 1991) was used to solve the models.

Allocation of tasks to equipment.

Many batch processes are characterized by a sequential task network. In other words, all the products require the execution of a sequence of processing steps which in the case of multiproduct plants has to be identical for all products. Sequential networks are shown in Figure 4. The processing steps might be tasks like mixing, reaction, crystallization etc. The major characteristics of sequential networks is that there are no diverging arcs from the task network. In case there are diverging arcs, the plant is characterized as a multipurpose rather than multiproduct plant. In this work only simple sequential networks are considered. Later it will be discussed how augmented sequential networks can be treated. A major design decision that will be addressed is how the tasks are assigned to the various equipment. This decision can be illustrated with the bipartite graph shown in Figure 5. This graph represents all possible assignments for a specific problem. The actual assignments are represented by a subgraph of the original graph. The assignment graph can be transferred to the mathematical programming model be using suitable subsets of tasks and equipment. Let t={l, ..., T} be the set of tasks and j={ 1,...,M} be the set of equipment. Then the sets Jt £ j or Tj £ t fully represent the bipartite graph of assignments. For example if tasks 1, 2, 3 can be performed in equipment 2 then T2 ={1,2,3} represent the three arcs connecting tasks 1, 2,3 with equipment 2. The restriction imposed in this work is that the entries of the subsets Tj have to be consecutive. In other words, instead of the full allocation problem, only the problem of merging or splitting of consecutive tasks is addressed. An alternative way to represent the assignment graph is be using the following set of dyads; G = { (j,t): t *e* Jt, V j}.

The selection of the actual assignments in the model is done using the following binary variable.

$$z_{tj} = \begin{cases} 1 & \text{if task t is assigned to equipment j} \\ 0 & \text{otherwise} \end{cases}$$

The domain in which this binary variable is defined is the set G.

Intermediate storage.

The use of intermediate storage is desired in cases that either time or capacity or both kinds of bottlenecks exist in the process. By using intermediate storage the operation of the upstream subtrain is decoupled from the operation of the downstream subtrain. In this way the number of batches of the upstream might differ from the number of batches of the downstream. The potential advantages can be understood with the following example. Suppose the upstream subtrain can produce one batch of a product every 5 hours and the downstream subtrain can process one batch every 1 hour. If intermediate storage is not used then an idle time of 4 hours is imposed downstream. On the other hand if intermediate storage is used, then it is possible to store the batch from the upstream and to feed 5 smaller batches to the downstream from the storage vessel. In this way no idle time is imposed to the downstream vessels.

The various alternatives concerning the use of intermediate storage can again be represented with a bipartite graph similar to the one shown is Figure 6. In this graph the upper part represents possible no-wait subtrains. The name of the subtrains indicates the tasks that will be separated if the intermediate storage vessel between the subtrains is selected. The lower part of the graph indicates the available equipment to be assigned. The generation of the arcs in the graph is done with a simple procedure. This procedure consists of identifying the set of required arcs for all possible combinations of storage vessel existence. The union of those sets of arcs constitutes the final graph. For example in the first phase it is assumed that storage between tasks 1 and 2 is the only one that exists. The arcs a1, a4, a6, a8 are the arcs required to express this. In the second phase it is assumed that all storage vessels exist This is expressed by introducing the extra arcs a7 and a10. So the decision whether to use only the first storage vessel or all the storage vessels, is expressed by arcs a1, a4, a6, a8, a7 and a10. By continuing in the same manner the graph shown in Figure 4 is obtained. For modeling purposes the graph is represented with a subset of the set of subtrains. Let $q = \{1, \ldots, T\}$ be the set of subtrains. The subset $Q_j \subseteq q$ indicates the subtrains in which an equipment might be assigned. The graph can also be represented with the set of dyads $Q = \{(j,q): q \in Q_j, \forall j\}$.

The selection of the actual assignments in the model is done using the following binary variable,

$$w_{jq} = \begin{cases} 1 & \text{if equipment } j \text{ is assigned to subtrain } q \\ 0 & \text{otherwise} \end{cases}$$

The domain in which this binary variable is defined is given by the set Q.

Parallel units.

A production stage in this work is an equipment or a group of equipment in which a task or a number of tasks is assigned according to the graph in Figure 5. The use of parallel equipment in-phase for every production stage is allowed. These equipment are of equal sizes. By using parallel units in phase potential capacity bottlenecks can be treated efficiently, while by using parallel equipment out of phase, potential time bottlenecks can be eliminated. Although the general model can treat parallel units out of phase, the timing constraints are not rigorous as was the case when only in-phase equipment are considered, and therefore a verification step is necessary to identify the feasibility of the schedule.

Tiniing constraints.

In model (M3) given earlier in this paper, constraint (6) is the timing constraint which makes sure that the demand specifications are satisfied during the design horizon. In case that parallel units out-of-phase or merging of tasks is considered, then the slacks Slikj cannot be precalculated and they have to be considered as variables in the model. This will introduce bilinear products NPikSlikj. The resulting model is therefore nonconvex and cannot be convexified using exponential transformations. We therefore propose alternative timing constraints, which consider the product changeover rigorously as is done with constraint (6). Note that by using constraint (7) as an equality only single product campaigns are allowed during a production cycle. In case the number of identical batches is large the product changeovers can be ignored without significant error. In this case the calculation of slacks is irrelevant and the timing constraint can be stated as a function of the cycle time. Since the production cycle can be relatively small compared to the design horizon, it is often the case that only a small number of batches from each product are considered during a period. As is shown in Figure 7 in this case the assumption to ignore product overlapping may introduce significant errors in the model. In Figure 7 it can be seen that when changeovers are ignored and the production cycle is calculated only from the cycle times, then 5 hrs are considered as a feasible Production Cycle length. In case however the changeovers are considered the production cycle has to be larger than 8 hrs. The above example shows that when the number of batches in a production cycle is relatively small, the assumption of ignoring the product changeovers can lead to infeasible schedules. The proposed timing scheme starts by decomposing the production cycle in two parts, the *single product part* and the *changeover part.* The timing constraint is thus expressed with the equation,

$$\sum_{i} CT_i + CP \leq P \qquad (17)$$

The single product part CT_i is the time required for the production of all the batches of product i during the production cycle minus one, whereas the changeover part is the length of time that all stages require to interchange from one batch of every product to another batch of the next product in sequence.

Proposition 1: In case parallel units out-of-phase are not considered then constraint (17) is a rigorous timing constraint.

Proof: See Appendix II.

The main difference of equation (17) with equation (6) is that the nonconvexities can be avoided and a set of linear timing constraints can be proposed as will be show later. When parallel units are considered then equation (17) can still be used but a verification step is required.

MINLP model.

Consider a multiproduct plant with potential equipment j=l,....,M in which N products i=l,...N each one requiring tasks t=l,..,T, have to be produced. The equipment can be assigned in q= 1,..., T subtrains. If two equipment are assigned in consecutive subtrains then a storage vessel exists between those equipment. The meaning of the various variables, if not defined directly, can be found in the nomenclature section. As discussed before, a significant advantage of an MINLP model is the expressive power it has. The expression of various concepts in terms of constraints is much easier when mathematical characteristics like linearity and convexity are not considered. For this reason the proposed design approach distinguishes between an initial step in which a general mathematical programming model (usually a nonconvex MINLP) is developed and a subsequent step where the MINLP is reformulated to simpler but equivalent mathematical programs. The second step might consist of a convexification step (e.g Kocis and Grossmann , 1989) or might go as far as replacing the nonconvex MINLP with an equivalent LP. In this work the nonconvex MINLP will be transformed into an MILP as will be shown in the next paragraphs. Theoretically (Sherali and Adams, 1989 , Lovacz and Shrijver , 1989), it is possible to transform the MILP model to an LP. This step however may require an exponential number of steps and therefore is not useful for practical purposes.

The mathematical program that will be developed will consist of constraints that can characterized qualitatively as shown in Figure 8.

The equipment capacity constraints represent simplified mass and energy balance for the process equipment and ensure that the proper equipment capacity is selected and that the production demands are satisfied- More specifically first we have to ensure that the capacities V_t dedicated to task t can accommodate all the products. This is enforced by the following constraint,

$$V_t \geq S_{it} B_{iq} \sum_j z_j ᵽ q \qquad V i, t, q$$

The sum of the product of the binary variables is one when task t is actually assigned to subtrain q and 0 otherwise. The capacity available for a task t is given by

$$\sum_{j \in J_t} \xi_j \hat{V}_j z_{tj} = V_t \qquad V(j,t) € G$$

where £j is the number of identical parallel units of equipment j operating in-phase. Since for a particular task t* exactly one equipment is assigned, it follows that among the entries of the sum in the left handside of the above equation, only one is nonzero. Therefore, the capacity of that particular task t' is equal to the capacity of the in-phase equipment that are assigned to it. The number of batches for a period P in a subtrain q is defined as follows,

$$nb_{iq} = \wedge - \qquad Vi$$

where qi is the amount produced of product i during a production cycle of length P. This in turn implies that the following constraint must hold,

$$_{qi}H = PQi \qquad Vi$$

The capacity constraints for the storage equipment are also simplified mass balances around the storage vessels. It has been proposed (Modi and Karimi, 1989) that a relatively tight upper bound for the size of the storage vessels is given by the following constraint,

$$\hat{V}_q \geq \hat{S}_{iq}(B_{iq} + B_{iq+1})dq \qquad V i, q < T$$

The binary variable dq is defined as follows,

$$\lambda_q = \begin{cases} 1 & \text{if storage q exists} \\ 0 & \text{otherwise} \end{cases}$$

This expression is valid when semicontinuous units are not considered and gives an upper bound for the volume of the storage vessel that is required to decouple the operation of the neighboring subtrains. Another way to obtain an upper bound for the storage vessels, but not as tight as the previous expression, arc the following constraints,

$$\hat{V}_q \geq 2 \hat{S}_{iq} B_{iq} d_q \qquad \forall i, q < T$$

$$\hat{V}_q \geq 2 \hat{S}_{iq} B_{iq+1} d_q \qquad \forall i, q < T$$

The horizon constraint for each subtrain can be stated as,

$$\sum_i CT_{iq} + CP_q \leq P \qquad \forall q$$

where CP_q is the cycle time of the changeover time for subtrain q and CTi_q is the length of time required for the production of the total number of batches of product i minus one.

The cycle time which is needed in order to satisfy the timing of the operations has to refer to those equipment that actually arc selected In order to be able to calculate the cycle times for every product, we need to introduce the equivalent processing time, etij, which is defined by the following constraint,

$$et_{ij} = \sum_{t \in T_j} (t_{it} Z_{tj}) \qquad \forall i, j$$

or in case the existence of parallel units out of phase is considered,

$$et_{ij} \geq \frac{1}{N_j} \sum_{t \in T_j} (t_{it} Z_{tj}) \qquad \forall i, j$$

where T j is the set of tasks t that can be performed on equipment j and Nj is the number of parallel units out-of-phase of equipment j. The cycle time is then defined as ,

$$T_{Liq} \geq et_{ij} W_{jq} \qquad \forall i, (j, q) \in Q$$

or in case that parallel equipment are considered,

$$Tuq\hat{}\hat{e}tij_{Wjq} \qquad \forall\ i, (j, q) \in Q$$

The amount of time required for the production of the batches of product i in subtrain q minus one batch is,

$$CT_{iq} = (nbi_q - 1)T_{Liq} \qquad \forall\ i, q$$

In order to calculate the term CP_q it is necessary to introduce scheduling constraints since this term is depending on the sequence in which the products are produced. One alternative is to include constraints and variables similar to the TSP problem. We can define the following binary variable,

$$\tilde{x}_{ik} = \begin{cases} 1 & \text{if i is immediately before k} \\ 0 & \text{otherwise} \end{cases}$$

Then the following constraints must be imposed for the optimal sequence (Pekny and Miller, 1991),

$$\sum_{k \neq i} \tilde{x}_{ik} = 1 \qquad \forall\ i$$

$$\sum_{i \neq k} \tilde{x}_{ik} = 1 \qquad \forall\ k$$

$$X \quad Z \quad \tilde{X}fc\text{-}\tilde{X}hh\hat{}lBl-1 \qquad \forall\ B \subset (U N >, h \in \{1,..,N\}/B, 2 \leq |B| \leq N$$

$i \in B\ k \in B/\{i\}$

Note that the TSP problem is solved in the space of products and not in the space of batches which reduces the number of the above constraints significantly. The main drawback of the above set of constraints is the fact that the subtour elimination constraints (third constraint) increase exponentially with respect to the number of products. An alternative way to represent the sequencing constraints is by defining the following binary variables,

$$xz_{ikl} = \begin{cases} 1 \text{ if product i is before product k which is ordered in position 1} \\ 0 \text{ otherwise} \end{cases}$$

and

$$x_{kl} = \begin{cases} 1 \text{ if product k is ordered in position 1} \\ 0 \text{ otherwise} \end{cases}$$

The sequencing constraints are then as follows,

$$\sum_k x_{k1} = 1 \qquad vi$$

$$\sum_l x_{ki} = 1 \qquad V\ k$$

$$\sum_l xz_{ikl} = *ki \qquad V\ i,\ k$$

$$\sum_k xz_{ikl} = Xki\text{-}i \qquad V\ i,\ 1$$

The last constraint is enforced cyclically. This means that for the first entry of 1, 1-1 corresponds to the last entry. It can be seen that the binary variable xziki can be treated a continuous variable even though the integrality is not explicitly enforced (Sahinidis and Grossmann, 1991). The above set of constraints offers the advantage that the constraints are not increased exponentially, but it has the drawback of yielding a worse relaxation compared to the initial TSP constraints (including the subtour elimination constraints) that were presented earlier. Overall in case the problem to be considered is just a TSP problem, the first set is superior since the overall computational requirements are smaller. In case though a TSP problem is just a subproblem, as is the case in this work, then the overall tightness of the model is influenced mainly by other constraints and thus the second set of constraints might be considered. The dilemma that is emerging here cannot be resolved in a general manner. For specific cases one has to choose which set of constraints offers more efficient representation. This is the first of a series of points from which the paths of model development might lead to different final models. CP_q is then defined as,

$$CP_q \geq \sum_i (etij\ w_{jq} + \sum_k ((Slag + Clikj)\ w_{jq} (\sum_1 xz^\wedge))\ ) \qquad V\ (j, q)\ e\ G$$

Where Cl_{ikj} is the cleanup time between products i and k in unit j. The slacks Sl_{ikj} are defined with the following equation.

$$W_{jq} W_{(j+1)q} \left(t_{i(j+1)} + Sl_{ik(j+1)} + Cl_{ik(j+1)} \right) = W_{(j+1)q} W_{jq} \left(t_{ij} + Sl_{ikj} + Cl_{ikj} \right)$$

$$\forall i,k, \quad j \le |J| - 1, \; q \in Q_j$$

The slacks Sl_{ikj} represent the idle time at stage j when i is immediately before k whereas the cleanup times Cl_{ikj} indicate the idle time that is enforced in equipment j when a product changeover from product i to product k occurs. Note that the crossproduct of binary variables $W_{jq} W_{j(q+1)}$ is one if both binary variables are one and zero in any other case. This means that the above equality is activated only when both equipment j and j+1 belong to subtrain q and it is trivially satisfied in any other occasion.

A condition that has to hold for the storage capacity constraints to be valid is that the productivity of the successive subtrains has to be equal. Or in other terms, the production time T_i assigned to product i has to be equal to the production time that is assigned to this product i in every subtrain, this is enforced with the following constraint,

$$T_i = CT_{iq} + \frac{1}{N} CP_q \qquad \forall i, q$$

where N is the number of products. For the above equation to truly represent the time assigned to every product, the following condition must hold,

$$\frac{1}{N} CP_q \ge T_{Li} \qquad \forall i, q$$

This condition is not directly enforced in the model but is used for verification. Since CP_q is depending on the cleanup times between products, the above condition holds in most of the cases when cleanup times are involved. In case no cleanup is required, the cleanup time parameters have still to be nonzero but rather small numbers sufficiently large to ensure the validity of the above condition. This is required for the proper operation of the storage vessels.

The constraints that follow represent the layout and logical constraints. These constraints involve only binary variables and enforce the logical consistency of the binary variables in such a way that a feasible flowsheet is obtained.

First we need to assign every task to exactly one appropriate equipment. This is represented as,

$$\sum_{j \in J_t} z_{tj} = 1 \qquad\qquad V t$$

where Jj is the set of equipment j capable of performing task t.

As noted by Yeh and Reklaitis (1987), we need to introduce cycle breaking constraints which do not allow merging of non consecutive tasks. A logical statement which treats this condition is " If task t is assigned to an equipment j and the next task t^1 is assigned to another equipment j^f then task t" which precedes task t cannot be assigned to equipment j^1 " . This is represented as,

$$z_t j + z_{t\bar{j}} + zt\text{-}j' \le 2 \qquad\qquad V t, j, t^f = t+1, j^f * j, j" < j, f < T$$

Every equipment has to be assigned in at most one subtrain

$$\sum_{q \in Q_j} w j \wedge l \qquad\qquad V j$$

Only consecutive equipment arc allowed to the same subtrain

$$w_{jq} + w_{j^{\cdot} q^{\cdot}} + w_{j^{\cdot} q^{\cdot}} \le 2 \qquad\qquad \forall q, j, j' = j+1, q^f * q, j" < j, j^1 < M$$

The above constraint represents the logical statement:" If equipment j is assigned to a subtrain q and the next equipment j^1 is assigned to another subtrain q' then equipment j" which precedes equipment j cannot be assigned to subtrain q^f ".

If an equipment j is assigned to a subtrain q then a subsequent equipment j* cannot be assigned to a subtrain q^1 that precedes q,

$$Wjq + Wj'q' \le 1 \qquad\qquad V j, q, j^f > j, q^f < q, j^f < M$$

If an equipment j is assigned to a subtrain q then the next equipment j+1 has to be assigned either to the same subtrain or to a subsequent one,

$$Wjq \le w_{Q+}i)q + WQ+i)\, q' \qquad\qquad V j, q, q' > q, q' < N$$

If some equipment j is assigned to subtrain q and some equipment j' is assigned to subtrain q+1 then the storage vessel exists.

$$w_{jq} + w_{j'(q+1)} - 1 \le y_q \qquad \forall j, j' \forall j, q, q < N$$

The NPV is defined in equation (1).The plant cost Pc can be calculated by the following equation,

$$Pc = \sum_j N_j \alpha_j \hat{V}_j^{\beta_j} + \sum_q \gamma_q \tilde{V}_q^{\delta_q}$$

which is the capital investment required for equipment (index j) and storage vessels (index q). The operating costs Oc are calculated by the expression (See Appendix I),

$$Oc = \sum_i \left(\mu_i \frac{Q_i}{2} (P - T_i) \right) + \text{mint } NC$$

where the first summation is the inventory cost and the second term is the sum of the setup costs paid every time the optimal schedule is repeated as was the case with the fixed topology model. In this way the proposed model corresponds to the following MINLP problem,

$$\max NPV \qquad \text{(M4)}$$

s.t

$$Pc = \sum_j N_j \alpha_j \hat{V}_j^{\beta_j} + \sum_q \gamma_q \tilde{V}_q^{\delta_q} \qquad \text{(A.1)}$$

$$Ic = \sum_i \left(\mu_i \frac{Q_i}{2} (P - T_i) \right) + \text{mint } C \qquad \text{(A.2)}$$

$$V_t \ge S_{it} B_{iq} \sum_j z_{tj} w_{jq} \qquad \forall i, t, q \qquad \text{(A.3)}$$

$$\sum_{j \in J_t} S_j \hat{V}_j Z_t \le V_t \qquad \forall (j,t) \in G \qquad \text{(A.4)}$$

$$nb_{iq} = \frac{q_i}{B_{iq}} \qquad \forall i, q \qquad \text{(A.5)}$$

$q i H = Q P$ $\forall i$ (A.6)

$\tilde{V}_q \geq 2 S\hat{i}_q B_{iq} d q$ $\forall i, q$ (A.7)

$\tilde{V}_q \geq 2 \hat{S}_{iq} B_{iq+1} d_q$ $\forall i, q$ (A.8)

$\sum_i CT_{iq} + CP_q \leq P$ $\forall q$ (A.9)

$etij = \sum_{teTj} (titZtj)$ $\forall i, j$ (A.10)

$\hat{et}_{ij} = \frac{1}{N_j} \sum_{t \in T_j} (t_{it} z_{tj})$ $\forall i, j$ (A.10a)

$T_{Liq} \geq \hat{et}_{ij} w_{jq}$ $\forall i, (j, q) \in Q$ (A.11)

$CT_{iq} = (nb_{iq} - 1) Tliq$ $\forall i, q$ (A.12)

$\sum_k Xki = 1$ $\forall I$ (A.13)

$\sum_l xu = 1$ $\forall k$ (A.14)

$\sum_1 xziki = Xki$ $\forall i, k$ (A.15)

$\sum_k xziki = Xki\text{-}i$ $\forall i, 1 < N$ (A.16)

$CT_q \wedge \sum_i (et_a w_{jq} + \sum_k ((Slikj + aikj) w_{jq} (\sum_1 xzi_{kl})))$ $\forall q$ (A.17)

$w_{jq} w_{(j+1)q} (et_{i(j+1)} + Slk_{ik(j+1)} + Cl_{ik}(j+1)) = w_{(j+l)q} w_{jq} (©tj j + Slk_{ik} j + Cl_{ikj})$

$\forall i, k, \quad j \leq |J| - 1, q \in Q j$ (A.18)

$T_i = CT_{iq} + \wedge \cdot CP_q$ $\forall q$ (A.19)

$$\sum_{j \in J_t} z_{tj} = 1 \qquad \forall t \qquad (A.20)$$

$$z_{tj} + z_{t'j'} + z_{t'j''} \leq 2 \qquad \forall\ t, j,\ t'=t+1, j' \neq j,\ j''<j,\ t' < T \qquad (A.21)$$

$$\sum_{q \in Q_j} w_{jq} \leq 1 \qquad \forall j \qquad (A.22)$$

$$w_{jq} + w_{j'q'} + w_{j''q} \leq 2 \qquad \forall\ q, j,\ j'=j+1, q' \neq q,\ j''<j,\ j' < M \qquad (A.23)$$

$$w_{jq} + w_{j'q'} \leq 1 \qquad \forall\ j, q,\ j'>j,\ q'<q,\ j' < M \qquad (A.24)$$

$$w_{jq} \leq w_{(j+1)q} + w_{(j-1)q} \qquad \forall\ j, q,\ q'>q,\ q' < N \qquad (A.25)$$

$$w_{jq} + w_{j(q+1)} \leq d_q \qquad \forall\ j, j\ \forall j, q,\ q<N \qquad (A.26)$$

$$\sum_p t_p = NC \qquad (A.27)$$

integrality constraints for w_{jq}, nbi_q, z_{tj}, x_{jd}, NC

Standard sizes for $\hat{V}_j$, $\tilde{V}_q$

nonnegativity constraints for the remaining variables

Model (M4) is a highly nonlinear MINLP and cannot be convexified with exponential transformations because many constraints are not in posynomial form. This model refers to the case of simple sequential networks. In case augmented sequential networks are considered then the binary variable denoting the existence of a storage vessel in the point where arcs are converging (Figure 2), has to be fixed to one. Even further more general layout constraints have to be addressed as will be shown in a future paper.

Linearization of MINLP model.

The exact linearization scheme is based on introducing the binary variable,

$$y_{jsn} = \begin{cases} 1 & \text{if stage } j \text{ has configuration } n \text{ of parallel equipment of size } s \\ 0 & \text{otherwise} \end{cases}$$

which must satisfy the constraint,

$$\sum_{s}\sum_{n} y_{jsn} = 1 \qquad \forall j \qquad \text{(A.28)}$$

Note that the index n might represent parallel equipment out-of-phase or parallel equipment in phase or both of them. More specifically consider that the maximum number of equipment j operating in-phase that are allowed is 2 and the maximum number of out-of-phase groups allowed is also 2. Then for n=1 only one equipment is considered, for n=2 two equipment in-phase grouped in one group are considered, for n=3 two equipment grouped in two distinct groups operating out-of phase are considered and for n=4 four equipment grouped in two out-of phase groups with the two equipment of each group operating in-phase, are considered. The number of parallel equipment that correspond to the index n is given by two parameters. Namely $parip_n$ represents the number of parallel units in phase, and $parop_n$ represents the out-of-phase groups. The volume of the equipment is given by,

$$\hat{V}_j = \sum_{si}\sum_{n} \hat{v}_{js} y_{jsn} \qquad \forall j$$

$$\sum_{si}\sum_{n} y_{jsn} = 1 \qquad \forall j$$

where $\hat{V}_{js}$ represents discrete size s for equipment j and the index si refers to nonzero sizes. Constraint (A.4) then becomes,

$$\sum_{j \in J_t} \delta_j \hat{V}_j z_{tj} = \sum_{j \in J_t}\sum_{si}\sum_{n} parop_n parip_n \hat{V}_{js} y_{jsn} z_{tj} = V_t \qquad \forall t$$

Similarly to the case 1 reformulation scheme (Grossmann et al, 1991) the inverse of the task capacity can be .written as,

$$\sum_{j \in J_t}\sum_{si}\sum_{n} \frac{y_{jsn}\, z_{tj}}{parip_n \hat{v}_{js}} = \frac{1}{V_t} \qquad \forall t \qquad \text{(T.1)}$$

By substituting (A.6) into (A.5) we get,

$$B_{iq} = \wedge \frac{P}{\underset{H}{} nt>i_q} \qquad Vi,q \qquad (T.la)$$

By substituting Bi_q in equation (A.3) we get,

$$V_t \geq S_{it} \frac{Q_i}{H} \frac{P}{nb_{iq}} \sum_{j \in Jt} X t j z W_{jq} \qquad V i, t, q$$

or with respect to nbi_{qt}

$$nb_{iq} \geq Sit \ § 7\frac{P}{7} - Z \ ^{z}ti^{w}jq \quad {}_{j \in J_t} \qquad V\ i,\ t,\ q \qquad (T.2)$$

By combining (T.2) and (T.I) we get,

$$nb_{iq} \geq \ H \sum_{j \in j} \sum_{S1} \sum_{,,} \frac{y_{jsn}}{parip..} \wedge \ 1P \wedge \ w \ \wedge \quad {}_{j \in J_t} \qquad Vi.t.q$$

It can easily be seen that the above constraint is equivalent to,

$$nb_{iq} \geq \frac{Q_i S_{it}}{H} \sum_{j \in J_t} \sum_{s_1} \sum_{n} \frac{y_{jsn}\ z_{tj}\ w_{jq}}{parip_n \hat{v}_{js}} P \qquad \forall i,t,q \qquad (T.3)$$

In (T.3) the crossproduct P y^ z_tj w^ can be eliminated by using the following variable,

$$h \cdot \ {}^{\circ}qjsnt\text{-}j = \begin{cases} P \text{ if } & sny \text{ and } z_{tj} \text{ and } w_{jq} \text{ are one} \\ & 0 \text{ otherwise} \end{cases} \qquad (C.1)$$

(T.3) then reduces to,

$$n \quad {}_{j \in J_t} \qquad \forall i,t,q \qquad (B.1)$$

In order to satisfy condition (C.I) we have to introduce the following equivalence constraints,

$$\sum_{q \in Q_j} \sum_{t \in T_j} b_{qjsnt} \leq H\,T\,y_{jsn} \qquad \forall j, s, n \qquad (T.4)$$

$$\sum_{q \in Q_j} \sum_{s_i} \sum_{n} b_{qjsnt} \leq H\,T\,z_{tj} \qquad \forall (j, t) \in G \qquad (T.5)$$

$$\sum_{t \in T_{jq}} \sum_{s_i} \sum_{n} b_{qjsnt} \leq n\,H\,w_{jq} \qquad \forall (j, q) \in Q \qquad (B.2)$$

$$\sum_{j \in J_t} \sum_{s_i} \sum_{n} b_{qjsnt} = P \qquad \forall t, q \qquad (B.3)$$

where II is the maximum number of tasks that can be assigned to an equipment j and T is the number of potential subtrains where an equipment might be assigned. The equivalence between (T.4), (T.5), (B.2), (B.3) and condition (C.I) can easily be proven. An alternative set of equivalence constraints for (T.4) and (T.5) can be given if the following pseudobinary variable is first defined.

$$a_{jsnt} = \begin{cases} 1 & \text{if } y_{jsn} \text{ and } z_{tj} \text{ are one} \\ 0 & \text{otherwise} \end{cases}$$

Constraints (T.4) and (T.5) can then be replaced by,

$$\sum_{q \in Q_j} b_{qjsnt} \leq H\,P\,a_{jsnt} \qquad \forall s, n, (j, t) \in G \qquad (B.4)$$

$$\sum_{t \in T_j} a_{jsnt} \leq T\,y_{jsn} \qquad \forall j, s, n \qquad (B.5)$$

$$\sum_{s_i} \sum_{n} a_{jsnt} \leq z_{tj} \qquad \forall (j, t) \in G \qquad (B.6)$$

$$\sum_{j \in J_t} \sum_{s_i} \sum_{n} a_{jsnt} = 1 \qquad \forall t \qquad (B.7)$$

Although the alternative set of constraints is a larger set, it considers explicitly the crossproduct between the y and z variables. This information is required a later point.

Another nonlinearity appears in constraint (A. 10a). We can treat it as follows. First we write Nj as ,

$$\frac{1}{N_j} = \sum_{s} \sum_{n} \frac{y_{jsn}}{parop_n} \qquad \forall j$$

Substituting this into (A. 10a) we get,

$$\widehat{et}_{ij} = \sum_{t \in T_j} \sum_{sl} \sum_{n} \frac{t_{jt}}{paropn_n} (z_{tj}\, y_{jsn}) \qquad \forall\, i \,.\, j \qquad \text{(T.6)}$$

Again we have the crossproduct between z and y which can be treated by the pseudobinary variable $aj_{sn}t$ which was defined earlier.

In case constraints (T.4) and (T.5) were used then the action of constraints (B.4) (B.5), (B.6) is performed by the following constraints,

$$ajsnt \wedge \underset{q \in Q_j}{X}\ {}^{b}qjsnt \qquad \mathbf{V\ S, n, (j, t)\ \in\ G} \qquad \textbf{(T.7)}$$

Note that there are 2 sets of constraints that represent the same concepts but mathematically are very different. The first set (SI) consists of constraints (T.4), (T.5), (B.2), (B.3), (T.7) and (B.7) whereas the second (S2) of constraints (B.2), (B.3), (B.4), (B.5), (B.6) and (B.7). Note that both sets of constraints employ the same number of constraints and variables. Even further, we can prove the following proposition.

Proposition 2: The set of constraints (S2) is an equivalent and tighter representation of the set of constraints (SI).

Proof: See Appendix **in.**

So the second set of constraint is going to be used since for the same number of constraints and variables, it gives a tighter representation. In this case the decision to chose set S2 can be supported with a formal mathematical explanation. In many other points in the model, however, among the many alternative reformulations only one is chosen without formal analysis. Coming back to the model constraint (T.6) can be written as,

$$\widehat{et}_{ij} = \sum_{t \triangleq T_j} \sum_{sf} \sum_{n} \frac{t_{jt}}{paropn}\, a_{jsnt} \qquad \forall\, i\,,\, j \qquad \textbf{(T.8)}$$

substituting this into (A. 10a) we get,

$$CT_{iq} \geq \sum_{t \in T_j} \sum_{sl} \sum_{n} \frac{t_{jt}}{parop_n} a_{jsnt} \; w_{jq} \qquad \forall\, i, (j, q) \in Q$$

and by substituting this into (A.12) we get,

$$CT_{iq} \geq \sum_{t \in T_j} \sum_{sl} \sum_{n} \frac{t_{jt}}{parop_n} a_{jsnt} w_{jq}(nb_{iq}-1) \qquad \forall\, i,(j,q) \in Q \qquad \text{(T.9)}$$

An alternative way to express this constraint is as follows,

$$CT_{iq} \geq \sum_{t \in T_j} \Big(\sum_{sl} \sum_{n} \frac{t_{jt}}{parop_n} a_{jsnt} \; (nb_{iq} - 1)\Big) - H\,(1-W_{jq}) \qquad \forall\, i, (j, q) \in Q \qquad \text{(T.10)}$$

This constraint is the one actually used for reasons that will be discussed later.
The product of $a_{jsnt}\,(nb_{iq}-1)$ is replaced by the following variable,

$$\rho_{iqjsnt} = \begin{cases} (nb_{iq}-1) & \text{if } a_{jsnt} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Hence, constraint (A.12) is written as,

$$CT_{iq} \geq \sum_{t \in T_j} \Big(\sum_{sl} \sum_{n} \frac{t_{jt}}{parop_n} \rho_{iqjsnt}\Big) - H(1-W_{jq}) \qquad \forall\, i,(j,q) \in Q \qquad \text{(B.8)}$$

and the following equivalence constraints have to be enforced,

$$\sum_{i} \sum_{q \in Q} \rho_{iqjsnt} \leq U_i\, a_{jsnt} \qquad \forall\, s, n, (j, t) \in G \qquad \text{(B.9)}$$

$$(nb_{iq}-1) \geq \Big(\sum_{j \in J, sl} \sum_{n} \rho_{iqjsnt}\Big) - Y^{l}_{iq}\Big(P - \sum_{j \in Q, sl} \sum_{n} b_{qjsnt}\Big) \qquad \forall\, i, t, q \qquad \text{(B.10)}$$

$$(nb_{iq} - 1) \leq \Big(\sum_{j \in J, sl} \sum_{n} \rho_{iqjsnt}\Big) + Y^{u}_{iq}\Big(P - \sum_{j \in Q, sl} \sum_{n} b_{qjsnt}\Big) \qquad \forall\, i, t, q \qquad \text{(B.11)}$$

Note that the multiple choice character is not present in constraint (B.8) since the right handside is summated with respect to the tasks t. It is always possible that two or more tasks are assigned to the same equipment j. Because of this loss of the multiple choice characteristic we cannot apply the usual reformulation scheme in constraint (T.9). For this reason, constraint (T.10) was selected. Due to that same loss of multiple choice structure the equality of the nonzero values of piqjsnt with (nbi_q - 1) is enforced through the inequalities (B.IO) and (B.I 1). In these inequalities the domain first is defined over those tasks t that can actually be assigned to a subtrain q. Whether a task is assigned to the subtrain q is indicated by the difference,

$$(P - \sum_{j \in Q_t} \sum_{s1} \sum_{n} b_{qjsnt}) \qquad V t, q$$

This difference is 0 when task t is assigned to q, and P otherwise. In the later case both inequalities (B.9) and (B.IO) become redundant and no equality is enforced. The parameters Y^{u}^ and $Y^{l}_{j}q$ which represent bounds, that have sufficiently large values to ensure redundancy.

Constraint (A. 17) can equivalently be written as,

$$CT_q \geq \underset{i}{X} \; (etij \; + \underset{k}{X} \; ((Slikj+ Cl_{ikj})w_{jq} \; (2 \underset{1}{\;} nan)) \;) - H \; (1\text{-}Wj,) \qquad V j, q \qquad (T.ll)$$

and the equality (A. 18) can be replaced with the following two inequalities,

$$(eti(j+i) + Slki_k(j+i) + Cli_k(j+i)) \geq (etij + SUq_{kj} + Cl^*_{kj}) - W^1 \; (2 - w_{jq} - w_{(j+1)q})$$

$$V j, q \qquad (B.12)$$

$$(eti \; (j+i) + Slki_k \; (J\text{^}D + Cli_k \; Q+i)) \leq (eti \, j + Slki_{kj} + Cli_{kj}) + W^u \; (2 - w_{jq} - w_{(j+1)q})$$

$$V j, q \qquad (B.13)$$

Where W^u and W^1 are bounds that are sufficiently large.

The nonlinear term (Slug +Cliicj) $w_{jq} \underset{l}{£} xz$^ in constraint (T.ll) is eliminated as follows. First we introduce the following variable,

$$A_{ikjq} = \begin{cases} (Slk_{ikj} + Clikj) \text{ if both } Wj_q = 1 \text{ and } \underset{l}{\sum} xzya = 1 \\ 0 \qquad \text{otherwise} \end{cases}$$

In order to satisfy these conditions we need to introduce the following equivalence constraints,

$$]T_q \; Afcjq \leq U \; X_i \; ^xZiki \qquad V\, i, k, j \qquad (B.14)$$

$$\sum_i \sum_k A_{ikjq} \wedge U w_{jq} \qquad V(j,q) \in Q \qquad (B.15)$$

$$Aikjq \; £ \; (Slfcj+Clfcj) \; - U \; (2- £_1 \; xzfc! \; - w_{jq}) \qquad Vi, k_f \; (j, q) \; e \; Q \qquad (B.16)$$

where U has a proper value that will ensure redundancies. In this way (A. 17) can be written as,

$$CP_q \geq \sum_i (et_{ij} + \sum_k (Aikjq)) - U(1-w_{jq}) \qquad V(j,q) e \; Q \qquad (B.17)$$

The nonlinearity in constraint (A.27) is eliminated as in model (M3). First the set sv={ 1, 2, ..., C} is defined where C is the maximum number of production cycles allowed during the design horizon. The following binary variable is defined

$$f_{sv} = \begin{cases} 1 & \text{if sv cycles are considered} \\ 0 & \text{otherwise} \end{cases}$$

It is evident that the following constraint holds,

$$X_{sv} \; ^r sv \; =1 \qquad (B.18)$$

Following the procedure in (Grossmann et al, 1991), the integrality of the number of cycles can be enforced with the following constraints,

$$H = \sum_{sv} sv \, \hat{P}_{sv} \qquad (B.19)$$

$$\sum_{sv} \hat{P}_{sv} = P \qquad (B.20)$$

$$\hat{P}_{sv} \leq H_{rsv} \qquad V sv \qquad (B.21)$$

where Psv represents the crossproduct of P r_{sv}. The size of the storage vessels is also considered to be available in standard sizes. For this reason the following binary variable is defined,

$$sx_{qm} = \begin{cases} 1 & \text{if storage q has size m} \\ 0 & \text{otherwise} \end{cases}$$

It is evident that the following constraint holds,

$$\sum_{m} sx_{qm} = 1 \qquad \text{(B.22)}$$

Note that m=l represents size 0, or in other words that no storage is used.
The volume of a storage vessel is then defined as,

$$\tilde{V}_q = \sum_{ml} \tilde{v}_{qm} \, sx_{qm} \qquad \forall q$$

or equivalently as,

$$\frac{1}{\tilde{V}_q} = \sum_{ml} {}^{v}{}_{qm} \qquad \forall q \qquad \text{(T.12)}$$

Note that mi stands for the nonzero volume sizes.
The batch sizes B_{iq} and $B_{i(q+1)}$ can are given by equation (T.la). Taking into consideration the equation (T.12), (A.7) and (A.8) we get the following linear storage capacity constraints,

$$nb_{iq} \geq \frac{2\,Q_i}{H} \sum_{m} \frac{\hat{S}_{iq}}{\tilde{v}_{qm}} sx_{qm} \, P \qquad \forall i,q \qquad \text{(T.13)}$$

$$nb_{i(q+1)} \geq \frac{2\,Q_i}{H} \sum_{m} \frac{\mathring{S}_{iq}}{\tilde{v}_{qm}} sx_{qm} \, P \qquad \forall i,q \qquad \text{(T.14)}$$

The crossproduct P sx_{qm} is then replaced by the variable $\tilde{P}_{qm}$ and the following constraints are obtained,

$$nb_{iq} \ge \frac{2Q_i}{H}\sum_m \frac{\hat{S}_{iq}}{\tilde{v}_{qm}}\tilde{P}_{qm} \qquad \forall\ i, q \qquad (B.23)$$

$$nb_{i(q+1)} \ge \frac{2Q_i}{H}\sum_m \frac{\hat{S}_{i(q+1)}}{\tilde{v}_{qm}}\tilde{P}_{qm} \qquad \forall\ i,\ q < 4 \qquad (B.24)$$

$$\sum_m \tilde{P}_{qm} = P \qquad (B.25)$$

$$\tilde{P}_{qm} \le sx_{qm} \qquad \forall\ q, m \qquad (B.26)$$

Constraint (A.I) and (A.2) can now be written as,

$$PC = \sum_j \sum_{s \in i} \sum_n \hat{c}_{jsn}\, y_{jsn} + \sum_q \sum_m \widehat{cs}_{qm}\, sx_{qm} \qquad (B.27)$$

$$Ic = \sum_i \mu_i \frac{Q_i}{2}(P - T_i) + mint \sum_{sv} sv\, r_{sv} \qquad (B.28)$$

where $\hat{c}_{jsn} = paripn\ parop_n\ a_j\ \hat{v}_{js}^{\beta_j}$ and $\widehat{cs}_{qm} = \gamma_q\, \tilde{v}_{qm}^{\delta_q}$.

The final MDLP model consists of the following constraints

max NPV (M5)

s.t (B.I)-(B.28)

(A.9), (A.10), (A.13) - (A.16), (A.19) - (A.26)

Nonnegativity constraints

Integrality constraints

Computational considerations.

Model (M5) is a large scale MILP. The solution method that was selected to solve it was Branch and Bound (B&B) (Nemhauser and Wolsey, 1988). This solution algorithm is very robust. The efficiency of the algorithm however is depending significantly one some particular characteristics of the model (e.g tightness of the LP relaxation, dimensionality of the problem), as well as some specific characteristics of the algorithm, (e.g branching rule). Since the proposed MDLP involves binary variables with a multiple choice character, these variables were treated as SOS1 variables. The use of the branching rule that this implies gives significant improvements is computational time as was the case in previous batch processing models (Voudouris and Grossmann, 1992).

The first SOS 1 variable in our model is r_{sv} which is used to indicate the number of Production Cycles in a design horizon. Note that an alternative way to represent the number of Production Cycles is through a binary expansion. This means that the variable NC can be written as,

$$NC = \sum_{p} 2^p \psi_p$$

where ψ_p is an unconstrained block of binary variables. Although the alternative way seems to be superior because the dimensionality of the problem is reduced (considerably fewer binary variables are needed to represent integer numbers), it actually performs worse. That is mainly because in the case of binary expansions the SOS1 structure is not exploited. The other SOS1 variable is yj_{sn}. Note however that constraint (A.28) is a double sum with respect to sets *s* and *n* . In order to retrieve the SOS1 character the two sets have to be merged to another superset *sn.* In this way constraint (A.28) is written as,

$$\sum_{sn} y_{jsn} = 1 \qquad \forall j$$

By doing so a whole series of changes in the domains of some parameters has to be performed. For example the parameter $\hat{v}j_s$ represents size s for equipment j. Since the set s has been merged and no longer exists, the above parameter has to be changed to $\hat{v}_{jsn}$. The parameters $parip_n$ and $parop_n$ are accordingly changed to $parip_{sn}$ and $parop_{sn}$.

A very important characteristic is that model (M5) involves many disjunctive constraints. In these constraints of particular importance is the tightness of the bounds that are used. Significant savings in computational time have been achieved by tightening the bounds for constraints (B.10) and (B.11) as well as for constraints (B.12) and (B.13).

Finally, we applied a *tree decomposition* method which exploits logical conditions inherent in model (M5). Most disjunctive constraints are activated by means of the binary variable w_{jq}. This means that in case these variables are fixed, then many of the undesirable disjunctive constraints are eliminated. The latter means that the relaxation gap of the corresponding LP's in the B&B tree is improved, leading to computational enhancement. Even further the domain of the binary variable w_{jq} can be easily decomposed yielding a relatively small Disjunctive Normal Form (DNF) for this variable. For example the (DNF) for W_{jq} that fully describes the problem shown in Figure 9 is **DNF_W = { (Wi2=l , $W_2$2 = 1 , W_{32}=1)V (Wn=l , $W_2$2 = 1 , W_{32}=l) V (Wi_2=l , $W_2$2 = 1 , W33=l) v (w u = l , W22 = 1»W33=l)}. The first term in the DNF shows that no storage is** used, the second entry shows that only storage vessel 1 is used, the third entry shows that only storage vessel 2 is used, and the fourth entry shows that both storage vessels are used. Thus, by fixing the values of variable W_{jq} to any other 4 terms of DNF_W , four subproblems are generated. Each of those subproblems is considerably easier compared to the full problem, because many disjunctive constraints are eliminated. Even further the optimal solution of each of these subproblems is a lower bound to the solution of the full problem meaning that each subproblem yields an objective function cutoff to the next subproblem. Finally the overall number of nodes that is enumerated when all the subproblems are solved, is going to be in all likelihood significantly smaller. This can be understood by means of the example in Figure 9. When the logical condition that is shown in the top of the figure is exploited by using the tree decomposition method, a total of 6 nodes is enumerated. In contrast when the tree decomposition method is not used then a total of 9 nodes is enumerated. It is possible to apply the tree decomposition method by using in addition to the DNF of variable w, the DNF of variable z. As was mentioned in the literature (Tricoire and Malone, 1991) the entries of this allocation variable can be significantly reduced by using a simple screening procedure. The tree decomposition method can be utilized even further by solving the subproblems in parallel. By doing so the computational requirements are dictated by the largest and most difficult subproblem. In case a sequential approach is adopted, then the sequence of solving the individual tree partitions can be constructed according to the heuristic rules proposed by Yeh and Reklaitis (1987).

Numerical examples and discussion of results.

The application of model (M5) incorporates many significant aspects of batch process design, like availability of units in standard sizes, consideration of product inventories, production cycle optimization, optimal sequencing, location and size of intermediate storage, utilization of parallel units, and allocation of tasks to equipment. It is clear that the representation of the interactions between all the issues mentioned above requires a large number of examples which would make prohibitive the length of this paper. Therefore, examples that illustrate interactions considered as most significant are given. In example 1 it was illustrated how the cleanup times that are required in case of product changeovers, affect the optimal sizing and operation of the process. In this example it is illustrated how decisions on the location of intermediate storage and its size, decisions on the utilization of parallel equipment-in-phase and decisions on merging of tasks, are interacting with each other. The data for example 2 are shown in Table V. In Figure 10 the potential decisions of the synthesis of the process are illustrated. It is shown in this figure that 3 levels are considered. The decisions in every level are going to be made by means of the proper binary variable. In the first level the tasks are assigned to one of the appropriate equipment. In the second level it is decided whether parallel units in-phase are going to be used and how many. Finally in the third level it is decided whether and where intermediate storage is going to be used and how large the vessels should be. If for example the mixer and the reactor are assigned to subtrain 1 and the crystallizer in subtrain 2, then one storage vessel is to be used between the reaction and crystallization steps.

The prices for the products are in all cases 0.2 $/kgr for A, 0.3 $/kgr for B and 0.5 $/kgr for C. For every equipment a cleanup time of 10 hrs is required when a changeover of products is occurring. No penalty is considered for setting up the optimal schedule in subsequent production cycles. The inventory cost per ton of final product is l$/tn/hr for all products. In order to verify whether the model captures the interactions between various issues of synthesis and design it was intended to, 4 cases in total have been considered.

In case 1 only one unit per stage is allowed and thus parallel units in phase arc not considered. Intermediate storage is also considered. In case 2 the above restriction is relaxed and up to 2 equal units operating in phase are allowed in every stage. It can be seen in Figure 11 that the optimal flowsheet is drastically changed compared to the first case. It turns out that the use of intermediate storage in the first case was not dictated because of better time utilization, but rather by significant capacity bottlenecks in the second and third stage. For this reason in case 2 all the equipment belong to the same ZW subtrain, no intermediate storage is used and the capacity bottlenecks are treated by employing parallel units in-phase. An interesting observation is that although the capital investment in the second case is significantly higher, because of the increased throughput of the plant the production cycle can be reduced from 153.8 hrs to 113.2 hrs and thus the inventory cost can be reduced from 40,000 $/yr to 29,437 $/yr. This reduction of inventory cost more than offsets the increase of the capital investment and thus the NPV is increased. In the third case it was assumed that the storage vessel between reaction and crystallization has size factors of 3 lt/(kg of product) for A , 5 for B and 4 for C. In other words the volume requirements for this storage vessel are significantly smaller than the

requirements in the previous cases. As can be seen in Figure 11, this turned out to have also a significant impact on the layout of the flowsheet. As expected a storage vessel between the reaction and crystallization steps is employed. Even more the third stage requires only one equipment since by using the intermediate storage the capacity bottleneck of the third stage was significantly alleviated. Finally, in the fourth case it was considered that the equipment for mixing had a cost coefficient of 450$/lt rather than 250 $/lt as in the previous cases. The outcome of the optimization comes not as a surprise. Since the reactor vessel can perform both the mixing and the reaction tasks, these tasks are merged and assigned to the reactor vessels which now have larger volumes. Note that in this case the number of stages of the plant has decreased from 3 to 2.

As mentioned before, because of the complexity of the model a verification step is required to ensure the consistency of the results. In order to make more clear how the results of the optimization runs are interpreted, a detailed table of the results (Table V) for case 3 is illustrated. In this table under *capacity available,* the maximum batch size of final product that can be accommodated in every particular vessel is shown. Under *batch size* the actual batch size that is proposed by the optimization model is indicated. By comparing the two above entries it can be seen by how much every vessel is actually overdesigned. Under *batch size for storage* twice the largest batch size of the subtrains neighboring a storage vessel is indicated. It can be seen that in case the sum of the batch sizes was used, this would not have had any effect on the size of the storage vessel. There were however many cases in which the overestimation of the size of the storage vessels was rather significant. Under *actual cycle* time and *required time for all the batches minus one,* the cycle time and required time respectively are shown that can be calculated only from the synthesis data (parallel units, merging). It has to be clarified at this point that the above variables in the model might have different values because of the many timing constraints imposed to the model, (e.g equal productivities, integrality of ratio Horizon/Production cycle). For example it can be seen the entries for *cycle time assigned* are larger than the entries for *cycle time required.* That is because in the former idle times imposed to secure equal productivities are considered. Interpretations like these are necessary in order to construct the optimal schedule for case 3 (which is shown in Figure 12). For these reason it is clear why an interpretation and a verification step was considered in Figure 1. Finally the optimal schedule for case 3 can be seen in form of Gantt charts in Figure 12. In the first Gantt chart the schedule was constructed sequentially by properly accommodating the next batch as soon as a vessel becomes available. Under this approach it can be seen that the time required for the third stage exceeds the time allocated for the production cycle making the schedule infeasible. The

schedule thus, has to be modified to decrease the total time requirement of the third equipment as was done in the schedule shown in the second Gantt chart. The schedule of the second Gantt chart is feasible and is the optimal schedule. It has to be noted here that in most of the cases more than one feasible schedules can be obtained. As shown by comparing the two Gantt charts, the utilization of the storage vessel is significantly increased in the second schedule. Thus a proper criterion in selecting among otherwise feasible schedules is to consider the one that offers the best equipment utilization. It is also interesting to note that for product C every single batch in the first subtrain is separated in three batches in the second subtrain.

Summarizing, it can be said the values of the decision variables might not represent exactly what they are supposed to represent, but rather values which will make possible to construct a solution that satisfies the optimality conditions. Thus an interpretation step is required in order to construct the optimal design.

As far as the computational requirement are concerned, an interesting point, is that the size of the model does not increase exponentially when the number of tasks, products or standard sizes are increased. The reason for this is the aggregation scheme (Grossmann *et al,* 1992) that was proposed in the previous sections. As noted the result of the aggregation is the reduction of the size of the LFs, but the relaxation gap deteriorates significantly making the branch and bound tree significantly larger and the overall computational requirement larger. If the computer memory available is large, then a totally disaggregated model is advisable. From our computational experiments we found out that a totally disaggregated model in which only constraint (B.9) is aggregated with respect to the indexes i and q, offers the best alternative for computational efficiency. Even further the results shown in Table VII indicate runs where the sequential tree decomposition scheme was used. In the last line of this table computational results for case 2 are shown when the MILP was solved without the tree decomposition method. It can be seen that the tree decomposition method offers significant savings in computational time (savings are of a factor larger than 3).

Summary and conclusions.

It has been shown in this work that accounting for the effect of final product inventories at the design stage has a significant impact on the profitability of a batch process. For this reason the above issue has been considered in a comprehensive MINLP model for synthesis and optimization of multiproduct batch plants. The model considers the allocation of tasks to equipment, location of intermediate storage, sizing of equipment and storage vessels, optimal sequencing of products and optimal length of the production cycle. It has also been shown that due to the availability of the equipment in discrete sizes the MINLP problem can be transformed into an MILP using exact linearizations. Alternative representations have been considered for some subsets of constraints and for which their relative tightness can be established. A tree decomposition method has also been outlined that can significantly decrease the computational cost for solving the MILP.

Finally, the results show that the advantage of the proposed model is that it can systematically account for the many complex trade-offs involved in the problem of synthesis, design, production planning and scheduling of multiproduct batch plants.

Acknowledgment

We gratefully acknowledge financial support from the National Science Foundation under Grant CBT-8908735.

Nomenclature.

Indexes.

i Index of products {A,B,C...} with cardinality N.

j Index of stages or potential equipment {1,2,...., M}.

q Index of subtrains and storage vessels {1,2,...., T}.

t Index of processing tasks. {1,2,...., T).

s Index of discrete sizes for processing equipment {1,2,...., nSj}.

m Index of discrete sizes for storage vessels {1,2,..., niriq}.

n Index of number of parallel units {1,2,....., npj}.

sv Index of number of production cycles.

Variables.

Bj_q Batch size for product i in subtrain q.

Clikj Cleanup time required in equipment j when product k follows product i.

CP_q Changeover time for subtrain q.

CTiq Time dedicated for the production of nbi_q-1 batches of product i in subtrain q

H Time horizon in which the demand has to be satisfied,

nbiq Number of batches of product i in subtrain q during a production cycle.

NC Number of production cycles during the design horizon H.

NPjq Number of occurrences of the pair i-k in a MPC schedule during a production cycle.

Oc Operating costs

P Length of production cycle time.

$parip_n$ Number of parallel units in-phase corresponding to index n.

paropn Number of parallel units out-of-phase corresponding to index n.

Pc Capital investment

qi Amount of production for product i during one production cycle,

Qi Market demand for product i.

Sy Size factor of potential equipment j for product i.

Si_q Size factor of storage vessel q for product L

Slfcj Idle time (slack) imposed in equipment j when product k follows product i.

ty Processing time of product i at stage j.

T_u Cycle time in single product campaigns, for product i. T_{Li} = max{ty}.

$T_{\{}$ Length of time which is dedicated to the production of product i.

Vj Volume of a vessel at stage j.

$\tilde{V}_q$ Volume of storage vessel q.

$\hat{V}j_s$ Standard volume of size s for potential equipment j.

$\tilde{v}_{qm}$ Standard volume of size m for storage vessel q.

Greek Letters.

Oj Cost coefficient for equipment j.

pj Cost exponent for equipment j.

y_q Cost coefficient for storage vessel q.

6_q Cost exponent for storage vessel q.

References.

Birewar D.B and Grossmann I.E (1990). Simultaneous Synthesis, Sizing and Scheduling of Multiproduct Batch Plants,//** *Eng. Chem. Res.,* Vol 29, Noll, pp 2242-2251

Brooke, A., Kendrick, D. and Meeraus, A. (1988). GAMS: A User's Guide. Scientific Press, Palo Alto.

Grossmann LE$_f$ Quesada I., Raman R. and Voudouris V.T. (1992). Mixed-Integer Optimization Techniques for the Design and Scheduling of Batch Processes. NATO Advance Study Institute Conference on Batch Process Systems Engineering, Antalya, Tuikey.

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.

Klossner J. and Rippin D.W.T. (1984). Combinatorial Problems in the Design of Multiproduct Batch Plants - Extension to Multiplant and Partly Parallel Operations. AiChE Annual Meeting, San Francisco.

Kocis, G.R. and Grossmann, I.E. (1989). Computational Experience with DICOPT Solving MINLP Problems in Process Synthesis Engineering. *Computers and Chem. Eng.* 13,307-315.

Lovacz, L. and Schrijver, A. (1989). Cones of Matrices and Set Functions and 0-1 Optimization, Report BS-R8925, Centrum voor Wiskunde en Informatics

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

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

Papageorgaki S. and Reklaitis G.V (1990) Optimal Design of Multipurpose Batch plants-1. Problem Formulation, *Ind£ng.ChemJRes.,$_y$Vo* 29, No. 10, pp 2054-2062

Patel A.N., Mah R.S.H. and Karimi LA. (1991). Preliminary design of multiproduct noncontinuous plants using simulted annealing, *Comp & Chem Eng.* 15,451-470

Pekny J.F and Miller D.L. (1991). Exact solution of the No-Wait Flowshop Scheduling Problem with a comparison to heuristic methods, *Comp & Chem. Eng.*, Vol 15, No 11, pp741-748.

Reeve A. Batch Control, the Recipe for Success ? (1992) *Process Engineering* , Vol.73, No.l, 33-34

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

Rippin D.W.T. Batch Process Systems Engineering: A Retrospective and Prospective Review. To appear in *Computers and Chemical Engineering.*

Sahinidis, N.V. and Grossmann, I.E. (1991a). MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines, *Computers and Chem. Eng.*, 15,85-103.

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

Shah N. and Pantelides C.C. , (1991). Optimal Long-Term Campaign Planning and Design of Batch Operations, *Ind. Eng. Chem. Res.*, Vol 30, No. 10, pp 2308-2321

Sherali, H. D. and Adams, W. P. (1989). Hierarchy of relaxations and convex hull characterizations for mixed integer 0-1 programming problems. Technical Report, Virginia Polytechnic Institute.

Tricoire, B and M. Malone (1991). A new approach for the design of multi-product batch processes. AIChE Annual Meeting, Los Angeles.

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

Yeh, N.C. and G.V. Reklaitis (1987). Synthesis and sizing of batch/semi-continuous processes: Single Product Plants. *Comp. & Chem. Eng.*, 11,639-654.

Appendix I

As seen in Figure 2a, for every product produced in a multiproduct batch process there is one inventory vessel. The level of inventory in every vessel during a production cycle can approximately be represented by the inventory triangles illustrated in Figure 2b. As mentioned this approximation is satisfactory when a large number of batches is considered, but for a small number of batches this doesn't hold. In this case a stepwise profile will evolve instead. However, when a small number of batches is considered, the overall cost of the inventory is small and it affects the objective function only marginally. Since usually more than one production cycles are considered during a design horizon, the actual inventory profile is consisting of many inventory triangles placed side by side. The rate of incoming material in one production cycle can be averaged over all the batches and is given by,

$$\hat{r}_i^{in} = \frac{q_i}{T_i}$$

whereas the rate of the outcoming material is

$$\hat{r}_i^{out} = \frac{q_i}{P}$$

At every moment t the level of inventory is

$$I_i(t) = \hat{r}_i^{in} t - \hat{r}_i^{out} t \qquad \text{for} \qquad 0 \le t \le T_i$$

and

$$I_i(t) = (\hat{r}_i^{in} - \hat{r}_i^{out}) T_i - \hat{r}_i^{out} t = I_{max} - \hat{r}_i^{out} t \qquad \text{for} \qquad T_i \le t \le P$$

where $I_{max} = (\hat{r}_i^{in} - \hat{r}_i^{out}) T_i = q_i - \frac{q_i T_i}{P}$ is the maximum inventory level during a period P.
The cost of inventory between time t_1 and time t_2 is given by the integral

$$C = \int_{t_1}^{t_2} \mu_i I_i(t)\, dt$$

thus the cost for $0 \le t \le T_i$ is

$$C_1 = \int_0^{T_i} \mu_i \left(\frac{q_i}{T_i} - \frac{q_i}{P}\right) t\, dt = \mu_i \left(\frac{q_i T_i}{2} - \frac{q_i T_i^2}{2P}\right) = \mu_i \frac{q_i}{2} T_i \left(1 - \frac{T_i}{P}\right)$$

and for $T_i \le t \le P$ is

$$C_2 = \int_{T_2}^{P} \int_{A_j}^{P} \frac{\cdot}{P}\, t\, dt = \mu_i \left(q_i - \frac{q_i T_i}{P} \right) (P - T_i) - \mu_i \left(\frac{q_i P}{2} - \frac{q_i T_i^2}{2P} \right) =$$

$$= \mu_i \frac{q_i}{2} (P - T_i) \left(2 - \frac{2T_i}{P} \right) - \mu_i \frac{q_i}{2} \left(P - \frac{T_i^2}{P} \right) =$$

$$= \mu_i \frac{q_i}{2} (P - T_i) \left(2 - \frac{2T_i}{P} \right) - \mu_i \frac{q_i}{2} (P - T_i) \left(1 + \frac{T_i}{P} \right) =$$

$$= \mu_i \frac{q_i}{2} (P - T_i) \left(1 - \frac{T_i}{P} \right)$$

The total cost of inventory for product i per production cycle is,

$$C_i = C_i + C_i = \mu_i \frac{q_i P}{2} \left(1 - \frac{T_i}{P} \right) = \mu_i \frac{q_i}{2} (P - T_i)$$

A total of NC production cycles are repeated for product i where,

$$NC = \frac{Q_i}{q_i}$$

Thus the total inventory cost during the design horizon for product i is,

$$I_i = NC\, C_i = \mu_i \frac{Q_i}{2} (P - T_i)$$

The total inventory cost for all products is given by,

$$Ic = \sum_i \mu_i \frac{Q_i}{2} (P - T_i)$$

and the total operating costs are given by,

$$Oc = \sum_i \mu_i \frac{Q_i}{2} (P - T_i) + mint\,NC \qquad (A.2)$$

where *mint* is the setup cost per schedule repetition.

Appendix II

Proof of proposition 1.

Note that in this appendix the indexes i, k and 1 refer to individual batches whereas the index p refers to products. As discussed in Birewar and Grossmann (1991) for the case of ZW subtrains with zero cleanup times the following constraint holds,

$$\mathbf{tij + Slikj = t_{k(H)} + Slik(j.i)} \qquad \mathbf{V\ i, k, j > 1} \qquad \mathbf{(D.I)}$$

Note that i and k represent individual batches, and j represents a production stage. This equation is referred to as *changeover constraint.*

The production cycle in a multiproduct line is defined by the maximum time that a stage requires to produce all the batches of all products . This is expressed by the following equation

$$P * (tij + Sli_2j)+(t2j+Sl23j)+...+(tK|+SlN_tlj) \qquad Vj \qquad (D.2)$$

We consider two particular cases

a) When changeovers between batches of the same product are considered.

In this case there is a stage j where the processing time t y is the maximum for all batches i belonging to the same product. This stage is considered as the *bottleneck* stage and its processing time is referred to as *Cycle time* for product i and is noted as Tu. Because the optimization direction is to minimize P and thus to minimize the slacks Slikj it follows from (D.I) 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. For these reasons equation (D.I) can be restated as

$$Ty = ty + Slikj \qquad V\ j,\ (i, k) \text{ belonging to the same product} \qquad (D.3)$$

In equation (D.2) the above term exists nbj -1 times for each product, where nbi is the number of batches of product i. For example for product A the term in equation (D.3) exists two times in equation (D.2). For this reason equation (D.2) can be restated as

$$P \geq \underset{\mathbf{i}}{X}\ CTi + (ty+ Sljkj + ... + ty+ SIKJ) \qquad V\ j \qquad (D.4)$$

where CTi = (nty-l) T^. Note that the second sum in the right handside of (D.4) refers to slacks between batches of different products. So i , k , 1 and 1 belong all in different products.

b) When changeovers between batches of different products are considered.

In this case we will prove that the last term in (D.4) that is defined as,

$$CP = (ty+ S1^) +(tfcj+ Sl_{krj}) +... + (t_{lj}+ Sl_{lij})$$

is constant for all stages j. First we will prove that the above statement is valid for stages j and j+1. Equation (D.I) for stage j+1 can be written as,

$$ti(j+i) + Slfc(j+u = tk j + Sliy \qquad V j, (i, k) \text{ belonging to different products}$$

By solving this for Slikj and replacing it in the definition of CP we get,

$$CP = (Uj+ (t_{i(j}+i) + Sliky+i) - tk j)) +(tfcj+ Slfaj) +... + (ty+ Sliij) \Leftrightarrow$$

$$\Leftrightarrow CP = (ty+ (t_{i(j+}i) + 81*0+1)+ Slkrj)) +... + (tij+ Sliy)$$

by continuing the substitutions and cancellations we get,

$$CP = (t_{i(j+1)}+ Sl_{ik(j+1)}) +(t_{k(j+1)}+ Sl_{kr(j+1)}) +... + (t_{l(j+1)}+ Sl_{li(j+1)})$$

thus the term CP is the same for stages j+1 and j. Since j can be any stage it follows that CP is the same for all stages. We refer to term CP as *changeover* term. Constraint (D.4) can now be written as

$$P \geq \sum_i CT_i + CP \qquad (D.5)$$

In case cleanup times are considered the above approach can easily be extended and the validity of (D.5) verified. As an example in Figure (A.I) the production cycle is 32 hours. This can be decomposed for stage 1 as 32 = (2x4) + (1x4) + (2x5) + (3 + 0 + 2 + 0 + 5 + 0) or for stage 2 as 32=(2x4)+(lx4)+(2x5)+(1.5 + 0.5 + 4 + 1 + 1 + 2) where the last parenthesis represents the changeover term. The second, fourth and sixth number in this parenthesis represent the slacks between products A and B , B and C and C and A respectively.

In case parallel equipment out of phase and merging of tasks are considered then the above approach can again be used but constraint (D.5) is not exact anymore. More specific the changeover term is calculated as if only merging is considered, and the CPi s are calculated with the consideration of both parallel units and merging. In most of the cases this means that the production cycle calculated is an upper bound to the required value. Because of the loss of the exactness however, a verification of the proposed schedule has to be performed.

Appendix **III**

Let the first set of constraints (S1) be defined by the following constraints,

$$\sum_{q \in Q_j} \sum_{t \in T_j} b_{qjsnt} \leq HT\, y_{jsn} \qquad \forall\, j, s, n \qquad (T.4)$$

$$\sum_{q \in Q_j} \sum_{s} \sum_{n} b_{qjsnt} \leq HT\, z_{tj} \qquad \forall\, (j, t) \in G \qquad (T.5)$$

$$a_{jsnt} \leq \sum_{q \in Q_j} b_{qjsnt} \qquad \forall\, s, n, (j, t) \in G \qquad (T.7)$$

and the second set of constraints (S2) by the following set,

$$\sum_{q \in Q_j} b_{qjsnt} \leq HT\, a_{jsnt} \qquad \forall\, s, n, (j, t) \in G \qquad (B.4)$$

$$\sum_{t \in T_j} a_{jsnt} \leq n\, y_{jsn} \qquad \forall\, j, s, n \qquad (B.5)$$

$$\sum_{s} \sum_{n} a_{jsnt} \leq z_{tj} \qquad \forall\, (j, t) \in G \qquad (B.6)$$

Proposition 2: The set of constraints (S2) is an equivalent and tighter representation of the set of constraints (SI).

Proof: Consider constraints (T.7) and take surrogates with respect to the tasks t. This then yields,

$$\sum_{t \in T_j} a_{jsnt} \leq \sum_{t \in T_j} \sum_{q \in Q_j} b_{qjsnt}$$

From (T.4) this constraint implies that,

$$\sum_{t \in T_j} a_{jsnt} \leq n\, HT\, y_{jsn}$$

This constraint is a weaker representation of constraint (B.5).

By taking surrogates of constraint (T.7) with respect to the sets si and n we get,

$$\sum_{s} \sum_{n} a_{jsnt} \leq \sum_{s} \sum_{n} \sum_{q \in Q_j} b_{qjsnt}$$

Because of constraint (T.5) this implies,

$$\sum_{si}\sum_{n} a_{jsnt} \le H z_{tj}$$

This constraint is a weaker representation of constraint (B.6).

Consider now surrogates of constraint (B.4) with respect to the tasks t. This gives,

$$\sum_{t \in T_j}\sum_{q \in Q_j} b_{qjsnt} \le H \sum_{t \in T_j} a_{jsnt}$$

Considering (B.5) this implies,

$$\sum_{t \in T_j}\sum_{q \in Q_j} b_{qjsnt} \le H\, n\, y_{jsn}$$

which is the same as (T.4).

By taking surrogates of constraint (B.4) with respect to the sets si and n we get,

$$\sum_{si}\sum_{n}\sum_{q \in Q_j} b_{qjsnt} \le H \sum_{sl}\sum_{n} a_{jsnt}$$

which combined with (B.6) implies that,

$$\sum_{sl}\sum_{n}\sum_{q \in Q_j} b_{qjsnt} \le H z_{tj}$$

which is the same to constraint (T.5).

In summary, we have that (T.7) through (T.4) and (T.5) implies (B.5) and (B.6) respectively but is weaker to both of them, and that (B.4) through (B.5) and (B.6) implies (T.4) and (T.5) and is the same to both of them. The second set of constraints is therefore an equivalent but tighter representation of the first set of constraints.

Table I. Data for example 1.

	SIziiFAUTUKS(LT/KG)				PKOCbSSING TIME (HRS)				CLEANUP TIMES (HRSO				
	A	B	C	D	A	B	C	D		A	B	C	D
Stgl	7.57	5.41	11.08	7.92	*45*	5.5	3.75	7.25	A	0	0.2	*05*	*2*
Stg2	15.14	10.82	22.16	15.84	*25*	2.5	2.5	*25*	B	0.2	0	*05*	*2*
Stg3	26.49	18.93	15.84	27.72	*15*	1.5	1.5	*15*	C	0.5	0.5	*0*	*05*
Stg4	7.57	5.41	11.08	7.92	3.75	1.5	5.75	8.5	D	2	2	*05*	*0*
Stg5	7.57	5.41	11.08	7.92	0.83	0.83	0.83	0.23					

DEMANDS (kg/yr) A = 400000,B1 = 200000, C = 200000, D = 600000 HORIZON = 8000 hrs

Table EL Economic data for example 1.

	Fixed cost fj	Costcoeffaj	Cost exponent bj	Price of prod. S/kg	Inv. cost ($An/hr)
Stage 1	105,000	650	0.6	A = 0.6	A = l
Stage 2	82,000	550	0.6	B=0.65	B = l
Stage 3	48,000	280	0.6	C = 0.7	C = l
Stage 4	65,000	350	0.6	D = 0.55	D = l
Stage 5	150,000	350	0.6		

Plant expected life slOyrs, taxation rate = 45%, Interest rate = 10 %, Setup cost pet interchange = 0 $

Table IIL Results for example 1.

		Small cleanup times		Large cleanup times
		Optimal Cycle	Bound in Cycle	Optimal Cycle
NPV		$ 1,451,262.0	$ 1,026,044.0	S 660.050.0
Inventory cst		$ 62^82 per year	$ 4,7073 per year	S 148.010 oeryear
Capital Inv.		$ 1,626,900.	$ 1,626,835.6	$ &320,000
Cyde length		129.032 hrs	400.00 brs	307,692 hrs
Equipment sizes	Stgl	10,000 liters	10,000 liters	20,000 liters
	StR2	20,000 liters	20,000 liters	40,000 liters
	Stg3	40,000 liters	40,000 liters	60.000 liters
	Stg4	10,000 liters	10,000 liters	20,000 liters
	Stg5	10,000 liters	10,000 liters	20,000 liters
		5 2 4 8 0 00 0 A B C D	16 26 12 6 0 0 0 0 A D C B	7 11 5 3 0 0 0 0 A D C B

Table IV. Computational results for example 1.

Model size		Computational perfonnance (GAMS 2.25/SCICONIC 2.11 on a VAX 6420)			
Constraints	87		Casel	Case 2	Case3
Variables	188	CPU seconds	21.440	7.81	27.14
0/1 variables	50	Nodes	285	99	424
Nonzeroes	681	Iterations	1253	286	1359

Table V. Data for example 2.

Processing time (t_{ij})				Size factors (S_{ij})			Qi	Stor. size factors	
	eql	eq2	cq3	eql	eq2	eq3	(Kg/yr)	storl	stor2
A	7	3	8	3	9	3	260000	9	9
B	9	2	7	2	10	2	260000	10	10
C	8	3	2	4	3	9	260000	4	9
Cost exp	0.6	0.6	0.6		Storage cost exponent			0.5	0.5
Costcoef	250	250	250		Storage cost coefficien t			350	350
DISCRETE SIZES	Equipment	SV; = {3500,4500,7500}							
	Storage vessels	SV, = {0,4000,10000,20000, 24000 }							
DESIGN HORIZON = 6000 hrs									

Table VL Detailed results for case 3 of example 2.

Vessel volumes		Capacity available (kg)				Total number of batches			Batch size (kg)		
			eql	eq2	eq3		subtrl	subtr2		subtrl	subtr2
eql	7500	A	2500	1666.6	1500	A	174	174	A	1494.2	1494.2
eq2	15000	B	3750	1500	2250	B	174	174	B	1494.2	1494.2
eq3	4500	C	1875	5000	500	C	174	522	C	1494.2	498.1

Batch size for storage (kg)		Capacity needed for storge (It)		Available storage capacity (It)		ActuaUCycle time (hrs)			Required time for total batches minus one (hrs)		
	Storl		Storl		Storl		subtrl	suhbtr2		subtrl	subtr2
A	2988.4	A	8965.2	A	20000	A	7	8	A	14	16
B	2988.4	B	14942	B	20000	B	9	7	B	18	14
C	2988.4	C	11953	C	20000	C	8	2	C	16	16

Assigned cycle time (hrs)			Assigned time for total batches minus one (jrs)			Total assigned time for each product (hrs)			Changeover time (hrs)	
	subtrl	subtr2		subtrl	subtr2		subtrl	subtr2		
A	7	8.408	A	14	16.816	A	32.482	32.482	subtrl	55.448
B	9	10.408	B	18	20.816	B	36.482	36.482	subtr2	47.000
C	8	2352	C	16	18.816	C	34.482	34.482		

Productivity (kg/hrs) A =138 , B = 122.87, C = 130

Total assigned time per subtrain (hrs) Subtrain 1 = 103.448, Subtrain 2 = 103.448

Table VII. Computational results for example 2.

	Constraints	Variables	Discrete vars	CPU time (s)	Nodes	Iterations
Casel	540	468	119	193.66	391	3065
Case 2	636	576	128	306.31	647	5180
Case 3	636	576	128	244.46	516	4830
Case 4	636	576	128	381.54	687	6837
Case2*	636	576	129	103935	1946	20183

* Tree decomposition has not been considered

+ GAMS2.25/SaCONIC2.11/Vax 6420

Figure 1. Proposed design apporach.

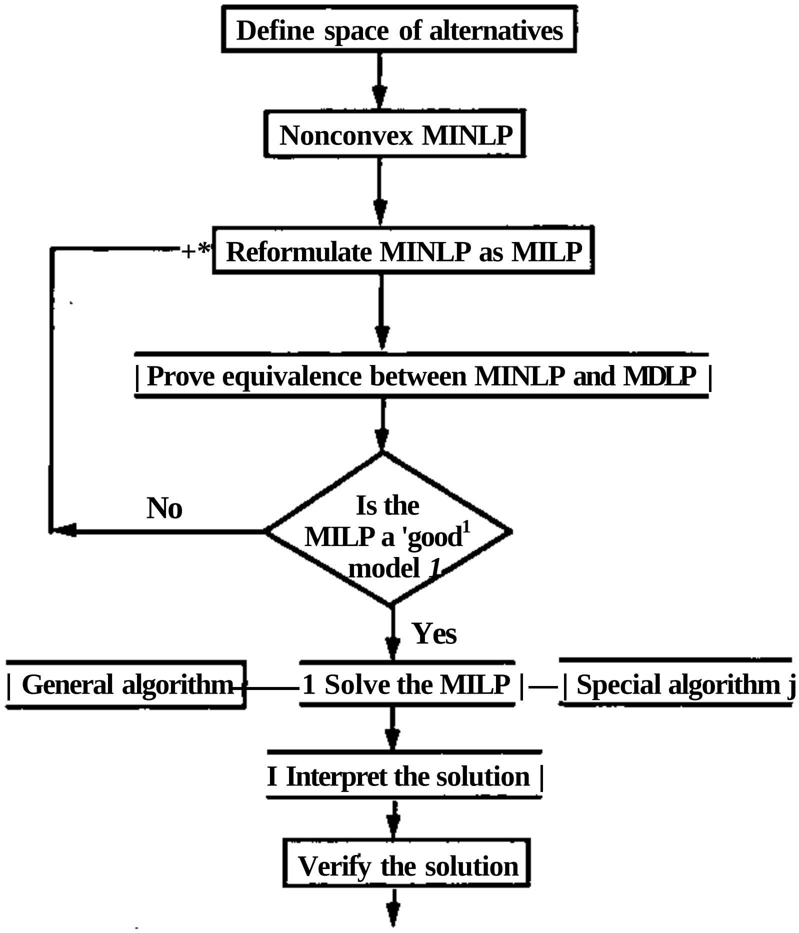
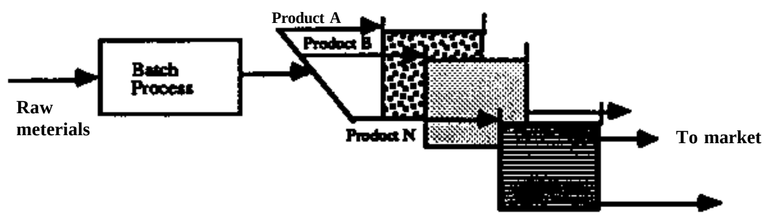
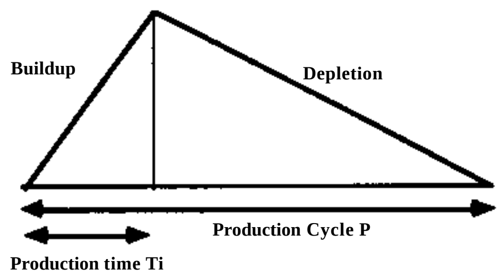


Figure 2. Final product inventory.

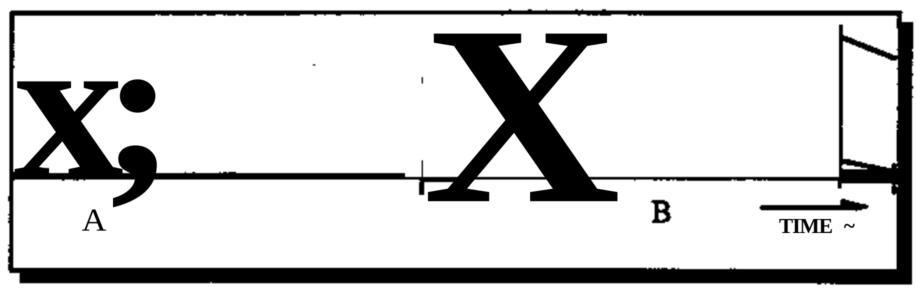


a) Product inventory vessels

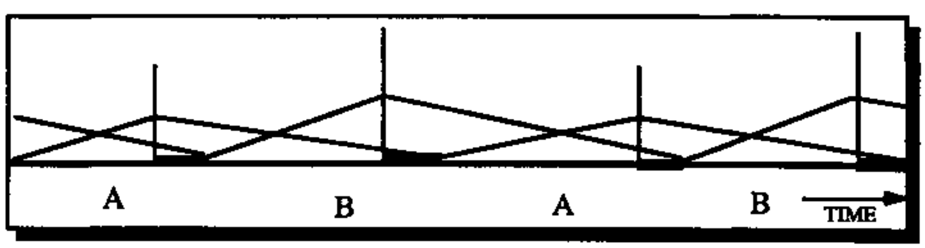


b) Inventory triangle

Figure 3. Production cycle trade-offs.



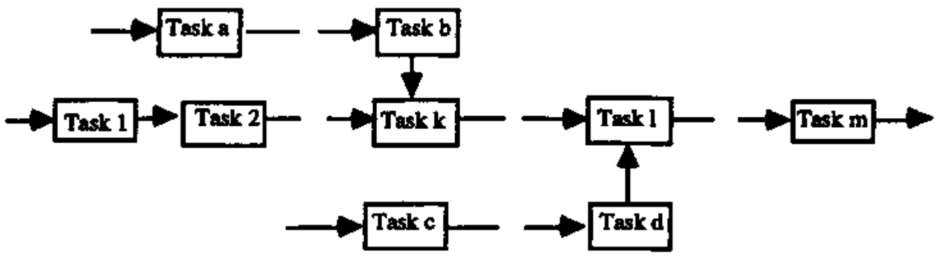
a) One repetition of the production cycle



b) One repetition of the production cycle

Figure 4. Sequential networks.

a) Simple sequential network



b) Augmented sequential network

Figure 5. Task to equipment allocation graph.

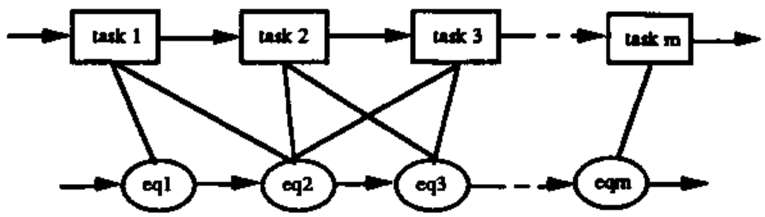


Figure 6. Equipment to subtrain allocation graph.

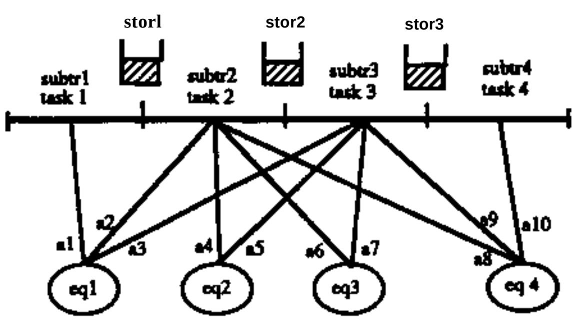


Figure 7. Changeover consideration.

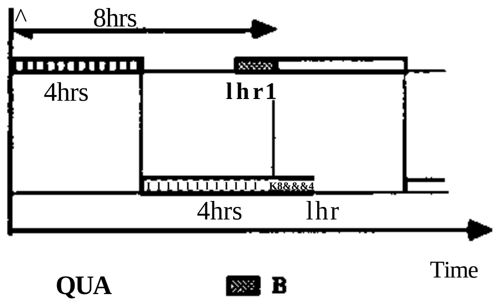


Figure 8. Qualitative representation of mathematical programming model.

0/1 variables: Standard sizes, Task allocation. Parallel units, Sequencing, etc

Integer variables: Number of batches

Continuous variables: Campaign lengths, Production cycle, Inventory costs etc.

Max NPV

s. t **Cost constraints** (NPV, inventories, operating costs)

Equipment capacity constraints (Mass and energy balances)

Storage capacity constraints (Mass balances around storage)

Layout constraints (Flowsheet synthesis)

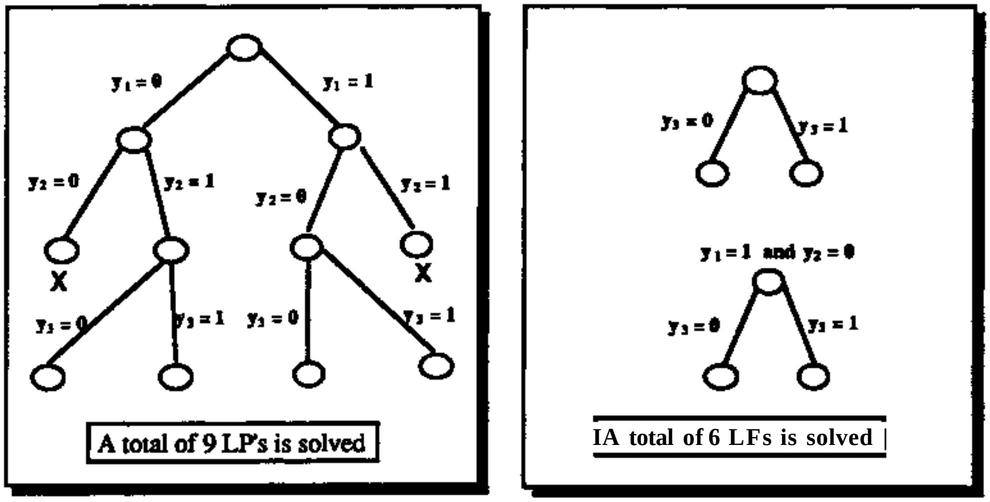
Logical constraints

Timing **constraints**

Scheduling constraints

Figure 9. Tree decomposition method

Consider only (yi = 0 and y_2 = l) or (yi = l M(| y* = °) are feasible



Usual branch and bound algorithm

Tree decomposition method

Figure 10. Levels of decision in example 2

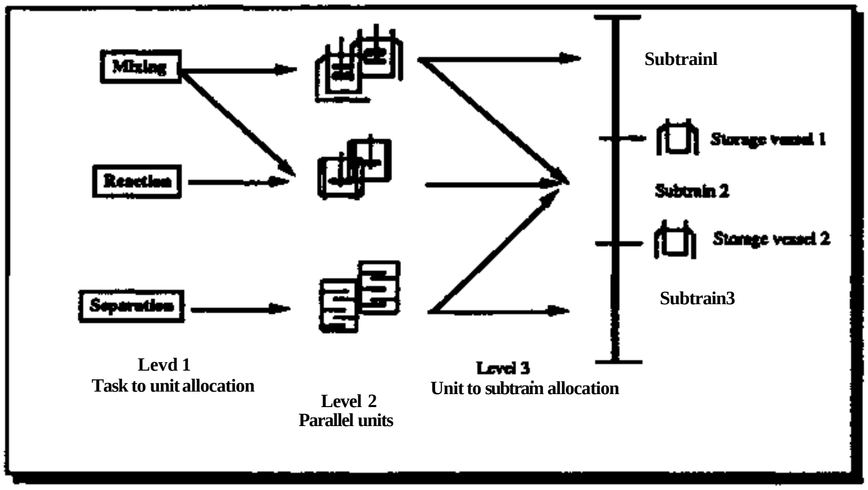


Figure 11. Optimal flowsheets for cases of example 2

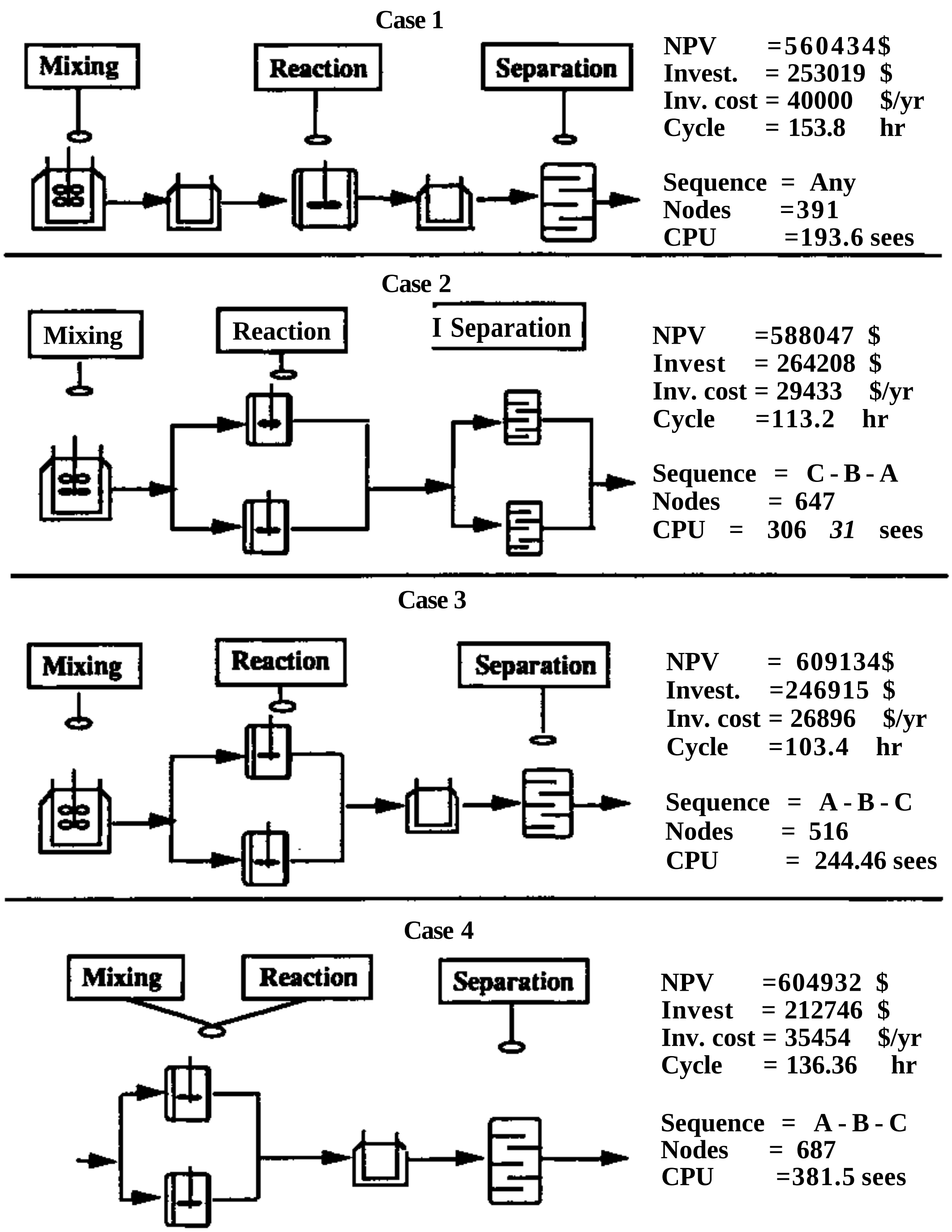


Figure 12. Gantt charts for case 3 of example 2.

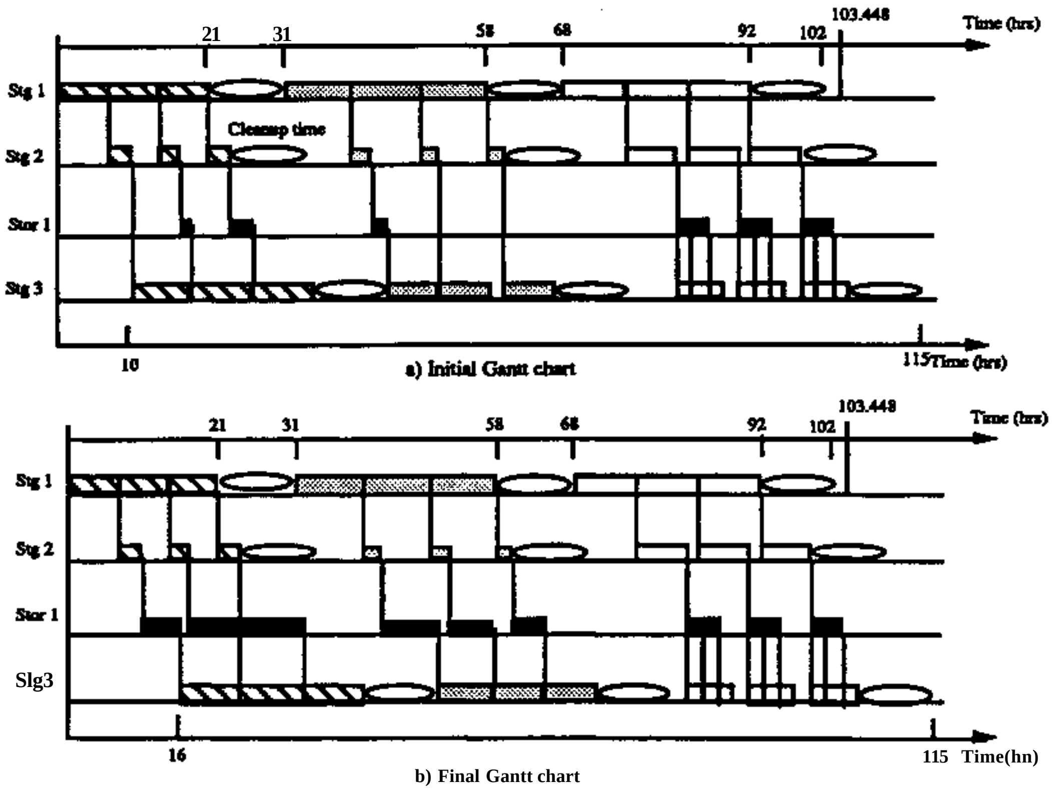


Figure A.I. Production Cycle decomposition.