An Outer-Approximation Method for Multiperiod Design Optimization

D.K. Varvarezos, I.E. Grossmann, L.T. Biegler

EDRC 06-98-91

An Outer-Approximation Method for Multiperiod Design Optimization

Dimitrios K. Varvarezos, Ignacio E. Grossmann and Lorenz T. Biegler
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper addresses the development of an efficient optimization method for convex nonlinear and mixed-integer nonlinear multiperiod design optimization problems. An example of this class of problems that are addressed in this paper are the multiperiod multiproduct batch plant design problems with single product campaign. A multiperiod model is presented for the design and future capacity expansions of such plants. Finally, numerical results are presented drawing comparison with existing general solution methods such as MINOS and SQP for the NLP case, and DICOPT++ for the MINLP case. The proposed method is advantageous in both time efficiency, with savings up to 90%, and robustness.

An Outer-Approximation Method for Multiperiod Design Optimization

Dimitrios K. Varvarezos, Ignacio E. Grossmann and Lorenz T. Biegler
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Introduction

Over the last decade there has been an increased interest in the development of systematic methods for the design of flexible chemical plants (Grossmann and Straub, 1991). The motivation for this comes from the fact that in practice the issue of flexibility is usually introduced by applying empirical overdesign, a practice that does not guarantee optimality or even feasibility over the desirable range of conditions. A major class of flexibility problems, the Multi-Period Design problem (MPD), involves designing plants which are capable of operating under various specified conditions in a sequence of different time periods (Grossmann and Sargent, 1979; Grossmann and Halemane, 1982). Another class deals with the uncertainty involved in some of the design parameters, which is the problem of optimal design under uncertainty (see Grossmann *et al*, 1983 for a review). As has been shown by Grossmann and Sargent (1978) and Halemane and Grossmann (1983), this problem requires the iterative solution of multiperiod design problems which involve Nonlinear Programming (NLP) or more generally Mixed Integer Nonlinear Programming problems (MINLP), where the number of decision variables and constraints can become rather large. It is the major objective of this paper to develop efficient techniques for solving convex versions of these problems which have applications in the design and planning of multiperiod batch plants.

The paper will begin with the statement of the problem and proceed with a short review of decomposition methods and a discussion of the limitations of these techniques as well as the case where no decomposition is performed. We then proceed to the development of an outer approximation based decomposition method for the NLP case, and present its extension for the general MINLP case. A multiperiod model is then presented for the design and future expansions of multiproduct batch plants with single product campaigns. Finally, numerical results are presented, drawing comparisons with solutions when no decomposition is performed.

Problem Statement

For the multiperiod design problem to be addressed in this paper it will be assumed that the plant is subjected to constant operating conditions in N successive time periods. The dynamics of the process will be neglected, as it is considered that the length of the transients is much smaller than the time periods of the successive steady states. The multiperiod design problem can be mathematically formulated as a Nonlinear Programming problem (NLP) when the topology of the process is fixed. More generally, however, its formulation will correspond to a Mixed Integer Nonlinear Programming problem (MINLP) in which the topology of the plant is also subject to optimization for a given superstructure of alternatives.

Consider first the case in which the topology is fixed. The variables are partitioned into two categories: the vector d, of design variables, is associated with the sizing of the units and remains fixed once the design is implemented for all the different periods of operation; the second class are the state and control variables, vectors x_i for each different period i, that can be manipulated in each period so as to meet the production specifications. Thus, the general NLP mathematical formulation becomes:

$$\text{minimize} \quad z = f_0(d) + \sum_{i=1}^{N} f_i(d, x_i) \qquad (1)$$

$$\text{s.t.} \quad \left.\begin{array}{l} h(d, x_i) = 0 \\ g(d, x_i) \le 0 \end{array}\right\} \quad i = 1, \dots N$$

$$r(d) \le 0$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^U\}, \quad i = 1, \dots N$$

$$d \in D = \{d \in R^n \mid d^L \le d \le d^U\}$$

Note that in the above formulation the order of the periods can be arbitrary since the operation of each period is independent of its relative position in the sequence. An important characteristic of this problem can be seen in the matrix representation of Figure 1, where the variables and the constraints form a block diagonal structure. The design variables, d, are the complicating variables and interconnect all different periods of operation.

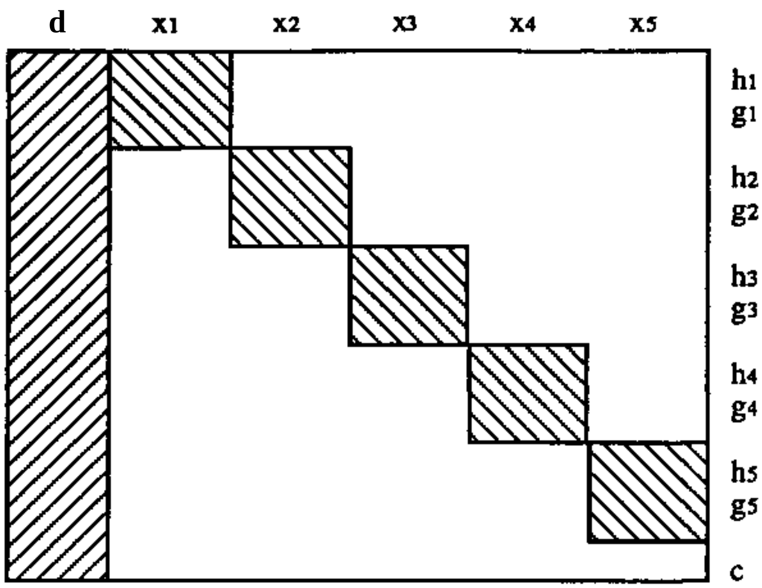

Figure 1. Block diagonal structure in the constraints of model (1)

More generally, if we allow for structural changes to be a subject of optimization, the problem will become a multiperiod MINLP. Here a third class of variables, the binary variables y, are introduced and they are primarily associated with the existence or nonexistence of a unit, or in general with a decision concerning the design of the plant. Structurally they are similar to the design variables, since they also remain fixed once the design is implemented for all the different periods. Assuming that the 0-1 binary variables can be represented linearly in the model, the multiperiod MINLP problem can be formulated as:

$$\text{minimize} \quad z = f_0(d) + \sum_{i=1}^{N} f_i(d, x_i) + c^T y \qquad (2)$$

$$\text{s.t.} \quad \left.\begin{array}{l} h(d, x_i) + Ay = 0 \\ g(d, x_i) + By \le 0 \end{array}\right\} \quad i = 1, \dots N$$

$$r(d) \le 0$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^u\}, \quad i = 1, \dots N$$

$$d \in D = \{d \in R^n \mid d^L \le d \le d^u\},$$

$$y \in Y = \{0,1\}^m$$

For large industrial multiperiod design problems there are two kinds of problems that arise as the size of the problem increases. Firstly, the computational requirements for solving the NLP in (1) or the MINLP in (2) can be very expensive, so that ultimately the solution of the problem may not be achieved in a reasonable amount of time. Secondly, the complexity and the size of the problems can be such that standard algorithms for NLP and MINLP fail. The major reason for this is that the number of variables and constraints increase with the number of periods. Despite the fact that this increase is linear in the number of periods, the computational demand increases in most cases at least quadratically. This behavior has been verified by the solution of two example problems, described in the Appendix. In all cases there is a quadratic/cubic dependence of the solution time to the number of periods, as it is seen in Figure 2.

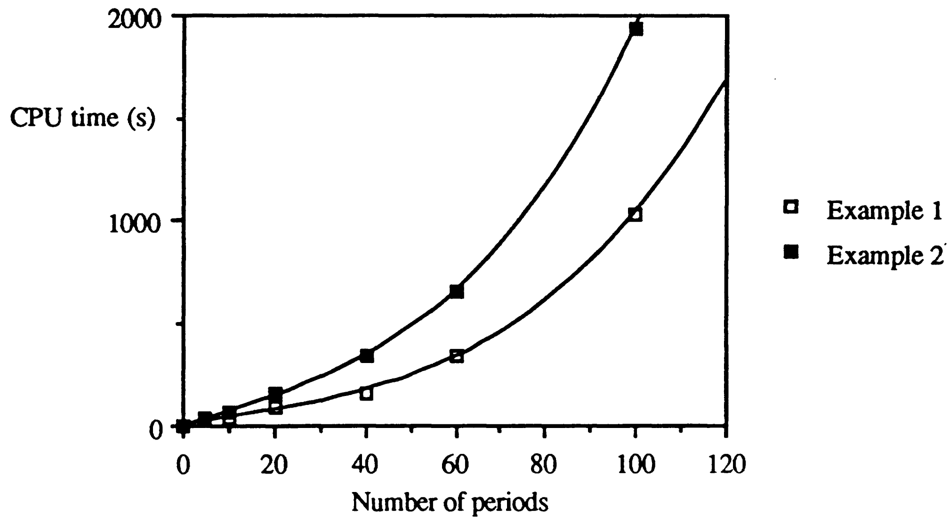

Figure 2. Direct NLP approach for two example problems

Decomposition Strategies

From the above discussion, the need for an alternative solution strategy becomes apparent, especially as the number of time periods increases. The most promising direction is one that exploits the block diagonal structure of the problem through a decomposition strategy. Note that, once the design variables are fixed the subproblems for each period become decoupled, and hence they can be solved independently.

The basic idea behind the NLP decomposition strategy by Grossmann and Halemane (1982), is based on a projection restriction procedure (Grigoriadis, 1979) by which the problem at the level of design variables is reduced into one in which variables are eliminated by using the active constraints at each time period A common characteristic in related methods is that they exploit the problem structure through an algebraic-matrix reformulation scheme. The idea in decomposition techniques that eliminate variables from active sets, is now partially addressed (the user cannot select which variables to eliminate) and implicitly embedded in some of the existing reduced gradient based NLP optimization packages, such as MINOS (Murtagh and Saunders, 1985). Therefore in this work, an alternative decomposition scheme will be developed in order to addresses the problem at a higher algorithmic level.

At this level, an existing method is the Generalized Benders Decomposition by Geoffrion (1972), which can be applied to NLP and MENLP problems. Here, the original problem is partitioned into two subproblems, the primal and the master, and the solution is found by iterating between them. The primal problem is the original problem in which the complicating variables, the design variables d (and the binary variables in the MINLP case) are fixed, and therefore the optimization is done with respect to the *x{* variables. The master problem is a dual problem in which the optimization is done with respect to the complicating variables, subject to approximations obtained from the Lagrangian of the primal problem. In each iteration the master problem, due to its relaxed form, predicts a lower bound to the objective function that increases monotonically with the number of iterations while the primal problem provides an upper bound to the objective. Generalized Benders decomposition was initially explored in this work but the results were not very promising. Firstly, the master problem is an NLP subproblem of increasing size, and secondly a large number of iterations between subproblems and the master problem may be required.

Proposed Method

A different solution alternative is motivated from the ideas of the cutting plane method for convex NLP's (Kelley, 1960) and the Outer Approximation method for MENLFs (Duran and Grossmann, 1986), which also involves an alternating sequence of primal and master problems. The primal problem is also an NLP with fixed values of the complicating variables, identical to the one in Benders decomposition. The main difference lies in the formulation of the master problem. Here, the information from the primal problem is used to approximate the feasible region by accumulating linear approximations of the constraints of the original problem at the optimal points, xiK in each iteration k of the primal problem. This then provides a polyhedral representation of the continuous feasible space of problems (1) and (2) where the optimization is done only with respect to the complicating variables d. This is the basic idea in Outer Approximation (OA). In this paper a slightly different formulation of the master problem has been introduced. The reason for this comes from the fact that OA was developed for MINLP while here, even in the MINLP form of the problem, the difficulties arise mainly from the NLP subproblem. In this formulation, the linearization of the constraints and the objective is being done with respect to both xi and d, leading to a master representation on the full problem space. The present algorithm will make use of cutting planes, based on characterization of convex sets through intersection of supporting hyperplanes. All the analysis that follows is based on convexity assumptions for all the functions involved, namely fi, gi should be convex as well as the relaxed inequality form of hi at the optimum, for i=1, N.

NLP Decomposition Strategy

In the case of the NLP multiperiod problem in (1), the primal and the master subproblem have the following form:

- Primal NLP (optimize xj for fixed d^)

$$\text{minimize} \quad z = \sum_{i=1}^{N} f_i(d^k, x_i) \qquad (3)$$

$$\text{s.t.} \quad \left.\begin{array}{l} h(d^k, x_i) = 0 \\ g(d^k, x_i) \leq 0 \end{array}\right\} \quad i = 1, \ldots N$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \leq x_i \leq x_i^u\}, \quad i = 1, \ldots N$$

Since this subproblem might be infeasible for some values d^k, a more general formulation that allows for a feasible subproblem follows by relaxing the above formulation using a penalty function approach for minimizing the violation of the constraints:

- Primal Feasibility NLP (optimize xi for fixed d^k)

$$\text{minimize} \quad z = \sum_{i=1}^{N} f_i(d^k, x_i) + pu \qquad (3')$$

$$\text{s.t.} \quad \left.\begin{aligned} h(d^k, x_i) - u &\le 0 \\ -h(d^k, x_i) - u &\le 0 \\ g(d^k, x_i) - u &\le 0 \end{aligned}\right\} \quad i = 1, \ldots N$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^u\}, \quad i = 1, \ldots N$$
$$u \ge 0, \; u \in R^1$$

where u is a scalar variable and p is a large positive number. The value of p should be greater than the maximum of the absolute value of the Lagrangian multipliers of the constraints at the optimum point. This is in order to guarantee optimality for (3'), according to the concept of the exact penalty functions (Han and Mangasarian, 1979).

- **Master problem OA/LP (optimize xi and d for K approximation points)**

$$\text{minimize} \quad \alpha \qquad (4)$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \left[f_i(x_i^k, d^k) + \nabla_x f_i(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d f_i(d^k)^T (d - d^k) \right]$$
$$+ f_0(d^k) + \nabla_d f_0(d^k)^T (d - d^k) \le \alpha \qquad k = 1, \ldots K$$

$$\left.\begin{aligned} T_i \{h(x_i^*, d^k) + \nabla_x h(x_i^*, d^k)^T (x_i - x_i^*) + \nabla_d h(x_i^k, d^k)^T (d - d^k)\} &\le 0 \\ g(x_i^*, d^k) + \nabla_x g(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d g(x_i^k, d^k)^T (d - d^k) & \end{aligned}\right\} \quad \begin{aligned} i &= 1, \ldots N \\ k &= 1, \ldots K \end{aligned}$$

$$r(d^k) + \nabla_d r(d^k)^T (d - d^k) \le 0$$

$$x_i \in X = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^U\},$$
$$d \in D = \{d \in R^n \mid d^L \le d \le d^u\}, \quad \alpha \in R^1$$

In the above master problem the equations of the original problem (1) are relaxed as inequalities following the equality relaxation scheme by Kocis and Grossmann (1987), where Ti is a diagonal matrix with elements tjj = sign (λj) and λj the Lagrangian multiplier of the j-th equation.

In the proposed decomposition scheme the primal problem provides an upper bound for the original objective function (provided that it is feasible), while the master problem predicts a monotonically increasing lower bound to the original objective, if convexity conditions hold. The convergence criterion in this case, will be the difference between the upper and the lower bound in each iteration; the two values will be within a small tolerance ε after a finite number of iterations.

The most important feature of the LP master formulation is the fact that it forms a linearized representation of the original NLP (1), in the space of both the d and x_i variables. This would suggest that successive solution of the LP master problem in which new linearizations are generated and added to the master problem, can help to obtain a better representation of the original NLP problem. In this way the solution of NLP subproblems can be avoided, at least at the initial stages of the algorithm. In particular, it is proposed that for a specified inner convergence tolerance ζ, the solution of the master problem should be satisfied before resorting to the solution of a primal NLP. Clearly, if a sufficiently small tolerance is specified the algorithm would reduce to a pure cutting plane without requiring any primal NLP solution.

Another step towards the development of a more efficient algorithm is the elimination of redundant constraints. This idea exploits the fact that convergence can be guaranteed even by dropping the inactive constraints after some iterations; this results in a reduction of the time requirements, proportional to the size of the problem and the number of inactive constraints at the optimum. Particularly, it has been proved (Eaves and Zangwill, 1971), that this constraint rejection scheme does not affect the convergence properties of a general cutting plane method, provided that certain conditions for dropping the constraints, related to a specific improvement in the objective function with respect to a separator function, hold. However, this proof holds without the above provisions when the objective of the master problem is nonlinear and hence the master problem is solved as a linearly constrained NLP (Topkis, 1982).

The proposed Outer Approximation / Repetitive Linear Programming algorithm (OA/RLP) can now be formally stated, assuming that convexity conditions hold in the multiperiod NLP problem (1). The main steps in the proposed OA/RLP algorithm are as follows:

Step 1. Select vecjprs xj^1 and d^1. Set K=l, set the upper bound z^u = °o and the lower bound z^l M™ = - oo. Select tolerances e, £.

Step 2. Set $z^{l\,old} = z^{l\,new}$. Set up and solve the master LP problem (4). Set new lower bound $z^{l\,new}$ = a. If I z^u - $z^{l\,new}$ I < e convergence is achieved, STOP. If I $z^{l\,new}$ - z^l old | ≤ £ then go to step 3; else set K=K+1 and repeat step 2.

Step 3. Solve primal NLP (3^f). If the solution is feasible for the original problem (u=0) set z^K = z ; else z^K = oo. Set upper bound z" = min (z^K, z^u). If I z" - $z^{l\,new}$ l< e convergence is achieved, STOP; else set K=K+1 and return to step 2.

An additional overall convergence criterion can been introduced in step 2 of the above implementation. This involves an e satisfaction of the constraints. Convergence is achieved if I h(xi^, d^) I< e and g(xi^, d^k) < e', where e and e^1 are vectors with elements small positive numbers. The overall convergence parameters e, e^1 and e^M, and the inner convergence parameter £ can be chosen as a small fraction of the current bounds and the value of the individual constraints. The actual number of the NLP calls could be as low as zero, if a small value is used for the inner tolerance £. In some cases it is advisable not to use a very small value. The reason is that the convergence of the LP's might be slow close to the optimum. In this case the primal NLP solution can accelerate convergence in the final steps. Typical values for Ç, and e used in this work were in the range of 0.1 - 0.01 and 0.01 - 0.001, respectively.

From an algorithmic standpoint, the constraint rejection scheme is applied after the solution of each master problem in step 2. If the difference between the value of the objective in the current and in the previous iteration is greater or equal to the value of a nonnegative separator function 8(xj^k, d^k) at the current point, that is $z^k \geq z^{k-1}$ + 8(xj^k, d^k), the inactive constraints from previous iterations are dropped. Possible choice for separator functions are g(xj, d) or g^2(xj, d) for scalar functions g, as it is suggested by Eaves and Zangwill (1971). In this algorithm, the separator function was chosen to be the maximum of the vector of inequality constraints, 8(xi, d) = maxj {gj(xj, d)}.

The above method converges to the optimum if the original NLP problem (1) has a finite optimal solution and its objective function and the constraints are differentiable and convex. We can always assume in (1) that the objective function is linear by introducing an additional variable a; also based on the premise that all the equations can be relaxed as convex inequalities, (1) can be equivalently written as:

$$\text{minimize a} \qquad (I^1)$$

s.t. $z(d,Xi)-oc \leq 0$

$h(d,_{Xi}) \leq 0$ i = l,...N

$g(d, Xi) \leq 0$

$r(d) \leq 0$

and the inequality constraints of $\{I^1)^{can}$ be collectively denoted as g. Since we accumulate constraints in the master problem (4) the objective function of the master problem forms a nondecreasing sequence a^k that approaches the optimum from below, as shown in the following Lemma:

Lemma 1: The value of the objective function a of the master problem (4) forms a nondecreasing sequence oc^k, k = 1,... K, which is always less or equal to the objective a* of the transformed original problem (I^1).

Proof: (i) We define the feasible region of problem (V) and (4) (at the K-th iteration) as: $F=\{x_{if}$ d/gfa, d) <0, xi e X_{i9} i=l, ... N,de D) and $S\wedge=\{x_{if}$ d / g(xfi, d^k) + VgT($_{Xi}{}^k$, d^k) A($x_l{}^{Jc}$, d^k) $\leq$ 0 for k=l, ... K, JC/ e X_{if} i=7, ... N,de D), respectively. In both cases g collectively denotes all the constraints. From the convexity assumptions for all the involved functions we have:

$$g(xfi, d^k) + VgT(xfr\ d^*)\ A(xt, d\wedge \leq g(x_{if}\ d)$$

therefore, from the definition of the two sets we have F $\wedge S^K$ and S^k 3 S^k+^J, k=l, ... K-7.

(ii) Since the constraints in the master problem (4) are accumulated, at iteration K we have forallk<K:

$$z(xi^k, d^k) + VzT(_{Xi}{}^k, d^k)\ A(x£, d^k) \leq a^K \qquad \text{for } k=l, ..., K$$

where A(x£, d^k) is the vector of the differences Xg - xfi and d - d^k.

Since we minimize a:

$$oF = max\ \{z(xfi, d^k) + VzTfa^*, d^k)\ A(x£, d^k)j$$

Whichthenimplies:

$$\min_{(x_{it}d)\in S^k} <x^k \leq \min_{(xi,d)\in S^{k+1}} a^{k+1} \leq \min_{(xi,d)eF} a^* \qquad k = 1,... K\text{-}l$$

Therefore at, k=l,... K is a nondecreasing sequence and oft $\leq$ a.*

The convergence of this method derives from the following result (Minoux, 1986):

Theorem 1: If the functions g are convex and continuously differentiate, and if (l^f) has a finite optimal solution, then any cluster point of the sequence {xi^k, d^k}, generated by the above method is an optimal solution of problem *(V).*

Proof. If problem (1') has a finite optimal solution then after some step K the sequence of points (xfi, d^k) k=l,..., K is contained in a bounded and hence compact set N.
Let {x}, d^l}\€i (Nz^L) be a subsequence which converges to (x{, d).
Consider the subsequence {x£, d^l}i€j (Lz>T) of points for which a cutting plane, with respect to the active constraints, is generated. If at each step we add cutting planes for all the active constraints, then we notice that either g(x_t^J, S) ≤0 after some step I ≥lo on, or the subsequence {x}, d^l}i€jis infinite.
In the case when (xf, d^l}i€j is infinite, we have for all t'eT (t' > t)

g(xf, d<) + VgT(xf9 df) A(x{, d<) ≤0 =>

=> g(xf, df) <H Vg(Xi<, d<) HII A(x{, df) II

Since H A(xf, d<) // ->OandH Vg(xf, cP) // ->// Vg(xi, d) // it follows that

g(x{, <P) -> g(Xi\ d) ≤0

and hence (x{, d) is a solution of(l').
Now, if(xi, (t) is an optimal solution of(l'), then from Lemma 1 we have at every step a^1 ≤ a*, from which we deduce that a' ≤a*, which shows that (x/, d) is an optimal solution, of the minimization problem (!').*

MINLP Decomposition Strategy

In the MINLP case, the decomposition strategy presented in the previous section could be directly applied to the NLP phase of the OA algorithm by Duran and Grossmann (1986). However, since additional linearizations are generated through the proposed OA/RLP scheme these cuts will be also used in the master problem of OA. More specifically, the original MINLP problem (2) is initially decomposed into the NLP subproblem where the binary variables y are fixed, and the Mixed Integer Linear Programming (MILP) master problem which supplies the new binary vector. The NLP subproblem is solved with the algorithm stated in the previous section. For the MILP master problem the ideas of Outer Approximation will be utilized. The advantage in this modified approach comes from the way that the primal NLP is solved with the OA/RLP approach, and the original OA solution for the master MILP problem (Duran and Grossmann, 1986). In this MINLP algorithm all the information from the different NLP phases is used as additional linearizations to the

MILP master, providing a tighter lower bound and therefore a better representation of the original problem at each iteration. The subproblems and the master problem are as follows:

- Primal problem - Feasibility NLP

    - NLP phase - Feasibility NLP (optimize x_i for fixed d^k and y^k)

$$\text{minimize } z = f_0(d^k) + \sum_{i=1}^{N} f_i(d^k, x_i) + c^T y^k + \rho\, u \tag{5}$$

$$\text{s.t.} \quad \left.\begin{array}{l} h(d^k, x_i) + A y^k = 0 \\ g(d^k, x_i) + B y^k - u \le 0 \end{array}\right\} \quad i = 1, \dots N$$

$$r(d^k) - u \le 0$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^U\}, \quad i = 1, \dots N$$
$$d \in D = \{d \in R^n \mid d^L \le d \le d^U\}$$
$$u \ge 0, \ u \in R^1$$

- LP phase - Feasibility LP (optimize x_i and d for fixed y^k)

$$\text{minimize } z = \alpha + \rho\, u \tag{6}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \{ f_i(x_i^k, d^k) + \nabla_x f_i(x_i^k)^T (x_i - x_i^k) + \nabla_d f_i(d^k)^T (d - d^k) \}$$
$$+ f_0(d^k) + \nabla_d f_0(d^k)^T (d - d^k) + c^T y^k \le \alpha \qquad k = 1, \dots K$$

$$\left.\begin{array}{l} T_i \{h(x_i^k, d^k) + \nabla_x h(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d h(x_i^k, d^k)^T (d - d^k)\} + A y^k - u \le 0 \\ g(x_i^k, d^k) + \nabla_x g(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d g(x_i^k, d^k)^T (d - d^k) + B y^k - u \le 0 \end{array}\right\} \begin{array}{l} i = 1, \dots N \\ k = 1, \dots K \end{array}$$

$$r(d^k) + \nabla_d r(d^k)^T (d - d^k) - u \le 0 \qquad k = 1, \dots, K$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^U\}, \quad i = 1, \dots N$$
$$d \in D = \{d \in R^n \mid d^L \le d \le d^U\}$$
$$u \ge 0, u \in R^1$$

In the above implementation both phases of the NLP subproblem are relaxed by using a penalty function scheme in order to prevent infeasibilities that may occur for some of the binary vectors y^k.

- Master problem - MILP (optimize x_i, d and y)

$$\text{minimize} \quad z = \alpha \qquad (7)$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \left[f_i(x_i^k, d^k) + \nabla_x f_i(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d f_i(x_i^k, d^k)^T (d - d^k) \right]$$
$$+ f_0(d^k) + \nabla_d f_0(d^k)^T (d - d^k) + c^T y \leq \alpha \qquad k = 1, \dots K$$

$$\left.\begin{array}{l} h_i(x_i^k, d^k) + \nabla_x h_i(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d h_i(x_i^k, d^k)^T (d - d^k) + Ay - u \leq 0 \\ g_i(x_i^k, d^k) + \nabla_x g_i(x_i^k, d^k)^T (x_i - x_i^k) + \nabla_d g_i(x_i^k, d^k)^T (d - d^k) + By - u \leq 0 \end{array}\right\} \begin{array}{l} i = 1, \dots N \\ k = 1, \dots K \end{array}$$

$$r(d^k) + \nabla_d r(d^k)^T (d - d^k) \leq 0 \qquad k = 1, \dots K$$

$$\sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq |B^k| - 1$$

$$x_i \in X_i = \{x_i \in R^n \mid x_i^L \leq x_i \leq x_i^U\}, \quad i = 1, \dots N$$
$$d \in D = \{d \in R^n \mid d^L \leq d \leq d^u\},$$
$$y \in Y = \{0,1\}^m$$
$$B^k = \{i \mid y_i = 1\} \quad \text{and} \quad N^k = \{i \mid y_i = 0\}$$

In the above scheme the primal problem provides an upper bound for the original objective function (provided that it is feasible) since it is a restricted form of the original MINLP (2), The master problem provides a monotonically increasing lower bound to the original objective provided that the convexity assumptions hold. The convergence criterion in this case, since the original multiperiod problem is a convex MINLP, will be the crossing of the bounds, provided that after each master iteration an integer cut of the previous point y^k is added to the master problem, as indicated in problem (7).

Another option in initializing the solution procedure, which was actually used in this algorithm, is to solve a relaxed version of the original MINLP (problem 8) through the OA/RLP scheme, in the first iteration, by relaxing the integrality requirements on y. This exploits the efficiency of OA/RLP for big problems, and avoids the problem of having to specify initial values of the binary variables (see also Viswanathan and Grossmann, 1990). This relaxed MINLP subproblem has the form:

$$\text{minimize} \quad z = f_0(d) + \sum_{i=1}^{N} f_i(d, x_i) + c^T y \qquad (8)$$

fc s.t. $h(d,x_i) + Ay = 0$ } $i = 1, \dots N$
$g(d,x_i) + By \leq 0$
$r(d) \leq 0$

$x_i \in X_i = \{x_i \in R^n \mid x_i^L \leq x_i \leq x_i^U\}, \quad i = 1,\dots N$
$d \in D = \{d \in R^n \mid d^L \leq d \leq d^u\},$
$y \in Y_r = \{y \in R^m \mid 0 \leq y \leq 1\}$

The above Outer Approximation / Mixed Integer Repetitive Linear Programming algorithm (OA/MIRLP) can now be formally stated, assuming that all convexity conditions hold The main steps in the algorithm are as follows:

Step 1. Select vectors x_i^1, d^1, y^1 and solve the relaxed MINLP (8) by using the OA/RLP approach. If y is integer, the solution is found, STOP. Otherwise, set K=l, $z^u = \infty$.

Step 2. Set up the MILP master problem (7) and solve to find the integer vector y^K with objective value z^K. Set $z^l = z^K$. If $z^l \geq z^u$ optimum is found. STOP.

Step 3. Solve the NLP subproblem for y^K via the sequence OA/RLP of (5) and (6) to find a point (x_i^K, d^K, y^K) with objective value z^K. If the problem is feasible (u=0) set $z^u = z^K$. If $z^l \geq z^u$ the optimum, z^u, is found, STOP. Otherwise go to step 2.

The convergence criterion is the crossing of the bounds, since the problem is convex and at each iteration an integer cut is added to the master problem. One interesting feature of the above algorithm is that the direct solution of NLP problems in step 3 is not needed. Instead, the NLP subproblems can be solved by a series of LFs, provided that the inner convergence tolerance Ç, is small enough, as indicated previously in the paper. In this scheme, phase one of the NLP subproblem (5) is eliminated.This particular scheme is not limited to multiperiod problems but can be generally applied to any MINLP problem.

The constraint rejection step is also utilized in this implementation, in the form earlier discussed, resulting in smaller size MILP's. It has been noted that this step does not affect the number of major iterations, while it significantly reduces the solution time requirements.

From a theoretical standpoint the above algorithm will converge, provided that the continuous functions are differentiable and convex and there is a finite optimum to the original problem ^2). Since it is guaranteed by Theorem 1 that the individual NLP subproblems will converge to the optimum, and the MILP representation has at least all the linearizations the original OA form has, convergence properties become identical to the ones stated in Duran and Grossmann, (1986).

Remarks

Based on the above OA/RLP decomposition method, there are some alternative schemes that can be proposed. Firstly, the master problem of **OA/RLP** can be formulated in a slightly different way, namely as a linearly constrained NLP, linearization could be done only in the space of the constraints while the objective remains nonlinear. In that case linear convergence can be guaranteed and the inactive constraints can be dropped after each iteration (see Topkis, 1982), without the conditions of a separator function stated in previous section. Secondly, the new point (xi^k, d^k) at which linearization is performed in the repetitive scheme could be found in a slightly different way. Instead of using the solution of the previous master problem, a linear combination can be used based on the current solution point of the master problem and a strict interior point provided by the most recent feasible primal subproblem. This scheme might further improve the speed of convergence.

On the implementation level, on the other hand, the successive LFs can be solved through the dual LP problem, in which case the addition of linearizations at each iteration would be more efficient, since the simplex tableau would be dual feasible and hence fewer pivots would be required.

Application to Design of Batch Processes

As an application of the algorithm to multiperiod design we will consider multiperiod batch plants which produce different products over different time periods. In addition to that, these plants can in general have different production demands over different time horizons. One key problem in the design of these plants, that stems from the nature of their products (pharmaceuticals, food products, etc.), is the changing pattern of the demand over a finite horizon time. Often, demand forecasts for a product are given over the next 1, 2 or 5 years. In order to satisfy the changing demand and minimize the discounted costs, planning for the expansion of production capacity in the initial design stage is of vital importance. Here the issue is to consider the tradeoff between the economies-of-scale savings of large initial capacities versus the cost of installing the capacity before it is needed. The major decisions in capacity expansion problems are the expansion sizes and the expansion times. The discounted cost of all expansions depends on the expansion cost function, the discount rate and the demand growth over time. The expansion cost function is usually concave, exhibiting economies of scale. As pointed out by Luss (1982), the discount rate has a significant impact on the optimal policy not only because of the different opportunity cost of money and inflation, but also because it must reflect reductions in expansion cost due to technological innovations.

The MINLP model, by Grossmann and Sargent (1979), for the optimal design of multiproduct batch plants without provision for potential expansions, involves several processing stages with possible parallel units in each of them. In this work, this model will be extended to include different periods of operations so that different amounts of products can be produced during each period. Although the problem of expansions has been partially addressed through a single-step expansion model (Wellons and Reklaitis, 1989), a more realistic approach will be introduced here. A model will be developed to allow for several expansions to occur (or not) at different times, so that the optimal expansion strategy over a finite horizon can be determined. The above structure can be easily visualized in Figure 3, where a typical prediction of demand for one of the products is presented. In this figure fluctuation of the demands is considered with 20 time periods and a 10 year horizon. The initial installation and expansions are considered with 5 potential expansion periods each with a length of 2 years.

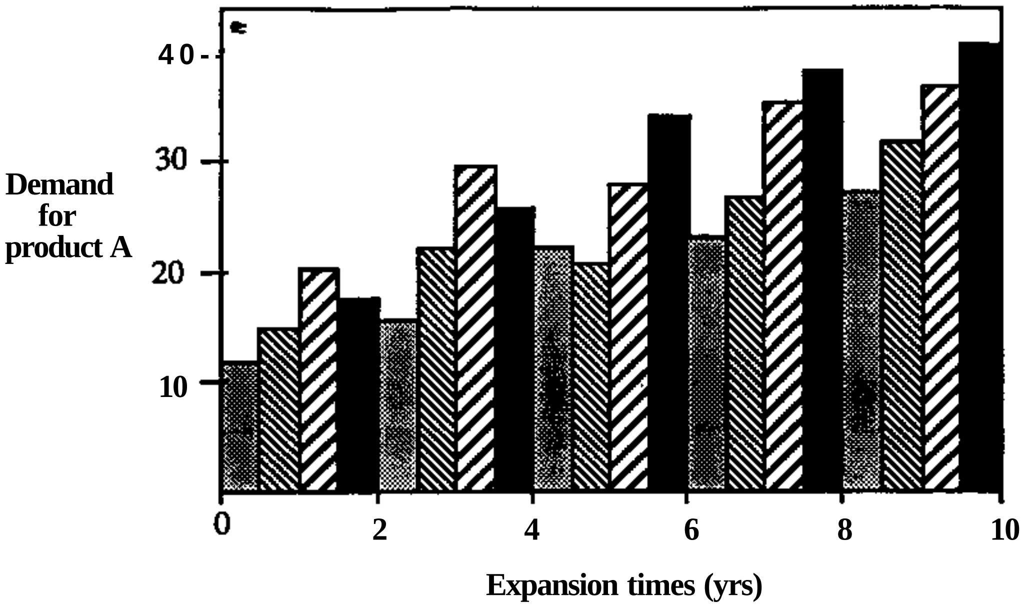

Figure 3. Demand prediction over a ten year horizon

Another extension of these model towards a more realistic approach involves simultaneous optimization with respect to the amount of production. Traditionally, in batch plant models of this type, the objective function involves only the investment cost which is minimized, for fixed amounts of production. The inclusion of a profit function with variable production amounts in the multiperiod model, adds a concave, and non-convexifiable, term in the objective function. This concave term makes the convexification of the model impossible. In this work, a special convex profit function formulation is added to the objective so that the production quantities can be optimized around an initial estimation that is given as a forecast, retaining at the same time all the global solution properties. In this formulation the profit function appears implicitly through a variable representing the additional available time at each period, assuming a uniform increase in production following the predicted pattern.

Two multiperiod multiproduct batch plant models with single product campaign will be developed. In the first model the goal is to optimally design a multiperiod and multiproduct plant with fixed topology consisting of M one-unit stages, producing N products all in T different periods of operation. This leads to the following multiperiod non-convex NLP which, however, can be convexified with exponential transformation and effectively solved with the proposed OA/RLP algorithm.

Model *(MA):*

(a) Investment cost^and potential profit

$$\min\ z = \sum_{j=1}^{M} \alpha_j V \qquad \sum_{t=i}^{T} \frac{H_t}{H_t - 0_t}$$

(b) Volume for each stage

Vj^SijtBu i=l,N; j=l,M; t=l,T

(c) Production horizon time for each period

$$\sum_{i=1}^{N} \text{^ T} \; \text{~it} \; \text{« V-/t — A^t} \quad \text{t ..... 1 · 1}$$

(d) Bounds

$$V_j^U \leq V_j \leq V_j^L \qquad j=l,M$$
$$B_{it}^U \leq B_{it} \leq B_{it}^L \qquad i=l,N; \quad t=l,T$$
$$0 \leq 0_t \leq 5_t H_t \qquad t=l,T$$

where the variables of the problem are the volume of each unit, Vj, the batch size for the product i in period, Bit, and the extra production time in each period, 0_t. The parameters of this problem are the cost coefficients for the units OCJ and pj, the units size factors Syt, the cycle time for each product at each period Q_t, the demand for every product at each period Qit, the sales profit at each period based on the predicted demands pt, and the horizon time for each period Ht. It is assumed here that the production of the different products is increased uniformly in proportion to the demands Qit during the extra production time 0_t. Therefore, pt is given by the profit per period, based on the demands Qit, during each time period t. For realistic purposes the extra time 0_t is bounded between zero and a given fraction 8t, of the total horizon time for each period. The meaning of the lower bound is that the given demand Qit must always be satisfied, while the allowable increase in production cannot exceed a certain percentage of this demand

In the second model, the above batch design problem is further extended in order to include the design of the optimal topology and to address the expansion problem of the batch plant. The resulting problem is a multiperiod MINLP which can be convexified and solved by the proposed OA/MIRLP algorithm.

In the following multiperiod formulation the periods of operation are denoted by t = 1,... T, while the periods for initial installation and expansions are denoted by x = 1,... n. The planning problem is then to determine the optimal number of parallel units Nj^T at each potential expansion period x (x = 1,... II). One assumption concerning the parallel units for each stage is that they are of the same volume.

Model (M.2):

(a) Investment and expansion cost and potential profit

$$\min\ z \ll \sum_{j=1}^{M} ajNjvf + \sum_{T=2}^{n} \sum_{j=1}^{M} aj\,(Nj - Nj''^{1})\, vf + \sum_{T=2}^{n} ?T? - \sum_{t=1}^{T} p\,t - {}^{1} \wedge - \frac{TT}{H_t - 0_t}$$

(b) Volume for each stage

$$Vj \geq Sij_t B_{it} \qquad i=1,N;\ \ j=1,M;\ \ t=1,T$$

(c) Cycle time for each product at each period

$$Nj^x Q_t \geq t_{ijt} \qquad x=1,ri;\ \ i=1,N;\ \ j=1,M;\ \ t=1,T;\ \ T\underline{D}\ n\ \text{and}\ t\bmod(x) = 0$$

(d) Production horizon time

$$\sum_{i=1}^{N} \% \wedge_{it} + 0_t \leq H_t \qquad t=1,T$$

(e) Number of parallel units

$$N_j^{\tau} = \sum_{k=1}^{K} {}^{kY} j_k^{\tau} \qquad J=1,M;\ \ x=1,\Pi$$

(f) Single choice of Nj*

$$\sum_{k=1}^{K} Y]_k = 1 \qquad j=1,M;\ \ t=1,n$$

(g) Non decreasing number of expansion units

$$N/^{+1} \geq Nj^x \quad x=1,n-1$$

(h) Logical conditions for each expansion

$$N_j^{\tau+1} \le N_j^{\tau} + K\, Z^{\tau+1} \qquad \tau=1, \Pi-1$$

(i) Bounds

$$1 \le N_j^{\tau} \le N_j^{\tau U} \qquad j=1, M; \quad \tau=1, \Pi$$

$$C_{it}^{L} \le C_{it} \le C_{it}^{U} \qquad i=1, N; \quad t=1, T$$

$$Y_{jk}^{\tau} = 0,1 \qquad j=1, M; \quad \tau=1, \Pi; \quad k=1, K$$

$$Z^{\tau} = 0,1 \qquad \tau=2, \Pi$$

Here the additional variables of the problem are the number of parallel units for each stage at each potential expansion period τ, N_j^{τ}, the cycle time for each time period C_{it} which is a variable in this model, the binary variable indicating the number of parallel units at each stage Y_{jk}^{τ}, and an additional binary variable Z^{τ} associated with the decision of an expansion at each potential expansion period. The parameter t_{ijt} represents the processing times for each product i at stage j at time period t. The parameter γ_j^{τ} represents the fixed cost associated with every expansion. The values of α_j^{τ} and γ_j^{τ} implicitly involve the discount rate.

Example problems

Example problems will be presented to illustrate the application of the proposed decomposition techniques. First the convexified NLP multiperiod multiproduct batch plant model will be solved for a small and a medium size plant, over several time periods in each case. Then the convexified MINLP model will be also solved for the two cases and for different operation and expansion periods. The models were developed and solved within the modeling system GAMS. The package used for LP's and NLP's was MINOS, for MILP's it was SCICONIC and for the undecomposed MINLP it was the academic code DICOPT++ (Viswanathan and Grossmann, 1990).

Example 1. In this example the NLP case will be considered. First we convexify the model through an exponential transformation as suggested by Kocis and Grossmann (1988). The idea is to express the nonconvex product terms as the sum of exponential functions which are convex. This requires the definition of the transformed variables v_j =

In [Vj], bit = l^n [B^- Using the above and performing the necessary exponentiations yields the following convex NLP model:

Model (M/Cl):

(a) Investment cost and potential profit

$$\min \; z = \sum_{j=i}^{M} ctj \; \exp[pjVj] \; - \; \sum_{t=i}^{T} Pt \frac{H - L\sim}{H_t - 0_t}$$

(b) Volume for each stage

$$\bar{VJ} > \ln (Syt) + bu \qquad i=l, N; \; j=l,M; \; t=l,T$$

(c) Production Horizon time

$$\sum_{i=l}^{N} Qit \; dt \; \exp[- biJ \; + \; 0_t \; \bar{<} \; H_t \qquad t = 1, T$$

(d) Bounds

$$\ln [VjU] \leq VJ \leq \ln [Vj^L] \qquad j=l, M$$
$$\ln [\, B_{it}U] \leq bu \wedge \ln [B_{it}^{\;L}] \qquad i=l, N; \; t=l, T$$
$$0 \;\leq 0t \leq 8 H_t \qquad t=l,T$$

Note that the second summation in the objective function is only a function of ©t and that each term is convex in this variable.

Case I. It is assumed that the plant consists of three processing stages (M=3) with a single unit each, produces five products (N=5) and its operation will be designed for a series of different time periods, namely from one (T=l), up to twenty (T=20). The total horizon time will be one year. Table I contains the data for this example. The optimal solution for a typical five period problem is shown in Table II. The number of constraints in this problem is 16T+1 and the number of variables 6T+3, shown for typical cases in Table III. The computational results and a comparison with other methods are listed on Table IV.

Case II. In this case the plant consists of six processing stages (M=6) with a single unit each, produces ten different products (N=10) and its operation will be designed for a series of time periods ranging from one (T=l), up to twenty (T=20), over a one year horizon. The number of constraints in this problem is 61T+1 and the number of variables 11T+6.

Input data and size information are listed in Tables I and in, while the computational results are shown in Tabl^TV,

Table I. Data for Example 1.

Case I	**Products, N=5**		**Stages, M=3**		
Cost Coefficients	ctj = 250 ($/yr)	**pj= 0.6**	**j=l,M**		
Bounds on Volumes (L)	**Vf = 25**	Vj^u == **25000**	**j=l, M**		
Horizon time (h)	H_t = 8000/TT		**t=l,T**		
Production	QA_t e {210,000 - 290,000}	**QBte {110,000-195.000}**			
ranges	**Qct** 6 {160,000-200,000}	<3Dt6 **{130,000-185,000}**			
(Kg/yr)	QEI <E {100,000- 150,000}				
Stage		1	**2**	**3**	
Size factor Sijt	A	7.9	**2.0**	*52*	
for product i at	B	0.7	0.8	0.9	
stage j (same for	**C**	0.7	**2.6**	1.6	
all periods t)	D	4.7	2.3	1.6	
(L/KR)	E	1.2	**3.6**	2.4	
Processing time qt for	A	B	C	D	E
product i (all periods t), (h)	8.3	**6.5**	**5.4**	**3.5**	4.2
Case II	**Products, N=10**		**Stages, M=6**		
Additional	QAAt € {140,000 - 320,000}	QBBI € {120,000 - 230,000}			
production	Qcct e {110,000 - 210,000}	QDDI € {125,000 - 270,000}			
ranges (Kg/yr)	$Q_E E_t$e {115,000-225000}				

The use of a multiperiod model for the short term variation in product demand results in an optimal design and production planning. The advantage of a multiperiod design versus a worst case approach is shown in Table IL The worst case design approach is a single period design consisting of the maximum demand for each product over the different time periods. Although this scheme guarantees feasibility for all potential periods it is more expensive as expected. Depending on the actual value of the marginal profit of overproduction p_t, the worst case design gives a higher total cost (or lower profit) ranging from 5% to 20%. Depending on the value of p_t, there are two different trends in the optimal multiperiod solution. If the value of the marginal profit pt, is low there is one bottleneck period in which the additional production is zero, 0_t = 0, which means that in this period it is not worth producing more that it is suggested. In this case the batch sizes are the same

for all different periods due to the fact that the horizon constraint is active in all periods (see Table II). If the value of the marginal profit p_t is high, then the additional production 0_t is driven to its upper bound, hence there is no bottleneck period and the batch sizes are different for each time period Another characteristic of the multiperiod solution is the fact that, apart from the additional production in the above case, no variables lie on their bounds at the optimal solution.

Table EL Comparison of the optimal solution and the worst case solution for a five-period problem of case I

Optimal Design						
Stage:		1	2	3		
Volume (L):		3658	2140	2408		
Period:		1	2	3	4	5
Batch	A	463.0	463.2	463.2	463.2	463.2
size for	B	2675.5	2675.5	2675.5	2675.5	2675.5
each	C	823.2	823.2	823.2	823.2	823.2
product	D	778.4	778.4	778.4	778.4	778.4
(Kg)	E	594.6	594.6	594.6	594.6	594.6
Additional production (%)		5.3	8.6	14.9	0.6	0.0
Annualized Cost ($)		59,820				
Worst Case Design						
Stage:		1	2	3		
Volume (L):		4010	2346	2640		
Period:		1	2	3	4	5
Batch	A	507.6	507.6	507.6	507.6	507.6
size for	B	2933.0	2933.0	2933.0	2933.0	2933.0
each	C	902.5	902.5	902.5	902.5	902.5
product	D	853.3	853.3	853.3	853.3	853.3
(Kg)	E	651.8	651.8	651.8	651.8	651.8
Additional production (%)		15.5	19.0	26.0	10.3	9.0
Annualized Cost ($)		62,180				

The sizes of the different problems with respect to the number of variables and constraints and the respective trends as the number of periods increases, are shown in Table III.

Table III. Problem sizes for cases I and II in Example 1.

Case I		
Number of periods	Total number of variables	Total number of constraints
5	33	81
10	63	161
20	123	321
30	183	481

Case II		
Number of periods	Total number of variables	Total number of constraints
5	61	306
10	116	611
20	226	1221
30	336	1831

Table IV. Results for Example 1.

Case I			
Number of periods	SQP CPU time (s)(i)	MINOS CPU time (s)(i)	OA/RLP CPU time (s)(i)
5	7.1	5.6	5.3
10	34.6	14.6	11.1
20	1140.5	38.9	14.6
30	3566.8	69.2	36.4

Case II			
Number of periods	SQP CPU time (s)(i)	MINOS CPU time (s)(i)	OA/RLP CPU time (s)(i)
5	43.6	23.0	11.2

10	fç 279.5	63.2	30.2
20	1579.1	(iii)	84.8
30	(ii)	(iv)	222.6

(*) CPU time results on a 6320 VAX mainframe

(ii) SQP failed due to working set size limitations

(iii) MINOS failed from the given initial point

(iv) MINOS failed to solve

From the computational standpoint the performance of the proposed algorithm is compared to the case when a reduced gradient method (commercial code MINOS, by Murtagh and Saunders, (1985)) and a successive quadratic programming method (academic code SQP, by Biegler and Cuthrell, (1985)) was applied directly to these methods without performing decomposition. As it can be seen from the results on Table IV, OA/RLP outperforms MINOS and especially SQP. CPU time savings were achieved in all cases, particularly as the number of periods increases. Moreover, as it is seen in the case II problem, which is larger, MINOS failed to solve the 20 period problem from the same starting point, succeeding from a "better" one, and it failed on all problems with more than 20 periods. On the other hand the proposed OA/RLP method was successful in all different sizes of the problem in both cases, showing a remarkable robustness. Another important characteristic of this method is the fact that the number of the inner LP calls is constant (five and six for case I and case II, respectively) and independent of the number of periods, and hence the size of a particular problem. For the first case no NLP subproblem solution was required, while in the second case only one NLP subproblem was solved.

Example 2. In this example the MINLP batch plant design will be considered. First we convexify model (M.2) through an exponential transformation. This requires the additional definition of the transformed variables VJ = In [Vj], bj_t = In [Bid, " j^x = In [Nj^x], q_t = In [Qd Using the above yields the following convex MINLP model:

Model (M/C.2):

(a) Investment and expansion cost and potential profit

$$\min \ z = \sum_{T=2}^{n} \sum_{j=1}^{M} ({}^{a}T^{l\prime}\ {}^{a}P\ {}^{ex}P({}^{n}J'^{1+}\ Pj^{v}j) + \sum_{j=1}^{M} <xjexp(nj+ PjYj) + \sum_{T=2}^{n} fP$$

$$- \sum_{t=i}^{T} n\ P_t \frac{H_t}{H_t - \Theta_t}$$

(b) Volume for each stage

$$VJ \text{ £ In } (S_{ijt}) + bu \qquad i=l,N; \;\; j=l,M; \;\; t=l,T$$

(c) Cycle time for each product at each period

$$nj^* + cu \geq \text{In } (tjjt) \qquad T=l,n; \;\; i=l,N; \;\; j=l,M; \;\; t=l,T; \;\; T \text{ a n andtmod}(x> = 0$$

(d) Production horizon time

$$\sum_{i=1}^{N} Q_{it} \exp(c_{it} - b_{it}) + 0_t \leq H_t \qquad t=l, T$$

(e) Number of parallel units

$$\sum_{k=l}^{K} \ln(k) y J_k = nJ \qquad x=l,T; \;\; j=l, M$$

(f) Single choice of Nj^x

$$\sum_{k=l}^{K} Y]_k = l \qquad j=l,M; \;\; x=l,n$$

(g) Non decreasing number of expansion units

$$nj^{x+1} \geq nf \qquad T=1,II-1; \;\; j=l,M$$

(h) Logical conditions for each expansion

$$nj^{x+1} \leq nf + \text{In } (K) \, Z^{T+1} \qquad x=l,II-l; \;\; j=l,M$$

(i) Bounds

$$0 \leq nj^x \leq \text{In } [NJ^{TU}]$$
$$\text{In } [C_{it}L] \leq c_{it} \leq \text{In } [C_{it}U] \qquad i=l, N; \;\; x=l, \Pi$$

<u>Case I.</u> The plant consists of three processing stages (M=3) and each of them may consist of up to five parallel units (K=5). It produces five products (N=5) and its operation will be designed for a series of different time periods, namely from two (T=2), up to twenty (T=20). The total time horizon will be ten years, and expansions will be allowed to occur

over a variety of different expansion periods (11=2,3,5). Table V contains the data for this example, while Table VEtt contains information on the sizes of several problems for cases I and II. The number of constraints in this problem is 16T+27I1-5, the total number of variables 6T+24I1+3, and the number of binary variables 16FI-1, where n is the total number of possible expansions. The optimal solution for a typical problem is listed in Table VI and a comparison case in Table VII, while computational results are listed in Table IX.

<u>Case II.</u> In this case the size of the designed plant is bigger consisting of six processing stages (M=6) and each of them may consist of up to five parallel units. It produces ten products (N=10) and its operation will be designed for a series of different time periods, namely from two (T=2), up to twenty (T=20). The total time horizon will be ten years, and expansions will be allowed to occur every year. The number of constraints in this problem is 61T+84n-ll, the total number of variables 11T+77I1+6, and the number of binary variables 6111-1, where n is the total number of possible expansions. Computational results are listed on Table DC.

Table V. Additional data for Example 2.

Case I	Products, N=5	Stages, M=3	
Cost Coefficients	Oj° = 250 ($/yr) 7° = $20,000	Pj = 0.6	j=l, M
Annual discount rate	n=0.1		
Max. number of parallel units	NJ^{TU} = 5		j=l, M
Horizon time (h)	H_t = 10*8000/T		t=l, T
Total Horizon 10 years			
Production	Q_{At} e {100,000 - 580,000}	Q_{Bt} e {50,000 - 680,000}	
ranges QQ^	{80,000-640,000}	Q_{Di}e {160,000-635,000}	
(Kg/yr)	Q_{Et}€ {120,000-575,000}		
Case II	Products, N=10	Stages, M=6	
Bounds on Volumes (L)	Vj^L = 25	Vj^U = 45000	j=1,M
Additional	QAAI e {140,000 - 450,000}	QBBI e {150,000 - 595,000}	
production	Qcct e {110,000 - 600,000}	QDDI e {150,000 - 575,000}	
ranges (Kg/yr)	QEEt^ {140,000-510000}		

The optimal solution of a ten period batch plant with five potential expansions over a ten year horizon is listed in Table VI. Both the above expansion model and a period-by-period capacity expansion model were investigated. In the period-by-period design approach the optimization is being done for each individual expansion period separately, while in the proposed model the expansion decision is made simultaneously with the optimization of the other design variables. The period-by-period design gives a suboptimal solution that always results in a substantially higher total cost ($934,800 vs. 655,500). The results from a comparative study for a three period plant with three potential expansion periods over a ten year horizon are presented in Table VIL The cost in the period-by-period expansion design approach is 42% higher than the design through the proposed expansion model, for this three period problem.

Table VI. Optimal solution for a five-expansion twenty period batch plant

Stage			1	2	3	
Parallel	1	(0)(0	2	2	2	
units for	2	(4)	2	3	3	
each	3	(8)	3	3	4	
expansion	4	(12)	3	3	4	
period®	5	(16)	3	3	4	
Volume per unit (L)			24288	18381	15987	
Period		**1**	**5**	**10**	**15**	**20**
Batch	A	3074	3074	3074	3074	3074
size for	B	17763	17763	17763	17763	17763
each	C	620	7070	6978	7070	7070
product	D	4222	5168	2929	5168	5168
(Kg)	E	5106	5106	5106	5106	5106
Additional production (%)		33.3	14.8	33.3	18.5	0.0

(*) In parenthesis the actual year of operation is shown

Table VII. Comparison between a period-by-period expansion design and the proposed optimal expansion design

		Optimal Expansion Design		
Stage		1	2	3
Parallel units for	1	1	1	1
each expansion	2	2	2	3

period	3	3	3	4
Volume per unit (L)		22,471	16,950	14,794
Total Cost ($)		655,500		
Period-by-Period Expansion Design				
Stage		1	2	3
Parallel units for each expansion period	1	1	1	1
	2	5	6	7
	3	9	9	12
Volume per unit (L)		7,911	5,603	5,208
Total Cost ($)		934,800		

Table VIII. Problem sizes for cases I and II in Example 2.

Case I				
Number of expansions	Number of periods	Total number of constraints	Total number variables	Number of binary vars.
2	2	81	63	31
2	10	209	111	31
2	20	369	171	31
3	3	115	93	47
5	5	200	153	79
5	10	280	183	79
5	20	440	243	79
10	10	425	303	159

Case II				
Number of expansions	Number of periods	Total number of constraints	Total number variables	Number of binary vars.
2	2	279	182	121
2	10	784	270	121
2	20	1377	380	121
5	5	714	446	304
5	10	1019	501	304

5	20	1629	611	304
10	10*	1439	886	609

Table DC. Computational results for Example 2.

Case I						
Number of expansions	Number of periods	Parall. Units Stage			DICOPT++ CPU time (s)(0 (major iter.)	OA/MIRLP CPU time (s)(0 (major iter.)
		1	2	3		
2	10	2	2	3	163.9 (2)	87.7 (2)
		2	2	3		
3	3	1	1	1	105.9 (2)	54.4 (2)
		1	1	1		
		1	1	2		
5	5	1	1	1	631.5 (3)	594.4 (3)
		1	1	1		
		1	2	2		
		2	2	3		
		2	2	3		
5	10	1	1	1	404.7 (2)	189.2 (2)
		2	2	3		
		2	2	3		
		2	2	3		
		2	2	3		
5	20	2	2	2	1982.9 (3)	1526.4 (3)
		2	3	3		
		3	3	4		
		3	3	4		
		3	3	4		
10	10	1	1	2	(ii)	1584.2 (2)
		1	1	2		
		1	1	2		
		2	2	3		
		......	......	......		
		2	2	3		

		fc	Case		II				
2	10	2	2	2	2	2	2	3722.2(3)	2631.6(2)
		3	3	3	4	3	3		
5	5	1	1	2	2	2	1	<ai)	3818.2(2)
		1	1	2	2	2	1		
		2	2	2	2	2	2		
		2	2	2	2	2	2		
		2	2	2	2	2	2		

® CPU time results on a 6320 VAX mainframe

(y) Internal MILP solver failed

(***) Limit of 5,000 sec exceeded in the first iteration

The computational performance of the proposed OA/MIRLP scheme is better than DICOPT++ in all of the examined cases. In the above examples the lower bound predicted at each iteration by OA/MIRLP was always higher than the one predicted by DICOPT++. The number of major iterations for the proposed method were fewer or equal to the ones of DICOPT++. This is attributed to the more accurate representation of the master problem in this method. However, since the major part of the computations are devoted to the solution of the master MILP, the results are not always showing a systematic increase in the time savings as the number of time period increases. This can be attributed to the different degree of difficulty in solving the primal NLP and the master MILP. Since the advantage of this method lies mainly in the NLP part of the problem, the difference becomes more apparent in problems where the NLP phase shares a substantial portion of the total solution time. Overall the proposed method is more robust and efficient in solving big multiperiod NLFs and MINLFs. This performance can be partially attributed to the fact that it mainly relies on the solution of LP's and MILFs and also because whenever an NLP subproblem is solved, it is done on the reduced space of state and control variables and for each period independently.

Conclusions

A decomposition algorithm based on outer approximation has been proposed for efficiently solving convex NLP and MINLP multiperiod design problems. The proposed methods have been successfully applied to the design of a multiperiod multiproduct batch plant, with a capacity expansion deterministic model. For this purpose a general model for multiperiod multiproduct batch plants was developed. Major components of the design problem, which have not been incorporated in a single formulation in the past, such as the optimal production policy and a realistic capacity expansion formulation, lead to an optimal and practical design. With this formulation large overdesign and suboptimal, hence expensive, expansion policies are avoided. For the efficient solution of such models the proposed algorithm was applied. The computational performance was illustrated for the NLP and MINLP problem on two examples for several periods of operation and several optional expansion times. The results show an advantage of the presented method over traditional solution approaches, in both robustness and time efficiency, issues very important for design applications in chemical process industries.

References

Duran M. A. and I. E. Grossmann, An Outer Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Math. Program.* 36, 307 (1986)

Eaves C. B. and W. I. Zangwill, Generalized Cutting Plane Algorithms. *SIAM J. Control* 9, 529 (1971)

Geoffrion A. M., Generalized Benders Decomposition. 7. *Optimization Theory Applic.* 10,237 (1962)

Grossmann I. E. and K. P. Halemane, Decomposition Strategy for Designing Flexible Chemical Plants. *AIChE J.* 28, 686 (1982)

Grossmann I. E., Halemane K. P. and R. E. Swaney, Optimization Strategies for Flexible Chemical Processes. *Comp. Chem. Engng.* 7, 439 (1983)

Grossmann I. E. and R. W. H. Sargent, Optimum Design of Chemical Plants with Uncertain Parameters. *AIChE J.* 24, 1021 (1978)

Grossmann I. E. and R. W. H. Sargent, Optimum Design of Multipurpose Chemical Plants. *Ind. Eng. Chem. Process Des. Dev.* 18, 343 (1979)

Grossmann I. E. and D. A. Straub, Recent Developments in the Evaluation and Optimization of Flexible Chemical Processes. Paper presented in COPE/91, Barcelona

Grigoriadis M. D., A Projective Method for Structural Nonlinear Programs. *Math. Prog.* 1, 321 (1971)

Halemane K. P. and I. E. Grossmann, Optimal Process Design under Uncertainty. *AIChE J.* 29, 425 (1983)

Han S. P. and O. L. Mangasarian, Exact Penalty Functions in Nonlinear Programming. *Math. Prog.* 17, 251 (1979)

Kelley J. E.,t The Cutting Plane Method for Solving Convex Problems. *Journal of the Society for Industrial and Applied Mathematics* 8,703 (1960)

Kendrick D. and A. Meeraus, GAMS, an Introduction. Technical Report Development and Research Department at the World Bank, Washington, D. C. (1985)

Kocis G. R. and I. E. Grossmann, Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Ind. Engng. Chem. Res.* **26**, 1869 (1987)

Kocis G. R. and I. E. Grossmann, Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis. *Ind. Engng. Chem. Res.* **27**, 1407 (1988)

Luss H., Operations Research and Capacity Expansion Problems: A Survey. *Operations Research* **30**, 907 (1982)

Minoux M., Mathematical Programming: Theory and Algorithms. Wiley, New York 1986

Murtagh B. A. and M.A. Saunders, MINOS User's Guide. Technical Report SOL 83-20, Systems Optimization Laboratory, Department of Operations Research, Standford University (1985)

Topkis M. D., A Cutting Plane Algorithm with Linear and Geometric Rates of Convergence. *Journal of Optimization Theory and Applications* **36**, 1 (1982)

Viswanathan J. and I. E. Grossmann, A Combined Penalty Function and Outer Approximation Method for MINLP Optimization. *Computers chem. Engng.* **14**, 769 (1990)

Wellons H. S. and G. V. Reklaitis, The Design of Multiproduct Batch Plants under Uncertainty with Staged Expansion. *Computers chem. Engng.* **13**, 115 (1989)

Appendix

The two analytic examples solved to identify the computational nature of MPD's are presented here.

Example 1

$$\min \sum_{i=1}^{N} (a_i x_i^2 + b_i d_1 + c_i d_2) + d_3^2 + d_4 - d_5$$

$$\text{s.t.} \quad -c_i x_i + d_1^2 + d_2^2 + d_3^2 - d_4 + d_5^2 \leq 0$$

$$-a_i x_i + 2d_1^2 - d_2 + 3d_3^2 + d_4^2 - d_5 \leq 0 \qquad i=1, \ldots, N$$

Example 2

$$\min \sum_{i=1}^{N} (a_i x_i^2 + b_i d_1 + c_i d_2) + d] + (U - d_5 + 4 d_6 + 4d_7 + 4d_8 + d_9 - d_{10}$$

s.t. -qxj + d? + dô + d; - d4 + dl + dl + <$ + dl + d^ ≤ 0

$$-a_i x_i + 2d_1^2 - d_2 + 3d_3^2 + d_4^2 - d_5 + d_7 + d_8 + d_9^2 + d_{10} \le 0 \quad i = 1, \ldots, N$$

where the parameters are a* e {2, 6}, bi e {-25, -10}, q *e* {10, 27} for different periods i. As stated in the problem statement section xj are the variables for each period i, and di, ...f dio arc the complicating variables.