Multiperiod Design Optimization with SQP Decomposition

D. Varvarezos, L. Biegler, L Grossmann

EDRC 06-155-93

Multiperiod Design Optimization with SQP Decomposition

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

December 1992

* Author to whom all correspondence should be addressed.

Abstract

This paper presents a new decomposition method for solving large scale multiperiod design problems. These problems are formulated as nonlinear optimization problems with a special block angular structure and linking variables. The proposed method, based on Successive Quadratic Programming, uses a decoupling scheme and projects the original problem into a quadratic subproblem involving only the complicating variables. Using this subproblem as the main coordination step, the problem in the full space is solved as a stream of independent single period problems.

The key property of this method is that the computational effort scales linearly to the number of periods compared to a quadratic and cubic increase for general purpose reduced gradient and SQP methods, respectively. To illustrate this property, the method is applied to four example problems including **two** in multiperiod chemical process design. Its performance is superior to both MINOS and SQP in terms of computational demands, number of function evaluations and solution robustness.

1. Introduction

In this work we address the problem of designing multiperiod chemical plants, a special class of the more general problem of design for flexibility. The advantage of a formal approach to the design of multiperiod plants is that operation is guaranteed to be feasible and optimal over the specified set of operating conditions. The importance of developing systematic methods for designing multiperiod plants as well as the difficulty associated with this approach was discussed in our previous work (Varvarezos, Grossmann and Biegler, 1992),

Difficulties in formulating and solving multiperiod design models stem from the large scale size of the problem. As the number of periods increases, there is a disproportionate increase in the computational demand and a corresponding decrease in the robustness for standard optimization methods. This behavior prohibits the solution of even moderately sized multiperiod models with standard general purpose nonlinear optimization methods. It is therefore the objective of this paper to develop an efficient method for solving nonlinear multiperiod problems.

In this work we propose a decomposition scheme based on Successive Quadratic Programming (SQP) principles for solving general nonlinear multiperiod models that guarantees global convergence to a local optimum. The theoretical properties and the conceptually simple Newton-based framework are the main reasons for choosing to base our decomposition on successive quadratic programming. On the other hand, by using the special structure of the multiperiod problem we can avoid the shortcomings usually associated with SQP methods. With this decomposition method (MPD/SQP) the issue of computational efficiency and robustness with increasing the number of periods is effectively addressed.

The rest of the paper is organized as follows. Section 2 describes the mathematical form of the problem and gives a brief review of decomposition methods for this particular problem structure. In Section 3 we present the development of our decomposition method starting with a decoupling scheme, a range and null space decomposition and finally a projection scheme on the small subspace of complicating variables. The problem of explicitly dealing with inequality constraints is addressed in Section 3.4 as an integral part of this work. Section 4 demonstrates the theoretical and algorithmic properties of the method. The performance characteristics are illustrated in Section 5 with two small multiperiod problems and through two multiperiod process design problems involving a Heat Exchanger Network (HEN) design and a small flowsheet problem. Comparisons are

drawn to general purpose optimization algorithms where it is shown that the computational effort for the proposed method scales linearly to the number of periods. Finally, conclusions are discussed in Section 6.

2. Problem Statement

The multiperiod design problem in steady state operation for a given topology can be represented in a mathematical programming form by a nonlinear programming problem (NLP). A structural characteristic of the multiperiod problem is that it involves two distinct classes of variables, the design and the state-control variables. The design variables, d, represent equipment parameters such as reactor and vessel volumes or heat exchanger areas, and remain the same in all periods of operation. The state and control variables, Xj, represent operating conditions such as temperatures, flowrates or concentrations, and are different for each period of operation i. The general NLP formulation for the multiperiod problem involves the minimization of a cost based objective function consisting of fixed costs (%) and operating costs (fj) for each period subject to a set of equality and inequality constraints for each period (hj, gj), as well as constraints on the design variables and bounds on all the variables. The resulting problem (P.I) is:

$$\begin{aligned} &\text{minimize } * = fo(d) + \sum_{i=1}^{N} /,(<*,*,) && \text{(P.I)}\\ &\text{subject to } \quad /*_i(d, x_i) = 0 \quad 1_1 \\ &\qquad\qquad r(d) \le 0 \\ &x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le X_iU\}, \qquad i = 1, \ldots N \\ &d \in D = \{d \in R^q \mid d^L \le d \le d^U\} \end{aligned}$$

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 lies in its block diagonal structure as shown in Figure 1a. The design variables, d, are the complicating variables interconnecting all periods of operation, whereas the overall matrix remains sparse but structured.

There are two major difficulties associated with the solution of the multiperiod problem that relate to the computational efficiency and robustness. Firstly, the computational requirements for solving (P.I) using general purpose optimization methods,

such as reduced gradient or successive quadratic programming, increase quadratically or even cubically as a function of the number of periods considered. Secondly, as a rcsult of the large problem size these methods are likely to fail to find a solution, especially as the number of periods becomes larger. Based on the above arguments, the need for a special solution method to efficiently address this problem becomes apparent A promising direction is one that exploits the block diagonal structure of the problem (in xO through a decomposition strategy. Once the design variables are fixed, the subproblems for each period become decoupled, and hence they can be optimized independently for x_9 i=l, N.

For the case of multiperiod Linear Programming (UP) problems, special block angular structured problems have been addressed through special solution procedures ranging from primal-dual decompositions to forward simplex methods (see Aronson, 1980 for a review). However, few methods have addressed the NLP problem through decomposition schemes. The main characteristic of these methods is a two phase solution procedure. In the first phase design variables are fixed and the decoupled problems are solved. In the second phase, the design variables are updated The basic idea behind the NLP decomposition strategy by Grossmann and Halcmane (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 An important limitation with these decomposition techniques lies in the definition of the restricted problem which requires the dynamic elimination of state variables from the equation and active inequalities which change at each major iteration.

The most recent work in multiperiod optimization addressed the problem in both NLP and MINLP formulations (Varvarezos, Grossmann and Biegler, 1992) for convex problems. An outer-approximation based method was developed in which the original problem was approximated by accumulating linearized versions through a two phase procedure. This method was successfully applied to the solution of convex problems and in particular to the design of multiproduct batch plants with future capacity expansions. However, since the method was developed for convex fonnulations there are limitations of applicability to non-convex problems.

3. Proposed MPD/SQP Method

This new method is motivated from the ideas of Successive Quadratic Programming (Han, 1977). These methods are known to have global convergence properties under mild

conditions to a local solution. The main reason for using SQP as the framework of this decomposition comes from a dual view of the method with respect to different modeling environments. From an equation based modeling standpoint, in which the linear algebra dominates the overall computational effort, function evaluations are not particularly expensive to obtain. In a modular process simulation environment, however, in which one function evaluation corresponds to a full simulation, the number of function evaluations is critical for a successful method It is also known that SQP-like methods are in general favored in such modeling environments due to the fewer function evaluations required. Therefore, the need for developing a method that can be successful in both modeling frameworks suggests an SQP-based approach to the decomposition.

The idea behind this method comes from a decomposition scheme for a special class of parameter estimation problems (Tjoa and Biegler, 1991) with a similar block diagonal structure involving equality constraints only. Here, a Quadratic Programming (QP) subproblem is solved only in the design variable space and the state and control variables at each period are calculated separately. The advantage of such a scheme is that the computational effort for obtaining the state and control variables increases only linearly with the number of periods.

One way to exploit the structure of the NLP problem (P.I) is by introducing additional period dependent pseudo-design variables 6j to replace the design variables in the equations for each period Since all of these have to be equal to the design variable vector d, we also add one set of equations for each period:

$$\$ \quad -d \quad = 0 \qquad\qquad \dot{i} = l_f \ldots \quad N \qquad\qquad (3.1)$$

and initialize 5} to d This transformation provides a special structure for the coupling variables d, although it does not completely decouple the different periods. In this structure the complicating variables, d, appear in a small subset of linear equations as seen in (P.2). The new structure of the problem after the decoupling can be seen in Figure 1b. This structure is further utilized through a range and null decomposition scheme for each period

In order to transform all the inequality constraints into equations and simple bounds, we introduce additional non-negative slack variables si. With all of the additional variables and constraints (P-l) becomes:

$$\text{minimize } 0 \quad =/</<^*) + \sum_{i=1}^{N} /, \wedge, xj \qquad\qquad (P.2)$$

$$\left.\begin{array}{l} \text{subject to} \quad h\text{-}\wedge S\wedge xJ = 0 \\ \qquad\qquad\qquad\qquad\qquad\quad 0 \\ \qquad\qquad S,\text{-}\,d = 0 \\ \qquad\qquad r(d) \le 0 \end{array}\right\} \quad i = 1....N$$

$$\begin{aligned} &x_i \in X_i = \{x_i \in R^n \mid x_i^L \le x_i \le x_i^U\}, && i = 1,\ldots N \\ &Si\,e\,Si = \{si\,e\,R^m\,I\,Si \wedge 0\}, && i = 1,\ldots N \\ &d\,e\,D = \{d\,e\,R < n\,d^L \pounds d \pounds d^u\}, \qquad \delta_i \in D && i = 1,\ldots N \end{aligned}$$

Applying an SQP scheme we can solve the above problem (P.2) iteratively by detennining a search direction at each iteration k from die following quadratic programming problem:

$$\begin{aligned} \text{minimize} < p - \; & V_d/_O{}^r \; M \; + \frac{1}{2}\text{-}M^T \nabla_d^2 L_0 \, M. + \\ & \prod_{i=1}^{N} (\nabla_p f_i^T \, p_i \; + \; \frac{1}{2} p_i^T \, \nabla_p^2 L_i \, p_i) \\ \text{subject to} \quad & \bar{h}_i \; + \; \nabla_p \bar{h}_i^T \, p_i = 0 \qquad i = 1, \ldots iV \\ & r \; + \; V_d r^T \, Ad \; \pounds \; 0 \end{aligned} \tag{P.3}$$

$$Pi \; = \; \begin{bmatrix} \chi\, r \text{-} \chi f \\ s_i^{k+1} \text{-} \; s_i^k \\ Sf^{*1} \text{-} 5_i^* \end{bmatrix} \qquad \bar{h} = \begin{bmatrix} h_i^k \\ 8t \; +^{**}_i \\ \text{-}Ad \end{bmatrix} \qquad V,K = \begin{bmatrix} 1 & & 10 \\ \nabla_p h_i^k & \mid \nabla_p (g_i^k + s_i^k) & \mid 0 \\ 1 & & 1 \; \mathrm{I} \end{bmatrix}$$

$$\begin{aligned} &p_i \in P_i = \{p_i \in R^{n+m+q} \mid p_i^L \le p_i \le p_i^U\}, && i = 1?\ldots N \\ &\Delta d \in D = \{\Delta d \equiv (d^{k+1} \text{-} d^k) \in R^q \mid d^L \text{-} d^k \le \Delta d \le d^U \text{-} d^k\} \end{aligned}$$

Here, LQ and L, are the Lagrangian functions based on (P.2) defined as:

$$L_0 \text{-} /.GO \; + \bar{r}^T(d) \, \lambda_0$$

$$L_i \; = \; \sum_{i=1}^{N} f_i(p_i) \; + \; \bar{h}_i^T(p_i) \, \lambda_i \; + \; R_i \, \mu_i$$

where $\bar{r}$ includes r and the bounds on Ad, *XQ* and X; are the Lagrange multipliers corresponding to F and ^ respectively. Matrix Rj is a diagonal matrix with its elements representing whether the bound on a state or control variable is active or not based on the current active set Aj = {j I py = pjj^B } (for simplicity of notation superscript B denotes either lower or upper bound), so that:

$$R\ddot{?} = \left\{ \begin{array}{l} 0 \text{ if } j \notin A_i \\ 1 \text{ if } j \in A_i \end{array} \right\} \tag{3.2}$$

and m are the Lagrange multipliers corresponding to the active bounds on the expanded state and control variables step pj.

Problem (P.3) is the full space QP that has to be solved at each iteration. In order to exploit the special structure of the above problem we will consider the first order Kuhn-Tucker optimality conditions for the state and control variable step vector pj

$$A \cdot \left\{ \begin{array}{l} \nabla_p f_i + \nabla_p^2 L_i\, p_i + \nabla_p \bar{h}_i^T \lambda_i + R_i\, \mu_i = 0 \\ \bar{k} + V_p \bar{h}?Pi = 0 \\ R_i \quad (Pi \quad -pf)-0 \\ tf\ ip, \sim pf) = 0 \end{array} \right\} \tag{3.3}$$

3.1 **Range and Null** Space Decomposition

In order to reduce the dimensionality of the problem and exploit its special structure we will introduce a range and null space decomposition scheme. The idea is to partition the full space of the state and control variables Pj at each period into two subspaces. The null space Zj will be tangent to the equality constraints of the quadratic problem (P.3), whereas the range space Yj will be orthogonal to Zj so that the following relations hold:

$$\begin{array}{l} zj\ v,/\bar{0}\% = o \\ ZjY, = 0 \end{array} \tag{3.4}$$

Accordingly the step direction for the state and control variables p; becomes:

$$A = {}^{z}.Pz, + YiPr, \tag{3.5}$$

By pre-multiplying the first equation in (3.3) by Z^T, using (3.4) and (3.5) the following system of equations gives the new direction for the variables p;. A standard assumption with reduced Hessian methods, that will be used here, is that ZiV^2LjYip_yj is negligible. Note also from the definition of $\bar{/i}$, in (P.3) that p_{Zl} and p_{Yi} are implicit functions of Ad.

$$
\begin{aligned}
& Z_i^T \nabla_p f_i + Z f V^2_p L, Z_{iPzi} + Z_{S_i}^T \mu_1 = 0 \\
& \bar{h}_i + \nabla_p ?\mathrm{r}, .\mathrm{Prj} = 0 \\
& Z_{S_i} p_{Z_i} + Y_{S_i} p_{Y_i} = R_i p_i^B
\end{aligned}
\tag{3.6}
$$

Here Zgj and Y_Sj are given by:

$$
\begin{aligned}
Z_{S_i} &= R_i Z_i \\
Y_{Si} &= R_i Y_i
\end{aligned}
\tag{3.7}
$$

Since the range and null spaces based on (3.5) are not uniquely defined, the exact forms used in this work are based on partitioning x; into (n-m) independent (control) and m dependent (state) variables, z; and *y*-$_v$ respectively. This allows the following definitions of Z; and Yj (Tjoa and Biegler, 1992):

$$
Z_{\cdot} = \begin{bmatrix} I \\ -(\nabla_y h \quad (\nabla_z h_i^T) \\ 0 \end{bmatrix} \quad \text{and} \quad Y_{\cdot} = \begin{bmatrix} (\nabla_z h_i)(\nabla_y h_i)^{-1} & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \tag{3.8}
$$

3.2 Projection Step

System (3.6) is now decoupled into two square systems that can be solved successively. First pYi is found by:

$$
Pr, = - <?\bar{*}\, YJ^l \bar{h} \tag{3.9}
$$

and then pzj is given by the solution of the following system:

$$
\begin{bmatrix} Z_i^T \nabla_p^2 L_i Z_i & \bar{Z}_{S_i}^T \\ \bar{Z}_{S_i} & 0 \end{bmatrix} \mathbf{ft}] - \begin{bmatrix} Z_i^T \nabla_p f_i \\ r\bar{R}_{iP} f + \bar{Y}_{S_i} p_{Y_i} \end{bmatrix} \tag{3.10}
$$

In the above system $\bar{Z}_S$^$\bar{Y}_{S_+}$ and/?, are matrices defined by the collection of the nonzero rows of the corresponding matrices. Similarly, /Z, is defined by the nonzero elements of /*,-. Note that in (3.9) and (3.10) the vectors Pzj and pyj are linearly related to Ad through $\bar{k}$, (see P.3). For simplicity of notation we will express pzi and pyi in terms of Ad as follows:

$$
\begin{aligned}
Pz_t &= A_{Z_i} + B_{Z_i} Ad \quad \text{and} \quad ZtPz, = ^{z} \backslash + Z_{B(}Ad \\
p_{Y_i} &= A_{Y_i} + B_{Y_i} Ad \quad \text{and} \quad YiPr, = Y_{A_i} + Y_{B_i}.Ad
\end{aligned}
\tag{3.11}
$$

where Azj, B_Zi, A_Yi and B_Yi are obtained through the solution of (3.9) and (3.10). Now if we substitute (3.11) into the original QP in (P.3) we get a projected QP expressed only in terms of Ad.

$$\text{minimize } \theta = \left[\nabla_d f_0^T + \sum_{i=1}^{N}\{\nabla_p f_i^T (Z_{B_i} + Y_{B_i}) + (Z_{A_i} + Y_{A_i})^T \nabla_p^2 L_i (Z_{B_i} + Y_{B_i})\}\right]\Delta d + \frac{1}{2}\Delta d^T\left[\nabla_d^2 L_0 + \sum_{i=1}^{N}\{(Z_{B_i} + Y_{B_i})^T \nabla_p^2 L_i (Z_{B_i} + Y_{B_i})\}\right]\Delta d$$

$$\text{subject to} \quad r + V_d r M \leq 0 \qquad \text{(P.4)}$$

Since all the variables Pi are expressed in terms of Ad the solution of the above QP is the main coordination step in this method since it provides all the information necessary to construct the next search direction for the state and control variables at each period. Formulating this projected QP in the space of d makes it much more convenient to solve for two reasons. Firstly, for process design problems we have dim(d) « N- dim(xj) and secondly we note the size of the projected QP (P.4) is independent of the number of periods N.

3.3 Quasi-Newton Updates to the Hessians

The need for an approximation to the exact Hessian is the main idea behind quasi-Newton methods. This approximation is **obtained by** using **the known curvature along the** search direction. A commonly used approximation is given by the BFGS formula given by:

$$H^M = H^k - \frac{H^k s s^T H^k}{s^T H^k s} + \frac{y y^T}{y^T s} \qquad \textbf{(3.11)}$$

where:

$$y = \nabla_x f^{k+1} - \nabla_x f^k$$

$$s = JC^{*+1} - x^k$$

and H^k is the approximation to the Hessian at the k* iteration.

In this problem there are two Hessian matrices to approximate. The first is the ***reduced Hessian of the Lagrangian for each period Z]^y V\L_i^* L_i*** **that appears in (3.10). In the** reduced Hessian update definitions of s and y in the BFGS update formula (3.11) are not unique and a number of different choices have been proposed (Nocedal and Overton, 1985). In this work we have:

$$y_{\sim} = Z^{*yk-\backslash\backslash} 7^* V_J^{\wedge *ik+l} - Z^{\wedge "\wedge \backslash 7} {}_J f^{\wedge "} \qquad (3.12)$$

$$s = a(x^{k+l} - x^k)$$

where a is the steplength from the linesearch (which is used in each major iteration to reduce the merit function). Note that based on the above definitions no Lagrangian multiplier estimates are required for evaluating y.

The second Hessian that has to be evaluated is the *projected Hessian for d,* that appears in (P.4) and has the form $[Vfc, + \sum_{i=1}\{ (Z_{Bi} + Y_{Bi})^T VjL_i. (Z_{Bt} + Y_{Bi}) \}J.$ *A* BFGS update scheme will be used here too. The choice of y and s for this update based on the gradient in (P.4) should be:

$$y = \left[VJS + \sum_{i=1}^{N} \{ V f (Z^{*+Y} B_t) + (Z^* + Y^{\wedge} f V\% (Z_{Bt} + Y_{Bi})) \right]^{k+1}_{J} - \left[\nabla_d f_0^T + \sum_{i=1}^{N} \{ \nabla_{JJ} (Z_{Bi} + Y_{Bi}) + (Z^{\wedge} + Y^{\wedge} f V\% (Z_{Bi} + Y_{B(}) \} \right]^k$$

$$s = \alpha (d^{k+1} - d^k)$$

However, since some of the terms in the above formulas are not explicitly known and also based on the fact that their contribution vanishes as p_y approaches zero, these tenns will be ignored leading to:

$$y = \left[k/_o{}^r + \sum_{i=1}\{ v_p/f\ Y_{BI} + z\backslash \nabla_p^2 L_i Z_{B_i} \} \right]^{k+1} - \left[\nabla_d f_0^T + \sum_{i=1}\{ \nabla_p/f\ Y_{Bi} + Zj\ VJE,Z^{\wedge} \} J \right] \qquad (3.13)$$

$$s = \alpha (d^{k+1} - d^k)$$

Here we note that $V^{\wedge}fJ\ Z_{Bi}$ was ignored in (3.13) because it involves terms that reflect on the active set that are redundant Since the second order information regarding the bounds is zero this term is only affected by changes in the active set and its contribution does not reflect on the true Hessian. Hence it deteriorates the approximation. An alternative to the update for evaluating the projected Hessian in d is an exact evaluation directly from (P.4). This, however, requires information on the full Hessian for each period that is not available. In order to resolve this, a direct update on the full Hessian, as opposed to the reduced Hessian, could be employed. The main problem with this approach, though, apart from the increased computational demands, is that the full Hessian updates can become singular, which leads to stability problems for the method.

In order to guarantee a descent direction for this method we need to ensure that both the reduced Hessian for each period (Zj^BjZ}) as well as the projected Hessian in d are

positive definite. The necessary conditions for a positive definite BFGS update is an initial positive definite matrix (the identity matrix is usually chosen in the absence of any additional information) and $s^T y > 0$. Normally, when $s^T y \leq 0$ it is necessary to skip the BFGS updates to avoid indefiniteness or singularity in the new approximation (Nocedal and Overton, 1985). An alternative to this (Powell, 1978) attempts to perform an approximate update under these conditions. As a result of testing we recommend updating the reduced Hessian for each period Z ^ B ^ using Powell damping when $s^T y$:£ 0, whereas for the projected Hessian in d we simply skip the updates. The reasons behind this selection come from the different role of the two Hessians in this method. For the projected Hessian in d we are being conservative by skipping in order to keep the matrix positive definite and well conditioned, since the actual projected Hessian in d is expected to remain positive definite. For the reduced Hessian for each period ZJTBJZJ, however, this is only necessary if none of the degrees of freedom are consumed.

Finally, in the case where llsll —> 0 and llyll » llsll the updates may become singular even if $s^T y > 0$ and the true matrix is not singular. In that case it is also advisable to skip the updates in order to maintain the superlinear convergence characteristics of the method.

3.4 Active Set Method

As described in the decomposition part of the method, all the inequality constraints involving state and control variables are transformed into equations by adding new slack variables. Therefore all the inequality constraints that remain have the form of simple bounds on the state and control variables.

The most important consideration in developing an active set scheme for the given decomposition comes from the need to preserve the simplicity and uniformity of the decomposition. The main idea is to develop the bounding scheme preserving at the same time the structure of the method. As already presented in the development of the method (in (3.1) through (3.10)), the null space Z is defined only in terms of the equality constraints and is completely independent of the changes in the active bounds. This allows a uniform representation of all the periods. Also in the context of the reduced Hessian updates, Z^k and Z^{k+1} have the same size and therefore s and y are always well defined. The handling of the active bounds is done through the solution of system (3.10).

In a standard SQP method each QP subproblem (P.3) is solved to optimality at each iteration. This is equivalent in our case to solving (P.4) to optimality and determining the correct active set for the state and control variable bounds at each iteration. This is not a

very attractive option, however, since it would require the solution of (P.4) and (3.10) several times, adding and deleting bounds in each major iteration. On the other hand, the solution of (P.3) for the correct active set at each iteration does not necessarily reflect upon the correct active set at the optimum, especially in the early iterations where the information built up in the problem is insufficient and the predicted steps are inaccurate. Although there are proposals in the literature to terminate the solution of the QP early, the great majority of SQP algorithms solve the QP subproblem to optimality. It can be shown, however, that under mild conditions (Prieto, 1989) the incomplete solution of the quadratic subproblem does not affect the convergence properties of SQP methods.

Here, we will introduce an adaptive scheme with an early termination criterion for the QP subproblem (P.3), that has a guaranteed bound on the effort necessary to satisfy it In particular, the projected subproblem in d (P.4) will always be solved to optimality for a *given active set.* The termination criteria are naturally connected to the decomposition method and are defined in terms of the active set and the bounds encountered along the predicted search direction.

In this method we attempt to follow a feasible path with respect to the state and control variable bounds. We start with a point Xj° within the bounds and we add bounds as they are encountered on the course of each new direction pj^k. By solving (3.10) and (P.4) we have a new direction for all the state and control variables. For this direction we find the largest steplength CXQ ($0 £ OLQ £ 1$) for which no bound is violated. The bound that corresponds to this feasible OQ (in case of course OQ < 1) is added to the active set If this OCQ is less than a certain threshold value related to the progress of the method (e.g. k/30, where k is the major iterations counter) a minor iteration is performed; L e., the projected QP (P.4) is solved with the new active set at the same point of linearization. In addition to that, no more than l_x; bounds are allowed to be added to any period in a single minor iteration cycle. The value of 1; should be a small number (between 1 and n-m) to avoid overloading periods with bounds in the early stages of the method. From the above arguments there is a tradeoff between adding many constraints early which may require more iterations to remove, and using a lower threshold value on <XQ (and /j) which may require more iterations to add all the necessary constraints. In this work a value of k/30 was used as a cutoff point for the minor iterations. The constant used as a denominator (30) is arbitrary but it reflects on an average maximum number of expected iterations. Also for the value of the maximum number of added bounds for a full minor iterations cycle, t_v we used (n - m)/2.

Another important characteristic of this active set method stems from our decomposition procedure. Since we have effectively decoupled all different periods, only n

- m active bounds can be set at each period. The total number of degrees of freedom for the multiperiod problem is N-(n-m)+dim(d) and therefore there is an additional number of dim(d) possible active bounds that should be accounted for. As is often the case in multiperiod problems, some or even all of the possible active bounds are active at a single period. Since, the maximum possible number for active bounds per period is actually n - m + dim(d), some of the degrees of freedom corresponding to the design variables have to be consumed in some of the periods. In order to deal with this problem some of the active bounds on state and control variables have to be added to the projected QP (P.4) so that they are explicitly enforced. This number, however, should not exceed (dim(d) - dim($\bar{r}^*$)) since this is the maximum number of degrees of freedom in the projected QP.

3.5 Model SQP/MPD Algorithm

Based on the above discussion the algorithm for the proposed SQP/MPD method can be summarized as follows:

Step 1. Choose starting point for *dfi* and *xfi.* Initialize the reduced Hessian for each period and the projected Hessian in d to identity. Set major iteration counter k = 0. Set active set $A_0^0 = \emptyset$.

Step 2. Check for convergence. If the Kuhn-Tucker error is less than the desired tolerance e, STOP. Evaluate the objective function and constraints and all the gradients <J>; h, V<fio> ^p^i* ^phi^at (^i^k» d^k)- Set minor iteration counter m = 0 and *t*\ = 0. Set up a loop for all periods i:

a. Update active set A_m^k by removing at most one inactive constraint with the most negative *L* Construct Rj.

b. Set up auxiliary matrices to store the function and the gradients partitioned into dependent and independent variables and use LU decomposition scheme to obtain Zi.

c. Update the inverse of the reduced Hessian (ZjT V_p^2Li Zi)-1.

d. Solve (3.9) and (3.10) to get matrices ZAI, ZBJ, Y_{Ai}, Y_{Bi}

Step 3. Calculate the gradient y in (3.13) for the projected QP. If m = 0 update the reduced Hessian for the projected QP. Solve the QP subproblem to get a new direction Ad^k for the design variables.

Step 4. Calculate the new directions Z* p_{2i}, Yi p_{yi} and *Axfc* for each period i. Test for feasibility with respect to state and control variable bounds in the new direction based on a **full** step ((XQ = 1). If feasible proceed to Step 5.

a. If (XQ = 1 results in bound violation, then find the maximum OQ without bound violation. Add the corresponding bound (i*» j*) to $A_m{}^k$. If Oo>k/30 proceed to Step 5.

b. Ifcco<k/3Othen: **Set** 4 <— 4 + 1. If *l*\ > (n - m)/2 proceed to Step 5. Set m<—m+1. For period i*:

b.l. Solve (3.9) and (3.10) to get the new matrices ZAi*, ZBI*.

b.2. Goto Step 3.

Step 5. Perform a line search using an augmented Lagrangian based merit function (Biegler and Cuthrell, 1985) to get a step size **a** starting from OCQ (0 £ **a** £ OQ). Set xjk+l = x£ + **a** Axi* and d*+l = d* + **a** Ad*. Set k <— k+1 and return to Step 2.

The above steps can be schematically seen in Figure 2. This algorithm was implemented in FORTRAN and was used to solve several multiperiod problems. The code used for solving the quadratic subproblems was QPSOL (Gill et al.$_9$1983).

4. Method Remarks

From a theoretical standpoint the SQP/MPD method described above has a global convergence property under mild conditions. This is the result of the local convergence properties associated with Newton-based methods combined with the minimization of an appropriate merit function through a line search that ensures global convergence to a local optimum. Note that although the proposed decomposition method greatly affects the computational characteristics of the solution process, it does not affect the convergence properties associated with the underlying SQP method.

One important characteristic of this method is the use of an incomplete solution to the QP subproblem (P.3) as the search direction for the merit function. Here each

subproblem is solved to optimality assuming that the current active set is correct. This early termination of the QP subproblems, regardless of the possible inefficiency that it may introduce, is needed to enforce a strict bound on the number of QP subproblems that need to be solved at each SQP iteration. This bound is important in practice, since the number of solutions of the QP subproblem (P.4) and (3.10) needed to solve (P.3) may be very high.

From a complexity standpoint, the computational demands for this method have a linear relation to the number of periods considered. Given a correct active set, the main computational effort is linear to the number of periods, as can be seen from the structure of the algorithm in Section 3.5. However, this property cannot be proved due to the NP-hardness associated with the general NLP formulation of the problem (Bellare and Rogaway, 1992). The complexity of the problem before the decomposition is NP-hard in N-n variables, C(N-n), where n can be any measure of the input variables. Through the decomposition we solve N problems with NP-hard complexity in n variables, C'(n). If the structure of C is similar to C* then the complexity of the new problem is N-C(n). Hence, if the above assumption holds, there is a reduction in complexity for the proposed method.

Another important characteristic of the SQP/MPD method is its embedded parallelism. As a matter of fact, the only non-parallel step in this algorithm is the solution of the projected QP which is the common coordination step for all the periods. This property can be utilized in a parallel computing environment to further decrease computational demands.

A drawback in the way this method is implemented is the approximation of the first term in the objective function of the projected QP in (P.4). This approximation affects the accuracy of the Hessian estimate and therefore is expected to increase the number of iterations of the method. One way to avoid this problem is by constructing or approximating the full Hessian for each period provided that the stability issue of this approximation is also addressed.

In this work the handling of bounds was done through an active set scheme described earlier. However, this is not the only alternative. A different approach includes the use of penalty or barrier methods. An exact penalty method was investigated in this work but the results were not satisfactory especially for large problems mainly due to the singularity in the Hessians introduced by the penalty related terms in the objective function. Another potentially promising approach, that was not investigated in this work, is the use of barrier methods which became very popular lately; it is not clear, though, whether the problems related to numerical stability as well as the choice of the barrier parameter have been successfully addressed.

5. Example Problems

In order to illustrate the proposed method we consider four example problems. The first two problems are general mathematical formulations of multiperiod models and are considered for examining the general trends of the method The other two are problems related to multiperiod process design in chemical engineering. The first is a small multiperiod flowsheet design problem involving a CSTR and a heat exchanger (Grossmann and Halemane, 1982) as seen in Figure 3. The goal is to decide on the volume of the reactor and the area of the heat exchanger, providing feasible and optimal operation for all periods. The other process design example addresses the optimal design of a heat exchanger network with fixed topology consisting of four units (Floudas and Grossmann, 1986) as seen in Figure 4.

5.1 Multiperiod Example Models A and B

In order to investigate the generality of this method on problems with different numbers of degrees of freedom as well as problems that are tightly or lightly constrained, we consider two small nonlinear multiperiod example problems. Although chemical process design problems have typically few degrees of freedom at the optimum, we constructed these two problems for the purpose of testing the method in a more general multiperiod NLP frameworic The model of problem A is:

(a) objective function

$$\text{minimize} f = d^2 + \sum_{i=1}^{\wedge}(a' e'^4 - 5x_2^l + *f) \qquad \text{(E.I)}$$

(b) equality constraints

$$A + P\%^2 - 2JC'_3 - 5d - 2 = 0$$

(c) inequality constraints

$$-x_2^i - \gamma^i x_3^i + 0.1<^{*2} \leq 0$$

$$1 \leq d \leq 50$$

Model B is a constrained version of (A). It is formed by the addition of upper and lower bounds to all the state and control variables:

(d) state and control variable bounds

$$0 \leq x\backslash \leq 3$$

$$0 \leq x\backslash \leq 3 \qquad \text{(E.2)}$$

$$0 \leq 4 \leq 3$$

The data for the parameters in this model (a^1, (3^1,7*) for some of the periods are listed in Table I. The maximum number of periods examined in this problem was 20. The total number of design and state and control variables for this problem are 1 + 3N (1 and 3N respectively) into 2N equality and inequality constrains (N and N respectively, excluding simple bounds on the variables)- The total number of the degrees of freedom for this problem based on the equality constraints is 1 + 2N. The problem sizes for models A and B for different number of periods are presented in Tables II and HI, respectively.

Using the above models A and B solving up to 20 period problems, the performance of the proposed MPD/SQP method is compared to a reduced gradient method (MINOS 5.2, Murtagh and Saunders, (1985)) and a reduced Hessian successive quadratic programming method (SQP, (Vasantharajan, Viswanathan and Biegler, 1990)). The results for models A and B, summarized in Tables IV and V respectively, show that the proposed method is superior to both methods in terms of computational efficiency. The SQP/MPD method, as anticipated from its algorithmic properties, presents a linear increase of the CPU time requirements with respect to the number of periods. In both models the performance pattern shows a linear increase for MPD/SQP, a quadratic for MINOS and a cubic for SQP, as shown in Figures 5 and 6.

5.2 Flowsheet Example

In this example a chemical plant is to be designed to produce different products in N different time periods. The reaction for all products is assumed to be first order exothermic of the type A —> B. The reactor temperature is controlled by a recycle stream through the heat exchanger using cooling water. The objective is to minimize the total annualized cost consisting of the fixed and operating costs. The mathematical model describing this system will form the constraint space of our optimization problem (E.3):

(a) net cost (annualized investment cost + operating cost)

$$\text{minimize } C = 0.3\,(2304\,\hat{V}^{01} + 2912\,A^{06}) + \sum_{i-1}(2.2\,10"\,\dot{W} + 8.82\,10"\,\dot{F}) \qquad \text{(E.3)}$$

(b) reactor material balance

$$\frac{C - C}{C}$$

(c) reactor heat balance

$$(-\Delta H) \wedge F_i \frac{C_{A_0}^i - C_A^i}{\tau_c} = F_0^i c_i (r_i - r_0) + Q^i$$

(d, e) **heat exchanger heat** balance

$$Q^i = F_1^i c_p^i (T_1^i - T_2^i)$$
$$Q^i = W^i c_w (T_{w_2}^i - T_{w_1}^i)$$

(0 heat exchanger design

$$Q^i = A\, UAV_n^i$$

$$\Delta T_m^i = \frac{(T_1^i - T_{w_2}^i) - (T_2^i - T_{w_1}^i)}{\ln\left\{\frac{T_1^i - T_{w_2}^i}{T_2^i - T_{w_1}^i}\right\}} \cong \left\{\frac{(T_1^i - T_{w_2}^i)(T_2^i - T_{w_1}^i)(T_1^i - T_{w_2}^i + T_2^i - T_{w_1}^i)}{2}\right\}^{\frac{1}{3}}$$

(g) reactor design

$$\hat{V} - V^* \wedge 0$$

(h) bounds on conversion

$$0.9 \leq \frac{\quad}{C_{A_0}^i} \leq 1$$

(i) **minimum approach temperature for heat exchange**

(j) maximum reactor temperature, minimum **water** temperature, **logical** conditions

$$T_1^i \leq T_{1_{max}}^i$$
$$r_i \leq r_i \leq 356$$

In addition, all the variables involved in this design are positive. The values of all the parameters for some of the periods considered are given in Table VI. This problem involves a total of 2 + 9N design and state and control variables (2 and 9N respectively) and 9N equality and inequality constraints (6N and 3N respectively, excluding simple bounds on the variables or inequality constraints that can be expressed as simple bounds).

Including the bounds we have 2 + 9N + 12N = 2 + 2IN total constraints. The total number of the degrees of freedom for this problem based on the equality constraints are 2 + 3N. A list of the problem size for different number of periods is presented in Table VIL

The results, summarized in Table VIII, show that the proposed method is superior to both methods in terms of computational efficiency, robustness and potential for large multiperiod problems. The SQP/MPD method, as anticipated from its algorithmic properties and shown in the two previous examples, preserves the linear behavior of the CPU time requirements with respect to the number of periods. The computational trends as a function of the number of periods for all the methods are presented in Figure *1*. Regarding the issue of robustness, for problems with more than 5 or 10 periods MINOS failed to reach the optimum from the given starting point and required a starting point within 20% of the optimum to succeed. However, for larger size problems with 15 or 20 periods even this starting point was not sufficient for converging to a solution. The proposed method on the other hand, appears to be far less sensitive to the selection of the starting point. The optimal solution of this problem in terms of the total annualized cost and the design variables is presented in Table DC for different numbers of periods.

5.3 Heat Exchanger Network Example

In this example a heat exchanger network (HEN) is to be designed that includes three heat exchangers (A_h A_2, A3), one utility unit (A4) and four process streams, two hot and two cold. The additional cooling load in this network is provided through water in the fourth heat exchanger (A_4). The objective is to minimize the annual cost of this network consisting of the investment plus the operating (utility) cost The mathematical formulation of this multiperiod problem (E.4) is given below:

(a) net cost (annualized investment cost 4- operating cost)

$$\text{minimize } C = 900\ (<*+\ /\%*+*£*+\ A_4^{06}) + \wedge\wedge\ \overset{N}{£}(310\text{-}^{3}<£) \quad \text{(E.4)}$$

(b) energy balances

$$1.5\ (r_i - r_2) = Q_i$$
$$2.0\ (r_i - r_j)_i = Q_i$$
$$r_5 - r_6 = Q_2$$
$$2.0\ (563 - T_A) = Q_2$$
$$T_i - T_i, = Q_i$$
$$3.0\ (393 - T'_s) = Q_s$$
$$1.5\ (T'_2 - 350) = Q_i$$

(c) design equations

$$\leq U A_1 \Delta T^i_{\ln_1}$$

$$Q^i_2 \leq U A_2 \Delta T^i_{\ln_2}$$

$$Q^i_3 \leq UA, \Delta T^i_{\ln_3}$$

$$Q^i_c \leq U A_4 \Delta T^i_{\ln 4}$$

(d) feasibility inequalities

$$r^i_2 - r^i_3 \geq \Delta T_{\min}$$

$$r^i_6 - r^i_5 \geq \Delta T_{\min}$$

$$\wedge - \mathbf{n} \wedge \Delta T_{\min}$$

$$\text{It.} - 393 \geq \Delta T_{\min}$$

$$r^i_5 \leq 323$$

The term ATi_n represents the mean logarithmic temperature and is approximated by the formula **given** in **the** previous **model (E.1-Q, (Chen, 1987). The data for the parameters in this model (Tj, T_3, T_5, Tg) for some of the periods are listed in Table X. The temperature range for the cooling water is 300-320 (K). The maximum number of periods examined in this problem is 20. The total number of design and state and control variables for this problem are 4 + 8N (4 and 8N respectively) into 12N equality and inequality constrains** (7N and 5N respectively, excluding simple bounds on the variables or inequality constraints that can be expressed as simple bounds). Including the bounds we have 4 + 12N + 9N = 4 + 21N total constraints. The total number of the degrees of freedom for this problem based on **the** equality constraints are 4 + N. A list of **the** problem size for different number of periods is presented in Table XL

The proposed method was applied to the solution of this problem with **up** to 20 period **problems. The results along with** comparisons **with MINOS and SQP** are **presented** in Table XII. In a manner similar to the flowsheet example the proposed method outperforms the above methods retaining the linear increase of computational demands with the number of periods. The computational requirements are plotted for all three methods in Figure 8, attesting to the trends observed in all the previous examples. As shown in Figures 5 through 8 for all the examples considered, the computational requirements for MINOS and SQP increase quadratically and cubically, respectively, with the number of periods, while the increase is linear for the proposed method. From the robustness standpoint MINOS failed to solve problems with more than 12 periods from the given starting point For 15 and 20 period problems it was necessary to provide a starting point

very close to the optimal solution in order for MINOS to find the optimum. However, for this problem the results for SQP regarding function evaluations were better than MPD/SQP. This can be attributed to the very few degrees of freedom that this problem has in combination with the fact that at the optimum there are no degrees of freedom for the design variables. On the other hand, for the proposed method since all the degrees of freedom in the projected QP in d are consumed it takes more iterations to find the correct set of projected bounds. In general though, even at the level of function evaluations, SQP/MPD outperforms the reduced space SQP method, as can be seen from the rest of the example problems. The optimal solution of this problem in terms of the total annualized cost and the optimal sizes of the heat exchangers is presented in Table XIII for different numbers of periods.

6. Conclusions

We have proposed a new decomposition method for solving multiperiod design optimization problems, based on successive quadratic programming. Our approach differs from previous approaches because it scales linearly to the number of time periods. This property allows the solution of large multiperiod problems considering a realistically high number of periods. Its performance is superior to general purpose optimization methods and overcomes the rapid increase in computational demands and convergence difficulties as a function of the time periods. Based on the algorithmic properties and some preliminary results, this method is more efficient in terms of function evaluations, computational time requirements and robustness compared to both a general purpose SQP method and the reduced gradient method MINOS.

The theoretical convergence properties for this method are in principle the same as for a standard reduced Hessian SQP method. Unlike standard SQP, however, the performance of this method is not affected by the total number of degrees of freedom which in turn depends on the number of periods. This comes as a natural result of the decomposition since the problem is addressed for the most part as a stream of parallel single period problems. Although the method's behavior regarding computational demands can be explained based on the decomposition arguments, the efficiency with respect to the function evaluations is not self evident The main reason for this efficiency comes from the fact that the Hessian estimates are independent for each period, and therefore more accurate, as opposed to uniform estimates for the (full or reduced) Hessian of the full multiperiod problem.

Directions for further improvement of this method should point towards a better Hessian approximation at the level of the projected QP that would decrease the number of major iterations. At the algorithmic level, addressing the issue of an efficient initialization of the Hessian would further increase the efficiency of the method

7. References

Aronson J. E_M Forward Linear Programming. Ph.D. Thesis - Carnegie Mellon University, 1980

Bellare M and P. Rogaway, The Complexity of Approximating a Nonlinear Program. IBM Research Report RC 17381 (#78493), 1992

Biegler L. T. and J. E. Cuthrell, Improved Infeasible Path Optimization for Sequential Modular Simulators. *Comp. Chem. Engng.* (1985) 9,257

Chen J. J. J_M Letter to the Editors: Comments on Improvement on a replacement of a Logarithmic Mean. *Chem Eng. Sci.* (1987) 42,2488

Gill P. E., Murray W., Saunders M. A. and M. H. Wright, User guide for SOL/QPSOL: A Fortran package for quadratic programming, Report SOL 83-7, Department of Operations Research, Stanford University, Stanford, California (1983)

Grossmann I. E. and C. A. Houdas, Active Constraint Strategy for Flexibility Analysis in Chemical Processes. *Comp. Chem. Engng.* (1987) 11,675

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

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

Han S. P., A globally Convergent Method for Nonlinear Programming. /. *Optimization Theory Control* (1977) 22, 297

Prieto F. J., Sequential Quadratic Programming Algorithms for Optimization. Ph.D. Thesis - Stanford University, 1991

Tjoa I. B. and L. T. Biegler, A Reduced Successive Quadratic Programming Strategy for Emors-in-Variables Estimation. *Comp. Chem. Engng.* (1992) 16,523

Varvarezos D. K., Grossmann I. E. and L. T. Biegler, An Outer-Approximation Method for Multiperiod Design Optimization. *Ind. Eng. Chem. Res.* (1992) 31,1466

Vasantharajan S., Viswanathan J. and L. T. Biegler, Reduced Successive Quadratic Programming Implementation for Large-Scale Optimization Problems with Smaller Degrees of Freedom. *Comp. Chem. Engng.* (1990) 14,907

List of Tables

Table I. Data for some of the periods for the example problems A and B (E.1), (E.2).

Period	1	5	10	15	20	Range
a	1.0	5.0	10.0	5.5	0.5	[0.5- 10.0]
p	1.0	7.0	8.0	7.5	8.5	[1.0- 9.2]
y	L0	6J)	1.0	6.5	7.5	[1.0- 8.5]

Table H Problem sizes for different number of periods in example model A (E. 1)

Number of periods	Total number of variables	Number (*) of constraints	Total number 0>) ofconstraints	Degrees <> of freedom
1	4	2	2	3
2	7	4	4	5
5	16	10	10	11
10	31	20	20	21
15	46	30	30	31
20	61	40	40	41

(t) Refers to equation and inequality constants eratadmg simple bounds
(b) Refers to equation, inequality constraints and simple bounds
(c) Defined as the total number of variables minus the number of equations

Table BL Problem sizes for different number of periods in example model B (E. 1 and E.2)

Number of periods	Total number of variables	Number(a) ofconstraints	Total number $) ofconstraints	Degrees <) offreedom
1	4	2	10	3
2	7	4	18	5
5	16	10	42	11
10	31	20	82	21
15	46	30	122	31
20	61	40	162	41

(a) Refers to equation and inequality constraints excluding simple bounds
(b) Refers to equation, inequality constraints and simple bounds
(c) Defined as the total number of variables minus the number of equations

Table IV. Computational results for example problem A (E.1).

Number of	MINOS		SQP		SQP/MPD	
periods	FuncL(a)	- CPU (s)0>)	Functfo)	- CPU (s)^)	Funct.(a)	- CPU (s)0>)
1	108	1.6	17	0.9	11	0.3
2	130	2.1	22	1.5	12	0.4
5	160	3.4	43	5.7	16	0.8
10	199	6.2	49	18.4	25	1.8
15	222	**9.1**	51	**50.8**	23	2.0
20	237	12.5	(c)	(c)	22	2.4

<*) Refers **to total number of** function **evaluations**
fa) CPU time results on a VAX station 3200
<c) SQP failed to converge as the limit of 250 major iterations was exceeded

Table V, Computational results for example problem B (E. 1 and E.2).

Number of	MINOS		**SQP**		**SQP/MPD**	
periods	Funct(a)	**- CPU (s)0>)**	**Funct(a)-CPU(s)O>)**		FunctXa)	- CPU (s)(b)
1	40	1.0	11	**0.7**	11	**0.4**
2	78	1.4	12	**0.8**	17	0.6
5	123	2.6	18	2.2	16	0.8
10	143	4.2	29	8.5	15	1.2
15	**146**	**5.8**	35	21.7	16	**1.6**
20	**175**	8.4	40	**57.2**	25	3.1

(a) Refers to total number of function evaluations
(b) CPU time results on a VAX station 3200

Table VL Data for some of the periods for the flowsheet example problem (E3).

Period	1	5	10	15	20	Units
(EVR)	555.6	500.0	555.6	583.3	542.0	(K)
$-\Delta H_{rxn}^{1}$	23,260.0	18,604.0	23,260.0	25,581.0	27,907.0	(KJ/Kmol)
ν_0^i	10	8	10	10.5	12	(h-1)
F_0^i	45.36	54.43	45.36	40.82	36.29	(Kmol/h)
C_{A0}^{1}	32.04	32.04	28.80	10.05	48.06	(Kmol/m3)
c_p^i	167.4	125.6	169.4	188.4	209.7	(KJ/Kmol K)
T_{1max}^{i}	389.0	400.0	389.0	383.0	378.0	(K)
Additional Data:	$T_0 = 333$(K),		$T_{w1} = 300$(K),		ATmin = 11.1 (K),	
	U= 1,635.34 (KJ/m2Kh)					

Table VII. Problem sizes for different number of periods in the flowsheet example (E.3)

Number of periods	Total number of variables	Number (a) of constraints	Total number (b) of constraints	Degrees (c) of freedom
1	11	9	23	5
2	20	18	44	8
5	47	45	107	17
10	92	90	212	32
15	137	135	317	47
20	182	180	422	62

(a) Refers to equation and inequality constraints excluding simple bounds

(b) Refers to equation, inequality constraints and simple bounds

(c) Defined as the total number of variables minus the number of equations

Table VIII. Computational results for the flowsheet example problem (E.3).

Number of	MINOS		SQP[d]		SQP/MPD	
periods	Funct.(a)	CPU (s)[b]	Funct.[f]	CPU (s)[b]	Funct.[g]	CPU (s)[b]
1	441	6.6	16	1.8	25	1.7
2	486	9.7	16	2.9	17	1.7
5	355	13.6	10	6.2	24	4.5
10	588	37.7	20	32.9	23	8.1
15	767	61.8	45	165.8	38	19.5
20	(c)	(c)	44	302.1	30	21.8

(*) Refers to total number of function evaluations
(**) CPU time results on a VAX station 3200
(°) MINOS failed to converge from the initial starting point **and a** set of points in its neighborhood
(••) A different starting point closer to the solution was required

Table IX. Optimal solution for the flowsheet example problem (E.3) considering different numbers of periods.

Number of periods	Reactor Volume (m3)	Heat Exchanger Area (m2)	Total Cost ($/yr.)
1	5.315	7.544	9,731
2	5.315	8.517	10,071
5	7.927	8.614	10,689
10	7.927	8.163	10,573
15	7.927	8.518	10,715
20	7.927	8.949	10,907

Table X, Data for some of the periods for the HEN example problem (E.4).

Period	1	5	10	15	20	Range
Ti(K)	620	620	620	630	635	[620-670]
T_3(K)	388	388	373	370	375	[370-390]
T_5(K)	583	583	583	570	585	[570-593]
T_8(K)	313	308	313	312	312	[308-314]

Additional Data: AT_{min} = 10 (K) U = 0.4 (KJ/m2 K s)

Table XL Problem sizes for different number of periods in the HEN example (E.4)

Number of periods	Total number of variables	Number[a] of constraints	Total number[b] of constraints	Degrees[c] of freedom
1	12	12	25	5
2	20	24	46	6
5	44	60	109	9
10	84	120	214	14
15	124	180	319	19
20	164	240	424	24

(a) Refers to equation and inequality constraints excluding simple bounds
(b) Refers to equation, inequality constraints and simple bounds
(c) Defined as the total number of variables minus the number of equations

Table XII. Computational results for the HEN example problem (E.4).

Number of periods	MINOS Funct[a]	MINOS CPU (s)[b]	SQP Funct[a]	SQP CPU (s)[b]	SQP/MPD Funct[a]	SQP/MPD CPU (s)[b]
1	78	2.1	5	1.3	8	0.7
2	120	3.7	7	2.2	9	1.1
5	215	10.0	10	6.0	23	4.8
10	168	15.3	5	14.4	18	6.5
12	259	24.3	7	20.1	17	7.3
15	(c)	(c)	7	30.0	23	12.9
20	(c)	(c)	7	46.2	24	17.4

(*) Refers to total number of function evaluations
(**) CPU time results on a VAX station 3200
(c) MINOS failed to converge from the initial starting point and a set of points in its neighborhood

Table XIIL Optimal solution for the HEN example problem (E.4) considering different numbers of periods.

Numberof periods	HE-Ai (m^2)	HE-A_2 (m2)	HE-A3 (m2)	HE-A4 <m2)	TotalCost ($/yr.)
1	26.052	3.467	11.218	2.938	15,618
2	26.052	3.467	11.218	2.938	15,840
5	31.750	3.467	13.953	3.127	17,007
10	38.068	6.962	14.212	3.533	18,951
15	39.933	6.962	14.212	4.206	19,560
20	39.933	6.962	14.212	4.206	19,500

List of Figures

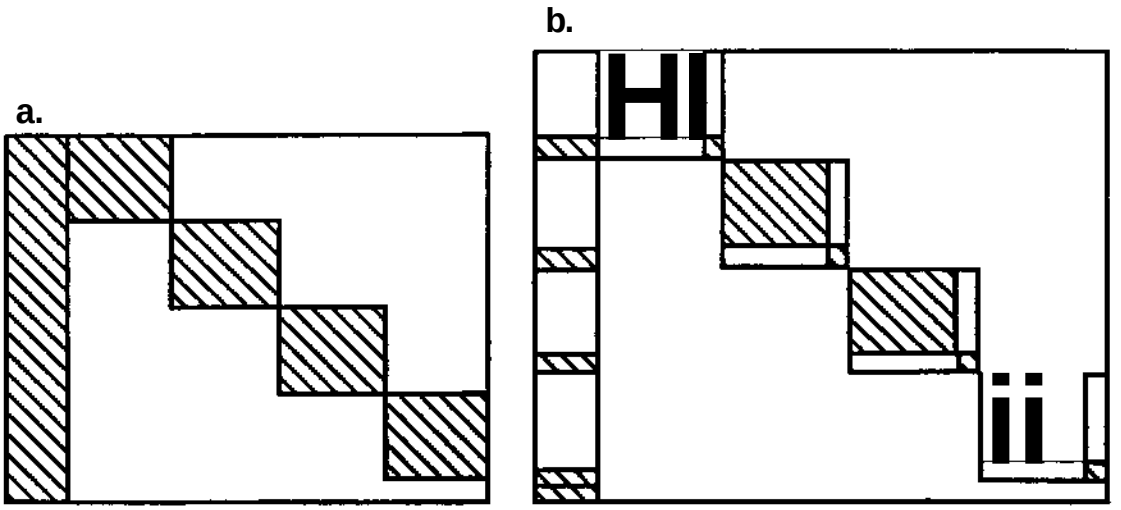

Figure 1. Structure of the multipenod models: a. before any decomposition, b. after the proposed decoupling scheme

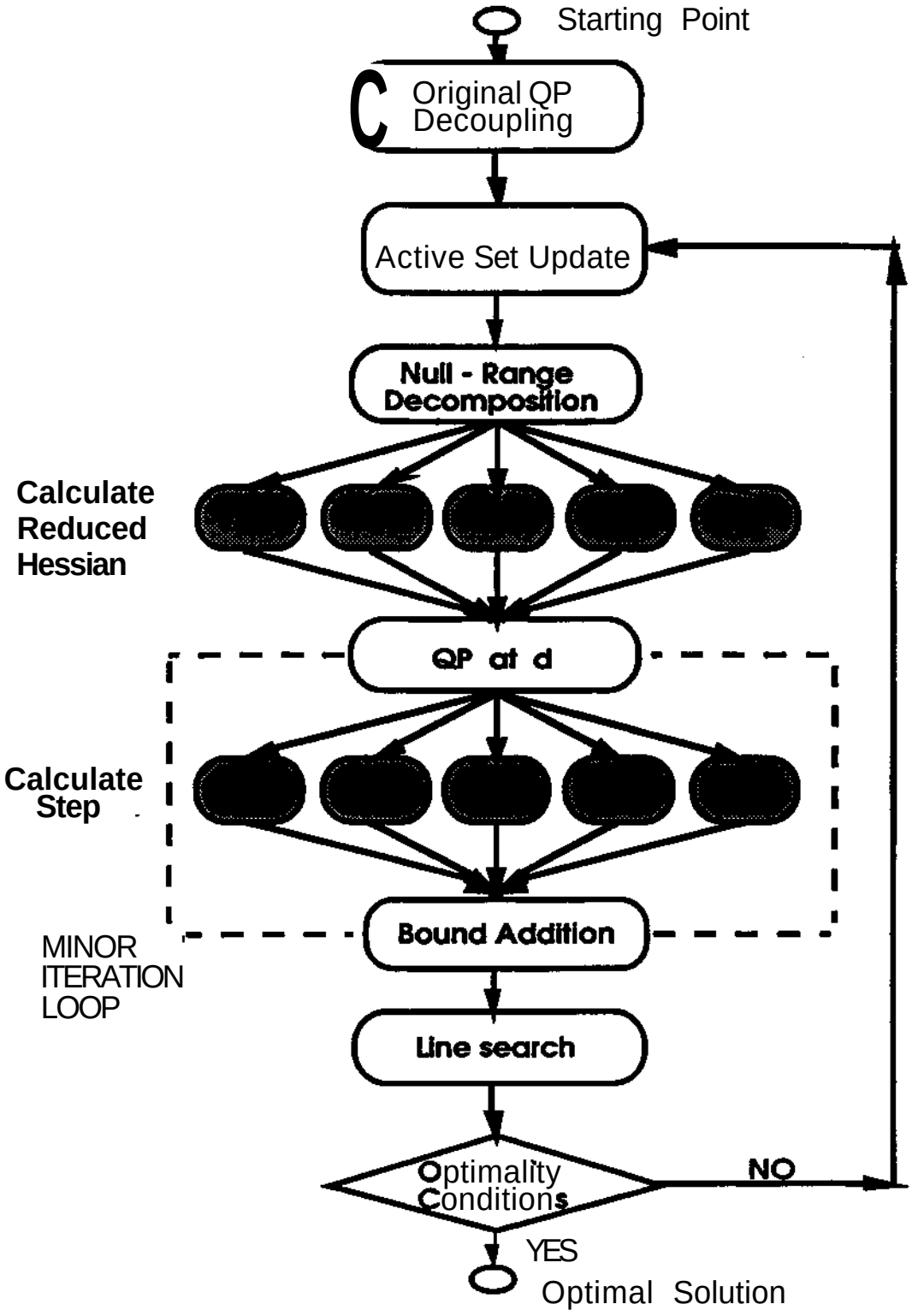

Figure 2. Schematic diagram of the MPD/SQP Algorithm

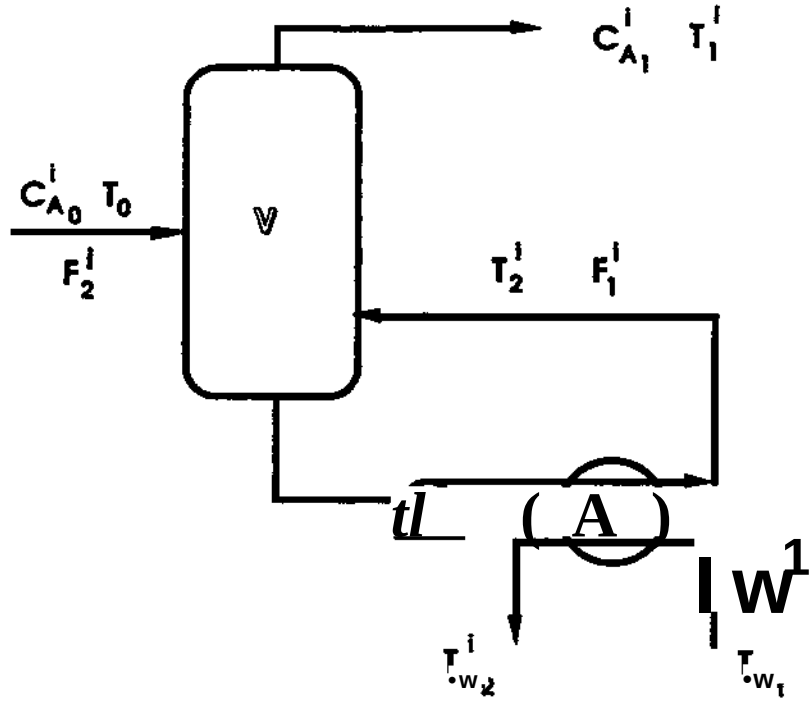

Figure 3. Flowsheet diagram in example problem (E3)

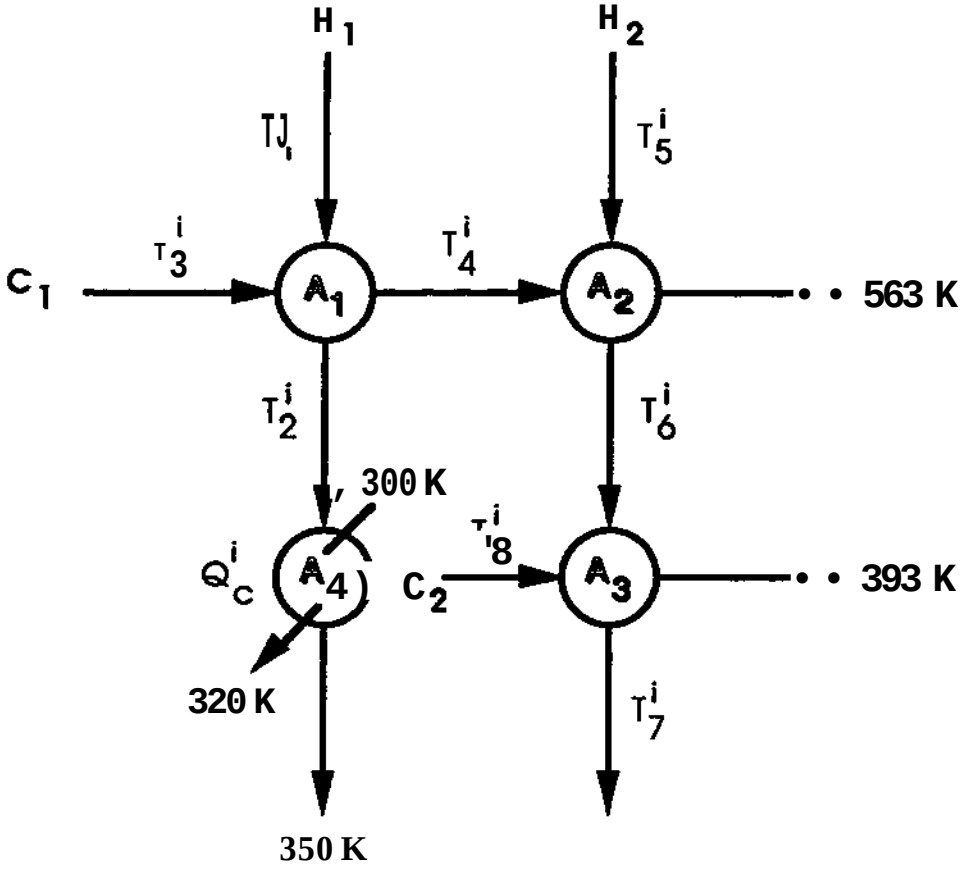

Figure 4. Heat exchange network in example problem (E.4)

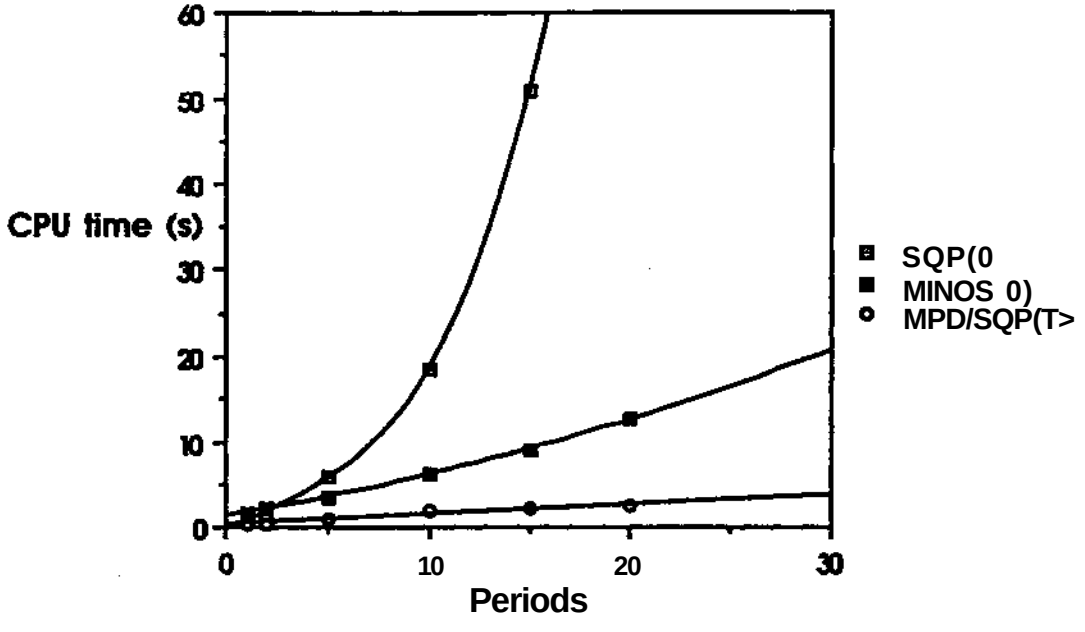

Figure 5. Comparison of the computational requirements for model A (E. 1) as a function of different number of periods

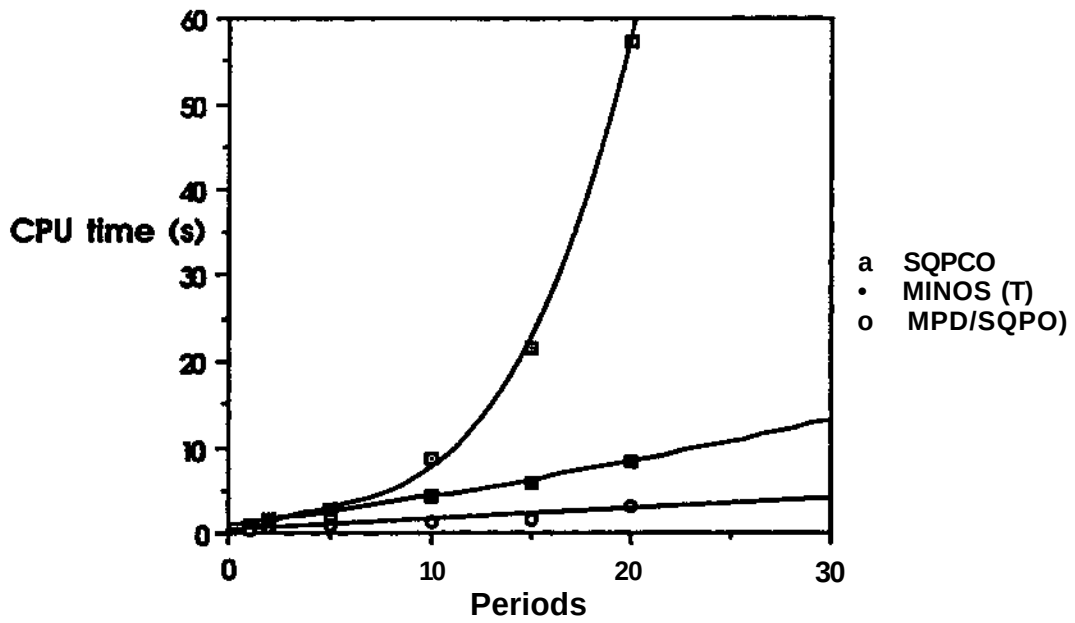

Figure 6. Comparison of the computational requiiements for model B (E2) as a function of different number of periods

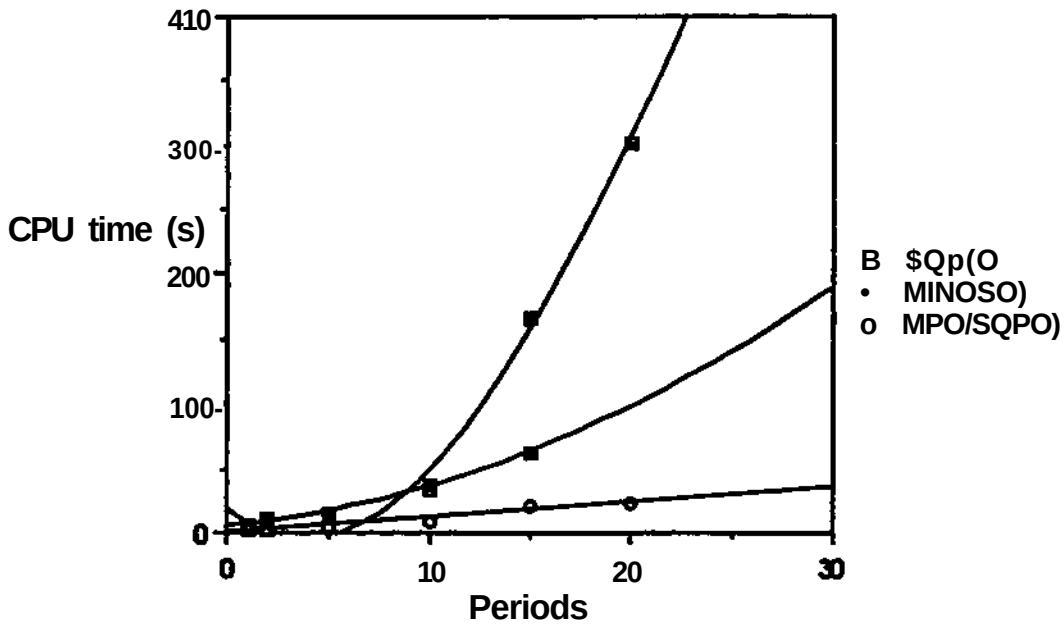

Figure 7. Comparison of the computational requirements for the flowsheet example problem as a function of different number of periods

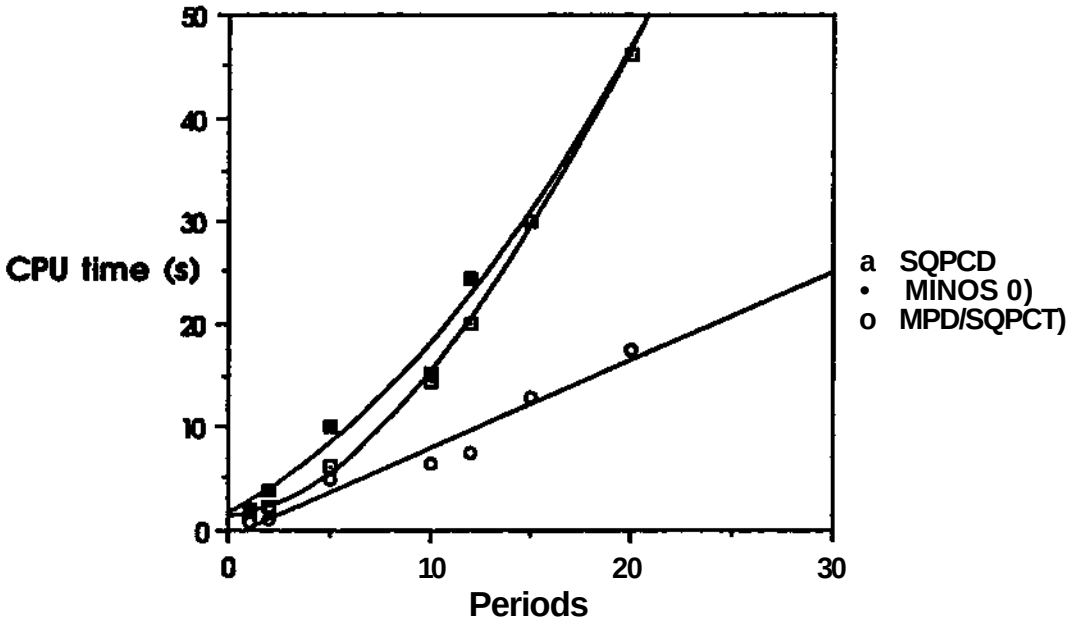

Figure 8. Comparison of the computational requirements for the HEN problem as a function of different number of periods