A DECOMPOSITION STRATEGY FOR DESIGNING FLEXIBLE CHEMICAL PLANTS

by

Ignacio E. Grossmann and Keshava P.rasad Halemane

DRC-06-20-81

May 1981

A DECOMPOSITION STRATEGY FOR DESIGNING FLEXIBLE CHEMICAL PLANTS

Ignacio E. Grossmann and Keshava P# Halemane
Department of Chemical Engineering
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

May 1981

Correspondence concerning this paper should be addressed to
Ignacio E, Grossmann

Abstract

One of the main computational problems faced in the optimal design of flexible chemical plants with multi-period operation is the large number of decision variables that are involved in the corresponding nonlinear programming formulation. To overcome this difficulty, a decomposition technique based on a projection-restriction strategy is suggested to exploit the block-diagonal structure in the constraints. Successful application of this strategy requires an efficient method to find ah initial feasible point, and the extension of current equation ordering algorithms for adding systematically inequality constraints that become active. General trends in the performance of the proposed decomposition technique are presented through an example.

Scope

Flexibility in chemical plants is normally introduced in practice by applying empirical overdesign factors to sizes of equipment that have been designed for a nominal operating condition. This procedure is clearly not very satisfactory as it has little rational basis. For instance, with empirical overdesign it is unclear what range of specifications the overdesigned plant can tolerate, or whether the plant is optimal according to a given criterion. Therefore, there is clearly a need for developing design methods that deal with flexibility in a more rational way.

Two classes of problems can be considered in the design of flexible chemical plants. The first one is the deterministic multi-period problem wherein the plant is designed to operate under various specified conditions in a sequence of time periods. Typical examples are refineries that handle various types of crudes, or pharmaceutical plants that produce several products. The second type of problem deals with the design of chemical plants where significant uncertainty is involved in some of the parameters. Examples of this case arises when values of feed specifications, transfer coefficients, physical properties or cost data are not well established. It must be noted that a design problem can also be a combination of these two distinct types of problems.

As has been shown by Grossmann and Sargent (1978, 1979) the two classes of problems require the solution of a nonlinear programme where the number of decision variables and constraints can become rather large. It is the purpose of this paper to present an efficient solution procedure for the deterministic multi-period problem, which in fact can also be applied for solving the problem of design under uncertainty (Halemane and Grossmann, 1981). It is shown that the block-diagonal structure in the constraints, and the fact that many inequalities become active at the solution, can be exploited effectively for reducing the computational requirements in the nonlinear programming problem.

Conclusions and Significance

It has been shown that the proposed projection-restriction strategy exploits effectively the mathematical structure of the multi-period problem for designing flexible chemical plants. The performance of the decomposition method as seen in the example shows the most encouraging feature that computational time only increases moderately with the number of periods in a problem. Also, the reliability for obtaining the desired solutions is greatly enhanced. Therefore, the proposed method provides a real possibility for tackling large-scale optimizations of flexible plants with a reasonable computational effort.

Introduction

A general approach based on nonlinear programming (NLP) for the design of flexible chemical plants has been proposed by Grossmann and Sargent (1978, 1979). The main features of their formulation are as follows.

In the case of the deterministic multi-period problem it is assumed that the plant is subjected to piecevise constant operating conditions in N successive time periods. Dynamics are neglected, as it is considered that the length of the transients is much smaller than the time periods for the successive steady-states. The variables in this problem are partitioned into three categories. The vector d of design variables is associated with the sizing of the units. These variables remain fixed once the design is implemented, and do not vary with the changes in the operation of the plant. The vector z^i denotes the control variables that can be manipulated in each period i so as to meet the specifications and also minimize the operating cost. It should be noted that the vector z^i corresponds to a given assignment of variables to the existing degrees of freedom in the operation of the plant. Finally, the vector x^i corresponds to the state variables in the operating period i (i=l,2,...N). Thus, the design problem leads to the nonlinear programme,

$$\underset{d,z^1,z^2,\ldots z^N}{\text{minimize}} \quad C = C(d,z^1,z^2,\ldots z^N,x^1,x^2,\ldots x^N)$$

$$\text{s.t.} \quad \left.\begin{array}{l} h\hat{}d.E^1,*^1) \;=\; 0 \\ \;i \quad\; i \;\; i \\ g^1(d,z^1,x^1) \;\leq\; 0 \end{array}\right\} \quad i - 1,2,\ldots N \qquad (1)$$

$$\pounds(d_f z^1{}_f z^2,\ldots z^N{}_f x^1{}_f x^2 \ \ldots x^N) \leq \quad 0$$

vh •••* d, z^x_i i=l,2,...N, are the decision variables in this problem, as the state va ables x , i=l,2,...N, can be determined from the equality constraints which represent the steady-state equations of the process. Note that in this formulation the order in which the periods are considered can be arbitrary, since the operation in each period is assumed to be independent of its relative position in the sequence. However, any specificat   3 involving all the periods can be represented by the last inequality constraint.

As for the design problem under uncertainty, consider that 9 is the vector of uncertain parameters. Assinning bounded values of these para* meters, $0^L \leq 0 \leq 0^U$, a design strategy may be considered based on the following reasoning: The initially installed plant may be defined by the vector of design variables d. During the operation of the plant, the control varialbes z would be adjusted, depending on the values of the parameters 9 being realized, so as to meet the design specifications. Hence the purpose of design is in selecting d such that the plant will be able to meet the specifications for all possible realizations of the parameters 9, while minimizing the expected value of an appropriate cost function. This may be represented as a two-stage programming problem,

$$\underset{d}{\text{minimize}}\ \underset{9}{E}\left\{\underset{z}{\min}\ C(d,z,x,9)\ \middle|\ \begin{array}{l} h(d,z,x,9) = 0 \\ g(d,z,x,9) < 0 \\ f(d) \leq 0 \quad \sim \end{array}\right\} \qquad (2)$$

$$\text{s.t.}\quad V\,9,\ 9^{\wedge} \leq 9 \leq 9^{U},\ 3z\ :\ \begin{array}{l} h(d,2,x,9) = 0 \\ g(d,z,x,9) \leq 0 \end{array}$$

A direct approach to the solution of (2) poses the problem of infinite dimensions in 9, demanding a tremendous computational effort. However, this difficulty can be overcome by using a discretization procedure (Grossmann and Sargent, 1979; Halemane and Grossmann, 1981), involving a finite number of points 9^1, i«l,2,...N, and still ensuring feasibility for all 9 within the specified bounds.This reduces problem (2) to a special case of problem (1), where the objective function is separable in the N points (or periods) and where there is no coupling of constraints for z^i, x^i, variables of different periods. That is, the problem has the form

$$\underset{d,z^1,z,\ldots z^N}{\text{minimize}}\quad C = C^{\circ}(d) + \sum_{i=1}^{n} C\hat{\ }d,z^{i},x^{i})$$

$$\text{s.t.}\quad \left.\begin{array}{l} h(d,z^{i},x^{i}) = 0 \\ g(d,z\backslash xi) \leq 0 \end{array}\right\}\ i = 1,2,\ldots N \qquad (3)$$

$$f(d) \leq 0$$

It should be noted that (3) is also the most common formulation encountered for the multi-period problems since, very often, there is no coupling of constraints for z^i, x^i, variables of different periods. Also, the usual cost functions are separable in the investment cost and operating cost for each period. Since solving the NLP in (1) is of fundamental importance for designing flexible plants, and problem (3) is a very important case having some interesting properties, this paper will address the problem on how to solve the latter NLP in the most efficient way.

Computational Aspects

For large industrial problems the computational requirements for solving the NLP in (3) can become rather expensive. The main reason for this is that the number of control variables z^i increases with the number of periods N, so that the number of decision variables in the NLP may become too large to be solved efficiently by the current algorithms. Since the NLP approach for designing flexible plants has proved to be very effective in small problems, and it also provides rational basis for overdesign, there is a very high incentive for deriving an efficient method for solving problem (3). This requires that its mathematical structure be fully exploited.

In order to take advantage of the sparsity of the constraints, state variables x^i can be eliminated from the system of equations so as to reduce the size of the problem. This can be achieved if the system of equations is ordered so as to provide a sequence of calculation where the number of torn variables is minimized (see e.g. Christensen, 1970). In this scheme, at each iteration of the optimization the ordered system of equations is solved. It must be pointed out, however, that by eliminating the equations and state variables the nonlinear programme in (3) still has to handle the large number of decision variables given by d, z^1, i = 1,2,...N. Therefore, it is necessary to exploit additionally another property of (3) for deriving an efficient method of solution.

Problem (3) has the interesting feature that it is an NLP with block-diagonal structure in the constraints, as shown in Figure 1. Since the cost function is separable in the N periods this implies that if the vector d is fixed, then the problem decomposes in N uncoupled subproblems, each having as decision variables the vectors z^i, for i * 1,2,...N. This would suggest that it should be possible to derive a suitable decomposition scheme which need not handle simultaneously all the decision variables. Ideally, this decomposition scheme should lessen the storage requirements, and more importantly it should reduce substantially the computational time for obtaining the optimal solution.

Decomposition Strategies

The two basic decomposition strategies that can be used for solving problem (3) are the feasible and infeasible decomposition schemes.

The feasible decomposition technique (Rosen and Ornea, 1963; Umeda, Shindo and Tazaki, 1972) consists of the following steps:

Step 1 - Find a feasible point d, z^1, x^1, i=l,2,...N, for problem (3).

Step 2 - By keeping the vector d fixed, solve the N subproblems (that is, for i=*l,2,...N):

$$\begin{aligned} &\underset{z^1}{\text{minimize}} \quad C^x(d,z^1,x^1) \\ &\text{s.t.} \quad h\text{^}d.z^1,*^1) = 0 \qquad\qquad (4) \\ &\qquad\quad gV.zV)\leq 0 \end{aligned}$$

Step 3 - Keeping the vectors z^1, 1=1,2,...N fixed, solve the problem:

$$\begin{aligned} &\underset{d}{\text{minimize}} \quad C = C^{o}(d) + \sum_{i=1}^{N} C^{i}(d,z^{i},x^{i}) \\ &\text{s.t.} \quad \left.\begin{aligned} h^{i}(d,z^{i},x^{i}) &= 0 \\ g^{i}(d,z^{i},x^{i}) &\leq 0 \end{aligned}\right\} \; i=1,2,\ldots N \\ &\qquad\;\; f(d) \leq 0 \end{aligned} \tag{5}$$

Step 4 - If convergence is not achieved, return to step 2.

The advantage with this technique is that the original problem is replaced by a sequence of subproblems with a smaller number of decision variables. However, convergence to the solution can become extremely slow particularly in the neighborhood of the solution (see Grigoriadis, 1971), since in fact this decomposition technique is equivalent to an alternate search in orthogonal directions in the space $(d,(z^{1},z^{2},\ldots z^{N}))$.

In the infeasible decomposition technique (Brosilow and Lasdon, 1965; Lasdon, 1970; Stephanopoulos and Westerberg, 1975b) it is first necessary to reformulate problem (3) as:

$$\begin{aligned} &\underset{\substack{\hat{d},d^{1},d^{2},\ldots d^{N}, \\ z^{1},z^{2},\ldots z^{N}}}{\text{minimize}} \quad C = C^{o}(\hat{d}) + \sum_{i=1}^{N} C^{i}(d^{i},z^{i},x^{i}) \\ &\text{s.t.} \quad \left.\begin{aligned} h^{i}(d^{i},z^{i},x^{i}) &= 0 \\ g^{i}(d^{i},z^{i},x^{i}) &\leq 0 \\ f^{i}(d^{i}) &\leq 0 \\ \hat{d} &= d^{i} \end{aligned}\right| \; i=1,2,\ldots N \\ &\qquad\;\; f(\hat{d}) \leq 0 \end{aligned} \tag{6}$$

Since the lagrangian of this problem is given by:

$$\begin{aligned} L = {} & C^{o}(\hat{d}) + \sum_{i=1}^{N} C^{i}(d^{i},z^{i},x^{i}) \\ & + \sum_{i=1}^{N} \left[(\lambda^{i})^{T} h^{i} + (\mu^{i})^{T} g^{i} + (\nu^{i})^{T} f^{i} + (\pi^{i})^{T} (\hat{d} - d^{i})\right] \\ & + \rho^{T} f \end{aligned} \tag{7}$$

where \ ^{i}, y, ^{i} , o^{i} » IT^{i} » p are the Kuhn-Tucker multipliers, problem (6) can be decomposed into the following N+l subproblems:

$$\text{minimize} \quad (\wedge(d^1,*^1,*^1) - (TT^1)^1 d^1$$

$$\text{s.t.} \quad n^{i}(a^{i},z^{i},x^{i}) = u$$

$$g^{i}(d^{i},z^{i},x^{i}) \leq 0$$

$$\wedge(d^1) \quad \leq 0 \qquad (8)$$

for i=l,2,...N, and

$$\underset{d}{\overset{A}{\text{minimize}}} \quad C\ (d) + \underset{i=l}{\overset{N}{>}} (FT^*)^{-} d$$

$$\text{s.t.} \quad f(\hat{d}) \leq 0$$

The infeasible decomposition strategy then consists of the following steps:

Step 1 - Guess the multipliers TT^1, i=l,2,...N.

Step 2 - Solve the N+l subproblems in (8).

Step 3 - If the constraints d = d^{Ai} , i=l,2,...N, are not satisfied, adjust the T by solving the dual problem:

$$\underset{TTITT2\,TX\wedge}{\text{maximize}} \quad C^{oA}(d) + \sum_{i=1}^{N} \Big[C^{Fi}(d^{i},z^{I},x^{i}) + (\pi^{iT})^{*}(d - d^{il}) \Big] \qquad (9)$$

and then return to step (2).

Note that in this decomposition technique it is not necessary to start with a feasible point in problem (3) as with the previous strategy. However, there are basically two difficulties when using this technique. The first one is that the method may not converge to the solution due to the non-convexities that are present in design problems which give rise to dual gaps. This difficulty can be overcome with the method proposed by Stephanopoulos and Westerberg (1975a), but with the disadvantage that it requires a significant amount of computational effort. A further disadvantage with the infeasible decomposition scheme is that a feasible solution is obtained only at the exact solution of the dual problem. Considering that one is dealing with nonlinear problems, this can become a significant drawback in practice.

With the two decomposition schemes that have been presented above it is unclear whether problem (3) can be solved more efficiently than when the problem is tackled with all decision variables simultaneously. It is for this reason that an alternative decomposition scheme must be considered.

The Projection-Restriction Strategy

Grigoriadis (1971) and Ritter (1973) have suggested a decomposition technique for the case when the objective function is convex and the constraints are linear in problem (3). As per the classifications given by Geoffrion (1970) this strategy belongs to the class of Projection-Restriction Strategies. The basic ideas are described in the following steps:

Step 1 - Find a feasible point d, z^{i}, x^{i}, i=1,2,...N, for problem (3).

Step 2 - (Projection)

Fixing the values of the vector d, solve the N subproblems in (4).

Step 3 - (Restriction)

(a) For each subproblem i convert the n_A^* inequality constraints g_A^* that are active in step 2 into equalities, and define

$$h_R^i = \begin{bmatrix} h^i \\ g_A^i \end{bmatrix} , \quad g_R^i = g_I^i \qquad i=1,2,...N \tag{10}$$

where h_R^i, g_R^* are the redefined sets of equality and inequality constraints and g_I^i are the sets of inequality constraints that are not active in step 2.

(b) Eliminate n_A^* variables z_A^* from the vector

[illegible] o as to define

$$z_R^i = z_I^i \quad , \quad x_R^i = \begin{bmatrix} x^i \\ z_A^i \end{bmatrix} \quad , \quad i=1,2,\ldots N \tag{11}$$

where z_R^i is the redefined vector of control variables which results from eliminating the vector z_A^i of n_A^i elements, and x_R^i is the expanded vector of state variables.

Step 4 - Solve the restricted problem:

$$\begin{aligned} \underset{d, z_R^1, z_R^2, \ldots z_R^N}{\text{minimize}} \quad & C = C^0(d) + \sum_{i=1}^{N} C^i(d, z_R^i, x_R^i) \\ \text{s.t.} \quad & \left.\begin{aligned} h_R^i(d, z_R^i, x_R^i) &= 0 \\ g_R^i(d, z_R^i, x_R^i) &\leq 0 \end{aligned}\right\} \quad i = 1, 2, \ldots N \\ & f(d) \leq 0 \end{aligned} \tag{12}$$

Step 5 - Return to step 2 and iterate until

(a) no further changes occur in the values of the variables d, or

(b) the same set of inequality constraints become active again, in step *2.*

Note that in step 4 the projection-restiction strategy really consists in solving problem (3) simultaneously for all variables, but in general with a smaller number of decision variables, since many of these get eliminated by the active constraints determined in step 2. Clearly, the effectiveness of this strategy relies heavily on the number of inequality constraints that actually become active at the solution.

Grigoriadis (1971) and Ritter (1973) found that in their problems relatively few inequality constraints in the subproblems would become active*

Therefore, they proposed to eliminate all the variables z^i, i=l,2,...N, in step 3; Grigoriadis (1971) with the use of the pseudoinverse of the corresponding matrix of z^i, and Ritter (1973) with a square matrix which was generated when solving the subproblems. Unfortunately these techniques can not be extended readily to the case when constraints are nonlinear, since they rely heavily on the assumption of linearity of the constraints. However, as will be shown, the basic idea of the projection-restriction strategy can indeed be used for designing flexible chemical plants.

An examination of the results for flexible plants obtained by Grossmann and Sargent (1978, 1979) shows that a surprisingly large number of inequality constraints are actually active at the solution, as can be seen in Table 1. The main reason for this appears to be the monotonicity of the cost functions which is characteristic of design problems.

Since in general one can expect to have a large number of active constraints at the solution, it clearly suggests that the projection-restriction strategy can greatly simplify solving problems of the type as formulated in (3). However, for successful application of the projection-restriction strategy there are two problems that have to be considered. The first one is finding an initial feasible point in step 1. The second is a procedure for the elimination of variables in step 3 which avoids singularities in the system of equations. These points are discussed in the following sections.

Finding a Feasible Point

The problem of finding a feasible point for a design problem is in general a nontrivial task, because of the nonlinearities involved. In problem (3) the main difficulty when using the projection-restriction strategy consists in finding a value of d such that feasible solutions exist for the subproblems in (4).

One approach to find a feasible point is to replace the cost function in (3) by the sum of squares of deviations of the violated constraints, thus leading to the nonlinear programme:

$$\begin{aligned} \underset{d,z^1,z^2,\ldots z^N}{\text{minimize}} \quad & \Phi = \sum_{i=1}^{N} \sum_{j=1}^{m} \left[\max\left\{0,g_j^i(d,z^i,x^i)\right\}\right]^2 \\ \text{s.t.} \quad & h^i(d,z^i,x^i) = 0 \\ & g^i(d,z^i,x^i) \leq 0 \qquad i=1,2,\ldots N \\ & f(d) \leq 0 \end{aligned} \tag{13}$$

This problem can be handled by an NLP algorithm based on an active set strategy for the constraints as indicated by Sargent and Murtagh (1973). Since the objective function in (13) has discontinuous second order derivatives the optimization should be performed with the steepest descent direction in the constraint space. As this procedure does not require an estimation of the inverse of the Hessian matrix, storage requirements can be reduced.

Although solving (13) simultaneously for all variables works very well for relatively small problems, it may be desirable to use a decomposition scheme for large problems. An alternative is to use the steps similar to those in the feasible decomposition strategy, with the objective function as given in (13) above. Hence it consists of the following steps:

Step 1 - Guess a starting point $d,z^i,x^i,i=1,2,\ldots N$.

Step 2 - By keeping the vector d fixed, solve the N subproblems (i.e. for $i=1,2,\ldots N$):

$$\begin{aligned} \underset{z^i}{\text{minimize}} \quad & \Phi^i = \sum_{j=1}^{m} \left[\max\left\{0,g_j^i(d,z^i,x^i)\right\}\right]^2 \\ \text{s.t.} \quad & h^i(d,z^i,x^i) = 0 \\ & g^i(d,z^i,x^i) \leq 0 \end{aligned} \tag{14}$$

Step 3 - By keeping the vectors z^i, $i=1,2,\ldots N$ fixed, solve the problem:

$$\underset{d}{\text{minimize}}\ \Phi = \sum_{i=1}^{N} \sum_{j=1}^{m} \left[\max \{0, g_j^i(d, z^i, x^i)\} \right]^2$$

$$\text{s.t.} \quad \left. \begin{array}{l} h^i(d,z^i,x^i) = 0 \\ g^i(d,z^i,x^i) \leq 0 \end{array} \right\} \quad i=1,2,\ldots N \qquad (15)$$

$$f(d) \leq 0$$

Step 4 - If convergence is not achieved $(\Phi > 0)$ return to step 2.

It is observed that unlike in the case of finding the optimal solution of (3), the convergence of this method to find a feasible point is quite good. The reason for this is that the NLP defined by (13) has an infinite number of minima when the feasible space is non-empty. In this case the objective function defines a plateau of zero-value for the feasible region as shown in Figure 2, and hence there is usually no problem of slow convergence in the neighborhood of a feasible solution. Also, note in Figure 2 that outside the feasible region the contours of Φ in (13) are quadratic in the constraint functions so that the objective function will tend to be well behaved.

Variable Elimination in the Restriction Step

The elimination of variables in step 3 of the projection-restriction strategy is performed for each period i by including the active inequality constraints in the set of equations. This implies that from the state and control variables x^i, z^i a set of control variables z_K^i must be determined, and that the sequence of calculation for the new set of equations h_K^i has to be derived, for each period i = 1,2,...N.

Since a number of algorithms are available for selecting decision variables and determining sequences of calculation for rectangular systems (Lee et al., 1966; Christensen and Rudd, 1969; Edie and Westerberg, 1971; Leigh, 1973; Stadtherr et al., 1974; Book and Ramirez, 1976; Hernandez and Sargent, 1979) it would seem that they could be applied without difficulty in our problem of designing flexible plants. It must be pointed out, however, that difficulties may arise when deriving the solution procedure for the restricted problem (12) since the added inequality constraints can lead to redundant or inconsistent equations, and hence, produce a singular system of equations. Therefore, these algorithms must be extended according to the following procedure for eliminating the variables in each period i in the restricted problem:

Step 1 - Add all the active constraints $g^{\wedge}_{--}(d,z^{x},x^{X}) = 0$ to the system of equations $h^{x}(d,z^{i},x^{x}) = 0$, thus giving rise to a new system of equations $h^{i}_{K}(d,z^{i},x^{i}) = 0$.

Step 2 - Perform the optimal reordering of the new system of equations $h^{i}_{R}(d,z^{i},x^{X}) = 0$, by minimizing the number of torn variables in z^{i},x^{i}.

Step 3 - Select control variables $z^{i}_{K.}$ as decision variables, and delete equations if necessary so as to obtain a non-singular square system of equations.

It should be noted that due to the reordering of variables in step 2, the vector z^{1}_{R} can in fact contain some of the state variables from x^{1}. Also, it is essential to keep the design variables d as decision variables throughout, and not to force them to become either state or torn variables during the reordering. Also, since it is possible that the resulting system in step 2 has more equations than variables, a suitable equation ordering algorithm must be used, for instance the one by Leigh (1973). The optimal sequence determined with such an algorithm is one where the system $h^{i}_{R}(d,z^{x},x^{x}) = 0$ is reordered as shown in Figure 3 in two sets of equations:

$$s(u,v) = 0$$
$$r(u,v) = 0 \quad (16)$$

Here the subsystems s and r are a partition of the vector of equations $h^{i}_{\wedge}$, whereas the vectors of variables u and v are a partition of the vector

$[(z^{i})^{T},(x^{i})^{T}\,3^{T}$ • As shown in Figure 3, s is the set of non-recycle equations with lower triangular structure, which can be solved sequentially for the vector v given a value of u, and r corresponds to the set of recycle equations. Since v can be treated as an implicit function of u, the system of equations in (16) can be reduced to the form

$$r(u,v(u)) = 0 \qquad (17)$$

The vector u represents the set of decision and torn variables, that is, $u^{T} = C({}^{z}{}_{R}^{i})^{T}.9\ t^{T}]$ and for the optimal sequence its dimensionality is at a minimum.

In order to delete the appropriate equations in step 3 it is sufficient to choose the largest nonsingular subset of equations at the current point. Since for a fixed u, the subsystem s in (16) can be assumed to be non-singular due to its lower triangular structure, the singularity of the system can be analyzed through the jacobian $J_c(r,u)$ of (17) (see Halemane and Grossmann (1981)), which is given by

$$j_{c}(r,u)\ \cdot\Big|£. \quad - \quad \wedge(l^{*})''^{1}\ \Big|1 \qquad (18)$$

This jacobian matrix can be evaluated numerically at the current point by performing perturbations in the vector u. To determine the equations to be deleted the following procedure can be followed. The square submatrix J_c^* of highest rank is obtained by performing a Gaussian elimination on the jacobian matrix (18). The variables in u that correspond to the columns of the submatrix J_c^* will be chosen as torn variables t, whereas the remaining variables in u will correspond to the decision variables $\dot{z}^{\wedge}_{R}$. Those equations r_r in r that are not included in the rows of the submatrix J_*^c, will be deleted and treated as inequality constraints. In this way, the jacobian matrix of the resulting system of equations can be ensured to be of full rank and hence non-singular. Also, note that the jacobian matrix J^c to be analyzed is usually of much smaller size than the jacobian of the system in (16).

The procedures indicated above for finding an initial feasible point and for the variable elimination in the restriction step, complete the algorithm required for the projection-restriction strategy.

Example

To evaluate the performance of the proposed projection-restriction strategy,an example problem has been solved. The flowsheet consists of a reactor and a heat exchanger as shown in Figure 4. The reaction is assumed to be first order exothermic, of the type A—>B. The flowrate through the heat exchanger loop is adjusted to maintain the reactor temperature below T^*_{Imax} as given in Table 2 and to get a minimum of 90% conversion.

This plant is to be designed so as to produce different products in N successive periods within each year. The performance equations of such a system, for any period i, i=l,2,...N, are as follows:

Reactor, material balance:

F X o -Cil)/CL =Vlk0 - P^/RTJ)& (19)

Reactor, heat balance:

(-ΔH) L F^i_o(C - C^i_AL)/C- FoCX - V+ "4 <20)

Heat exchanger, heat balances:

$$Q^i_{HE} = F^i_1 C^i_p (T^i_1 - T^i_2) \tag{21}$$

$$Q^i_{HE} = W^i C_{pw} (T^i_{w2} - T_{w1}) \tag{22}$$

Heat exchanger, design equations:

$$Q_{HE}^{i} = AU(\Delta T)_{m}^{i} \tag{23}$$

$$(\Delta T)_{m}^{i} = \frac{(T_{1}^{i}-T_{w2}^{i}) - (T_{2}^{i}-T_{w1})}{\ln \left\{ \frac{T_{1}^{i}-T_{w2}^{i}}{T_{2}^{i}-T_{w1}} \right\}} \tag{24}$$

The values of the parameters of the problem are given in Table 2. The design problem corresponds to an optimization problem with the equality constraints given by the performance equations above, and the following inequality constraints for each period i, i=1,2,...N.

$$\hat{V} \geq 0 \tag{25}$$

$$A \geq 0 \tag{26}$$

$$V^{i} \geq 0 \tag{27}$$

$$\hat{V} - V^{i} \geq 0 \tag{28}$$

$$W^{i} \geq 0 \tag{29}$$

$$F_{1}^{i} \geq 0 \tag{30}$$

$$0.9 \leq (C_{Ao}^{i} - C_{A1}^{i})/C_{Ao}^{i} \leq 1.0 \tag{31}$$

$$T_{1}^{i} \leq T_{1max}^{i} \tag{32}$$

$$T_{1}^{i} - T_{2}^{i} \geq 0 \tag{33}$$

$$T_{w1} \leq T_{w2}^{i} \leq 356 \tag{34}$$

$$T_{1}^{i} - T_{w2}^{i} \geq \delta \tag{35}$$

$$T_{2}^{i} - T_{w1} \geq \delta \tag{36}$$

The objective function being minimized is the total annual cost ($/yr),

$$C = (2304\ \hat{V}^{0.7} + 2912\ A^{0.6})\ 0.3 + \sum_{i=1}^{N} (2.20x10^{-4}W^{i} + 8.82x10^{-4}F_{1}^{i})t^{i} \tag{37}$$

where t^{i} corresponds to the number of hours of operation for each period i=1,2,...N, in one year. The objective function includes the investment cost of the reactor and heat exchanger, and the operating cost of the cooling water and recycle.

There are 2+9N variables $\hat{V}$, A, C^, T^*_1, T^*_2, T^, F^*_1, W^1, V^1, $(AT)^*_m$, $\overset{i}{Q}$jjg, i=l,2,...N; 6N equations and 2+10N inequality constraints and bounds, for a problem with N different periods. This gives rise to 2-K3N degrees of freedom. Selecting as decision variables the design variables V, A, and the control variables T^i^ T^*_2 T^*_w2, i=l,2,...N, the sequence of calculation for the equations in each period is given in Table 3. The corresponding NLP consists of 2-r3N decision variables, 6N nonlinear inequality constraints and N linear inequality constraints. Note that (25), (26), (32), (34) and (36) are simple bounds on the decision variables, that (33) is a linear inequality and the remaining constraints are nonlinear.

The problem was solved for five cases corresponding to N = 1,2,...5. In each case, the plant is designed to produce N different products that have different feed flowrates, concentrations, reaction rate constants, etc. as shown in Table 2. In all the five cases, when solving the subproblems in the projection step it was found that constraint (31) is active at its lower bound, and constraints (32) and (34) are active at their upper bounds, for all periods. Adding these active constraints to the equations in Table 3 the variables T^i_i, T^*_2, T^i_w9, i=l,2,...N were eliminated by ordering the new system of equations.

A

This gives rise to only two decision variables V and A for the restricted problem, as shown in Table 4. In all cases, the optimum solution was found by solving this restricted problem, thus requiring only a single pass for the projection-restriction strategy.

The starting point given in Table 2, which is infeasible, was used for all five cases. The initial feasible points used in the projection-restriction strategy were obtained by minimizing alternately with respect to d and z^i , i=l,2,...N the sum of squares of deviations of violated constraints. The optimizations were performed using the variable-metric projection method (Sargent and Murtagh, 1973), and the solutions were obtained with a tolerance of 10 for the norm of the gradient of the objective function projected in the constrained space.

The optimal sizes of the reactor and the heat exchanger are presented in Table 5 for the five cases. The formulation of the problem itself ensures that these optimal designs are flexible, as they meet the specifications for the various products involved at a minimum annual cost.

The computing requirements for finding the initial feasible points are shown in Table 6. Here, it was found that optimizing alternately for d and z^i is more efficient than considering all these variables simultaneously, particularly when the number of periods increases. However, the more significant gains in computational requirements are achieved when the projection-restriction strategy is applied, once the problem becomes feasible. Table 7 gives the CPU-time requirements for the projection and restriction steps. Table 8 and Figure 5 give a comparison of the computational requirements in solving the design problem with and without the decomposition. A striking feature in the performance of the proposed decomposition strategy is that the CPU-time increases only linearly with the size of the design problem. It is interesting to note that the design problem for the five-period case was solved by using the proposed decomposition strategy in only 31.4 sec which is about the same time required for solving the one-period problem without using any decomposition.

Discussion

The results of the above example show that the performance of the proposed decomposition strategy for the design of flexible chemical plants is very encouraging. An important trend in the results is that the computational time required is approximately linear with the size of the design problem (number of periods). This suggests that reduction in the computational effort with the proposed method in larger problems should be even more dramatic, when compared with the simultaneous optimization of all the decision variables. This is to be expected, since experience with different nonlinear programming algorithms indicate that they are much more likely to be successful in converging to the optimal solution when the number of decision variaLles is relatively small.

In the Appendix a simple analysis is presented which explains the linear relation for the computer time obtained in the example. It is also shown in the Appendix that when the number of periods is large the projection-restiction strategy can be expected to perform better than the simultaneous solution even if the percentage of control variables that are eliminated in the restriction step is not very large.

Acknowledgement

The authors would like to acknowledge the financial support provided by the National Science Foundation under Grant CPE 79-26398

Nomenclature for the example problem

A : Heat transfer area of the heat exchanger, m^2

C^i_{A0} : Concentration of reactant in the feed, kg mole/m^3

C^i_{A1} : Concentration of reactant in the product, kg mole/m^3

C^i_P : Heat capacity of the reaction mixture, kJ/kg mole K

C_{pw} : Heat capacity of cooling water, kJ/kg K

(E^x/R) : Ratio of activation energy to gas constant, K

F^i_o : Feed flowrate, kg mole/hr

F^i_2 : Flowrate of the recycle, kg mole/hr

k_o : Arrhenius rate constant of reaction, hr^{-1}

Q^i : Heat exchanger load, kJ/hr

T_o : Temperature of feed, K

T^i_1 : Reactor temperature, K

T^i_2 ; Recycle temperature, K

T_{w1} : Inlet temperature of cooling water, K

T^i_{w2} : Outlet temperature of cooling water, K

t^i : Length (time) of i^{th} period of operation, hr

U : Overall heat transfer coefficient, kj/m^2 hr K

V^i : Reaction volume, m^3

V^* : Volume of reactor (design capacity), m

W^i : Flowrate of cooling water, kg/hr

δ : Minimum approach temperature, K

$(\Delta H_{rxn})^*$: Heat of reaction, kj/kg mole

$(\Delta T)^i_m$: Mean temperature difference, K

Superscript i refers to the period of operation, $i=1,2,...N$

REFERENCES

Book, N.L. and F. Ramirez, "The Selection of Design Variables in Systems of Algebraic Equations," AIChE J. 2£, 55 (1976).

Brosilow, C# and L. Lasdon, "A Two-Level Optimization Technique for Recycle Processes," AIChE-I. Chem. E. Symp. Ser. No. 4, 67 (1965).

Christensen, J.H., "The Structuring of Process Optimization," AIChE J. 1JS, 1771 (1970).

Christensen, J.H. and D.F. Rudd, "Structuring Design Computations," AIChE J. 15, 94 (1969).

Edie, F.C. and A.W. Westerberg, "Computer-Aided Design: Part 3. Decision Variable Selection to Avoid Hidden Singularities in Resulting Recycle Calculations," Chem. Eng. J. 2, 114 (1971).

Geoffrion, A.M., "Elements of Large-Scale Mathematical Programming," Manage. Sci. Ij6, 652 (1970).

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

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. 1£, 343 (1979).

Halemane, K.P. and I.E. Grossmann, "Optimal Process Design~Under Uncertainty", presented at the 74th Annual Meeting of the AIChE, New Orleans, Nov. 1981.

Halemane, K.P. and I.E. Grossmann, "Selection of Decision and Torn Variables in Process Design Computations", presented at the Summer Computer Simulation Conference, Washington, D.C#, July 1981.

Hernandez, R. and R.W.H. Sargent, "A New Algorithm for Process Flowsheeting," paper presented in the 12th Symposium on Computer Applications in Chemical Engineering, Montreux, Switzerland (1979).

Lasdon, L.S., "Optimization Theory for Large Systems," Collier-MacMillan Ltd., London (1970).

Lee, K.F., J. Christensen and D.F. Rudd, "Design Variable Selection to Simplify Process Calculations," AIChE J. 12, 1104 (1966).

Leigh, M., "A Computer Flowsheeting Programme Incorporating Algebraic Analysis of the Problem Structure," Ph.D. Thesis, University of London, U.K. (1973).

Ritter, K., "A Decomposition Method for Structured Nonlinear Programming Problems", in D. Himmelblau (Ed.) Decomposition of Large-Scale Problems, North-HoHand, Amsterdam (1973).

Rosen, J.B. and J.C. Ornea, "Solution of Nonlinear Programming Problems by Partitioning", Manage. Sci. 10, 160 (1963).

Sargent, R.W.H. and B.A. Murtagh, "Projection Methods for Nonlinear Programming", Math. Prog. 4, 245 (1973).

Stadtherr, M.A., W.A. Gifford and L.E. Scriven, "Efficient Solution of Sparse Sets of Design Equations", Chem. Eng. Sci. 2£, 1025 (1974).

Stephanopoulos, G. and A.W. Westerberg, "The Use of Hestenes[1] Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization", J. Opt. Theory Appl. L5, 285 (1975a).

Stephanopoulos, G. and A.W. Westerberg, "Synthesis of Optimal Process Flowsheets by an Infeasible Decomposition Technique in the Presence of Functional Non-convexities", Can. J. of Chem. Eng. 53^, 551 (1975b).

Umeda, T., A. Shindo and E. Tazaki, "Optimal Design of Chemical Process by Feasible Decomposition Method", Ind. Eng. Chem. Process Des. Dev. H^, 1 (1972).

Appendix

Based on some simple assumptions, a relationship can be derived between the CPU-time and the size of the design problem in terms of number of periods.

Let n_d be the number of design variables, n_z the number of control variables in each period and N the number of periods considered in the design problem. Assume that the CPU-time t_p for the projection step is given by

$$t_P = A_p(n_z)^P. \qquad \text{(A1)}$$

If $^{a}{>}0 \leq a \leq 1$, is the average fraction of the control variables remaining in the restriction problem, then the CPU-time t_K for the restriction step can be expressed as

$$t_R = A_r(n_d + N a n_z)^r \qquad \text{(A2)}$$

Also, let the CPU-time t_Q for solving the problem without decomposition be

$$tq = V^n d + ^N V^q \qquad \text{(A3)}$$

If K is the number of iterations (passes) through the projection and restriction steps that is required for convergence, then the total CPU-time needed to solve the design problem using the decomposition strategy is

$$t_m = KNA_p(n_z)^P + KA_r(n_d + Nan_z)^r \qquad \text{(A4)}$$

In general the exact values of p, q, r and A_p, A_q, A^, K depend on the particular problem at hand. However, for a given problem the values of A_p, A_q and A_r can be expected to be of the same order of magnitude, and for a gradient based non-linear programming algorithm one can expect to have the values of p, q and r to lie between 2 and 3. Also, the value of K is likely to be small.

If all the control variables are eliminated in the restriction step, $\alpha = 0$ and from (A4) it is then clear that the CPU-time t_{PR} is linear in N the number of periods, as given in (A5):

$$\alpha = 0 : \qquad t_{PR} = KNA_p(n_z)^p + KA_r(n_d)^r \tag{A5}$$

This is in fact the trend that is observed in the results of the example problem, as can be seen in Figure 5.

The relative savings in computation time in using the decomposition strategy can be determined from (A3) and (A4),

$$t_{PR}/t_Q = \frac{KNA_p(n_z)^p + KA_r(n_d + N\,\alpha\, n_z)^r}{A_q(n_d + Nn_z)^q} \tag{A6}$$

Since A_p, A_q, A_r are of some order of magnitude, $0 \leq \alpha < 1$ and $p, q, r > 1$, it is clear from (A6) that the relative advantage in using the decomposition strategy is enhanced by larger values of N and smaller values of α. In fact, for a given value of N there is a threshold value α^t for α, below which savings in CPU-time can be ensured by using the decomposition strategy. This threshold value determines a useful range $0 \leq \alpha \leq \alpha^t$, which can be determined from (A6) with $t_{PR} \leq t_Q$, thus obtaining α^t as

$$\alpha^t = \left[\left\{(A_q/A_rK)\ (n_d + Nn_z)^q - N(A_p/A_r)\ (n_z)^p\right\}^{1/r} - n_d\right]/Nn_z \tag{A7}$$

Assuming $A_p = A_q = A_r$ and $p = r = q$, (A7) can be simplified as

$$\alpha^t = \left[\frac{1}{K}\left\{n_d/Nn_z + 1\right\}^q - \frac{1}{N^{q-1}}\right]^{1/q} - \frac{n_d}{Nn_z} = 1 - \beta^t \tag{A8}$$

where β^t indicates the minimum fraction of control variables to be eliminated in the restricted problem. Figure A1 shows some plots of β^t versus N as given by the expression in (A8) above for the case when $n_d = n_z$. There are two sets of three plots, for $q = 2,3$ and $K = 1,2,3$. As can be seen from this figure,

$/\}^{t}$ is smaller for larger values of q and for smaller values of K. Also, for a given q and K, $/\$^{t}$ decreases rapidly with N even for relatively small values of N, and approaches zero asymptotically for large N.

As an example, take the case when q » 2 and when a single pass (K=l) in the decomposition strategy can lead to the optimal solution. If the design problem involves five periods, only 8% of the control variables must be eliminated to achieve a relative gain in CPU-time with the decomposition strategy. These percentages increase to 18% for K=2 and 28% for K•3. For a ten period problem the percentages reduce respectively to 4%, 10% and 15% for K ^{s} 1,2,3. Thus, in general one can expect to obtain significant gains in computation time when solving the multi-period problem with the projection-restriction strategy, even when the number of active constraints in not very large.

Table 1 - Active constraints in the problems solved by Grossmann and Sargent (1978, 1979)

Problem	Number of decision variables	Number of inequality constraints in the problem	Number of active constraints at the solution
Pipeline	8	20	5
Multiproduct Batch plant			
(a) problem la	10	23	11
(b) problem 2	14	39	14
Reactor-separator system	7	24	4
Heat exchanger net-work	15	65	15

Table 2 - Data for the example problem

T_o = 333°K, T_{w1} = 300°K, S = 11.1°K, U = 1635.34 kj/m^2 hr K

Period	1	2	3	4	5	
(E^R) -	555.6	583.3	611.1	527.8	500.	°K
$-(\Delta H)^i_{rx_i}$ =	23260.	25581.	27907.	20930.	18604.	kJ/kg mole
k^1_o =	0.6242	0.6867	0.7491	0.5619	0.4994	m^3/kgmole hr
C^i_p =	167.4	188.4	209.3	146.5	125.6	kJ/kg mole
I =	32.04	40.05	48.06	24.03	32.04	kg mole/m^3
F^i_o =	45.36	40.82	36.29	49.90	54.43	kg mole/hr
T^*_{Imax} =	389.	383.	378.	394.	400.	°K
t^i =	(8000/N)					hr

Starting point: $\overset{A}{V}$ = 14.1584 m^3

A = 11.1484 m^2

T^i_1 = 367 °K

T^i_2 = 328 °K   i = 1 to N.

T^i_{W2} = 333 °K

Table 3 - Equation Ordering In the Projection Step

VARIABLES

WQUAHION∞	$\hat{V}$	A	A	$*^{l}_{2}$	T^{1}_{W2}	$(\Delta T)^{i}_{m}$	4	C^{i}_{A1}	V^{1}	FJ	W^{1}
24			X	X	X	X					
23		X				X	X				
20			X				X	X			
19			X					X	X		
21			X	X			X			X	
22					X		X				X

Table 4 - Equation Ordering in the Restriction Step

VARIABLES

		$\hat{V}$	A	C^i_{A1}	<	T^i_{w2}	V^i	4	$(AT)^i_m$	T^i_2	F^1_1	W^i
	31			X								
	32				X							
55	34					X						
O	19			X	X		X					
				X	X			X				
<!	23		X					X	X			
					X	X			X	X		
w	21							X		X	X	
	22					X		X				X

Table 5 - Solution of the example problem

[1]Number of	"K	1	Optimum
periods N	V (m˘)	A (m¯)	Annual Cost ($/yr)
1	5.318	7.562	0.980×10^4
2	5.318	8.417	1.010×10^4
3	5.318	9.513	1.042×10^4
4	7.915	9.262	1.096×10^4
5	7.915	9.095	1.080×10^4

[1] Number N indicates periods 1,2,...N taken together.

Table 6 - Computational results for finding an initial feasible point - CPU Time[1]

[2]Number of	Optimize all variables	Optimize alternately
periods N	simultaneously	d and z
1	0.228	0.204
2	0.371	0.432
3	3.745	0.670
4	1.763	0.878-
5	2.954	1.083

[1] DEC-System 20 sec.

[2] Number N indicates periods 1,2,....N taken together.

Table 7 - Computational results for the Projection-Restriction algorithm - CPU Time[1]

[2]Number of periods N	Projection	Restriction	Total CPU Time
1	4.531	1.304	5.835
2	12.129	2.221	14.350
3	15.918	1.416	17.334
4	21.998	2.181	24.179
5	27.311	3.005	30.316

[1] DEC-System 20 sec.

[2] Number N indicates periods 1,2,...N taken together.

Table 8 Computational results for solving the design problem - CPU Time[1]

			Computational Time[1]	
[2]Number periods N	Number of decision variables	Number of inequality constraints	without decomposition	with decomposition
1	5	7	32.2	6.0
2	8	14	176.2	14.8
3	11	21	479.6	18.0
4	14	28	-	25.0
5	17	35	-	31.4

[1] DEC System - 20 sec.

[2] Number N indicates periods 1,2,...N taken together.

Figure 1. Block-diagonal structure in the constraints of problem 3.

Figure 2. Contours of the objective function in problem (13).

Figure 3. Structure resulting from equation ordering.

Figure 4. Flowsheet of the example problem.

Figure 5. Computational time (DEC-20, sec) for solving the design problem versus the size of the problem.

Figure Al. Minimum fraction $/3^{\dagger}$ of control variables that must be eliminated as given in (A8).

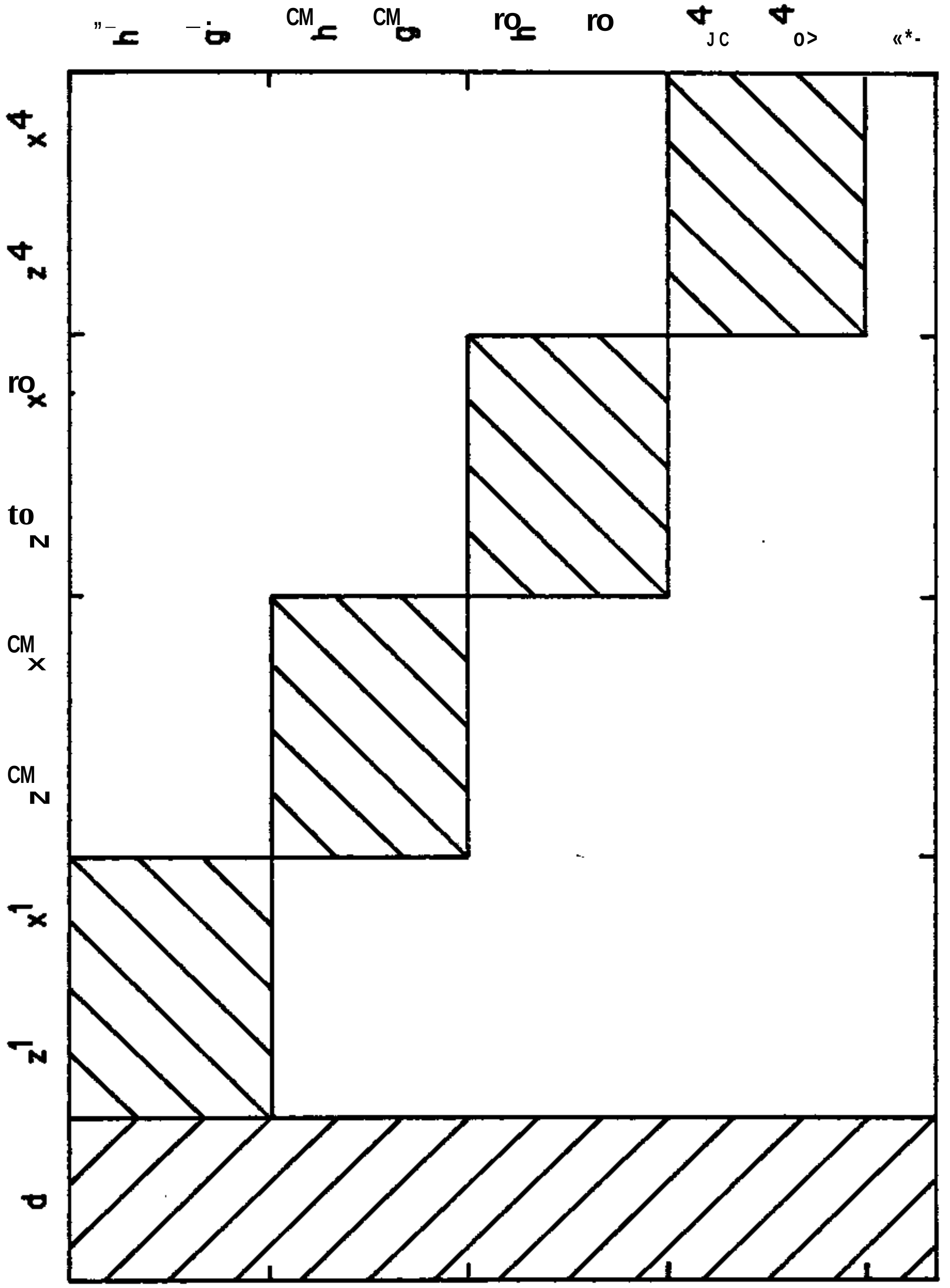


Fig. 1

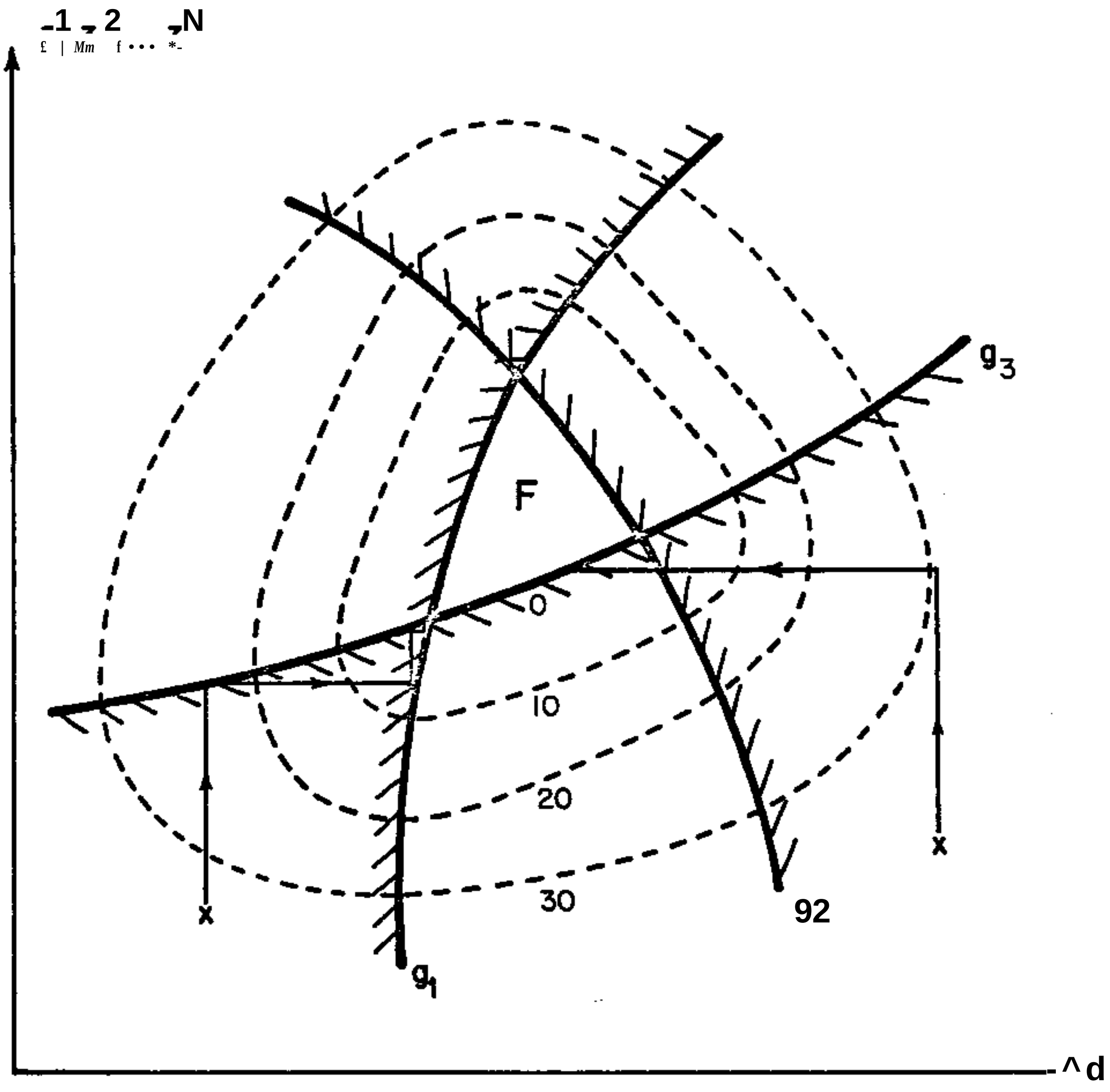


Fig. 2

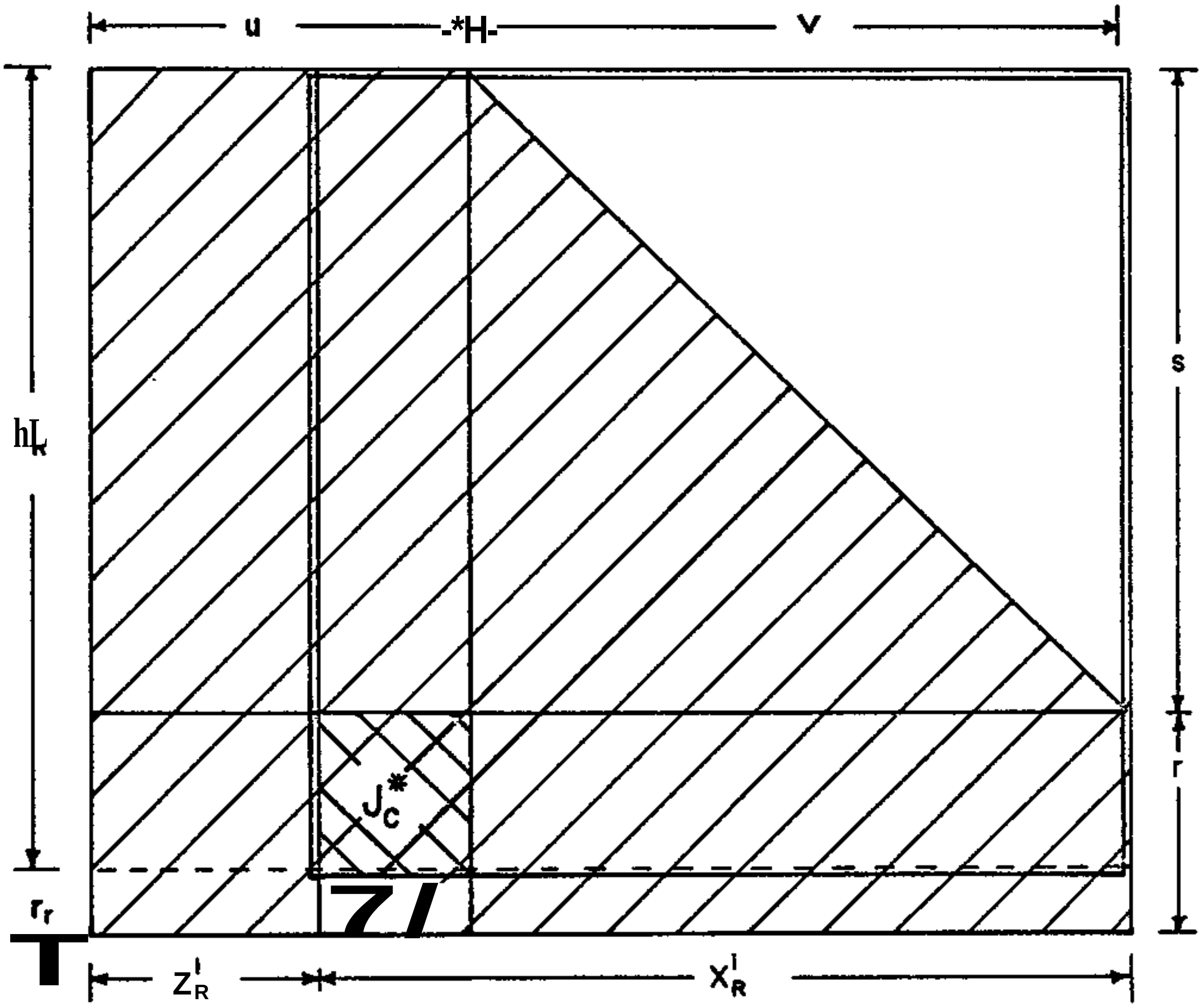


Fig. 3

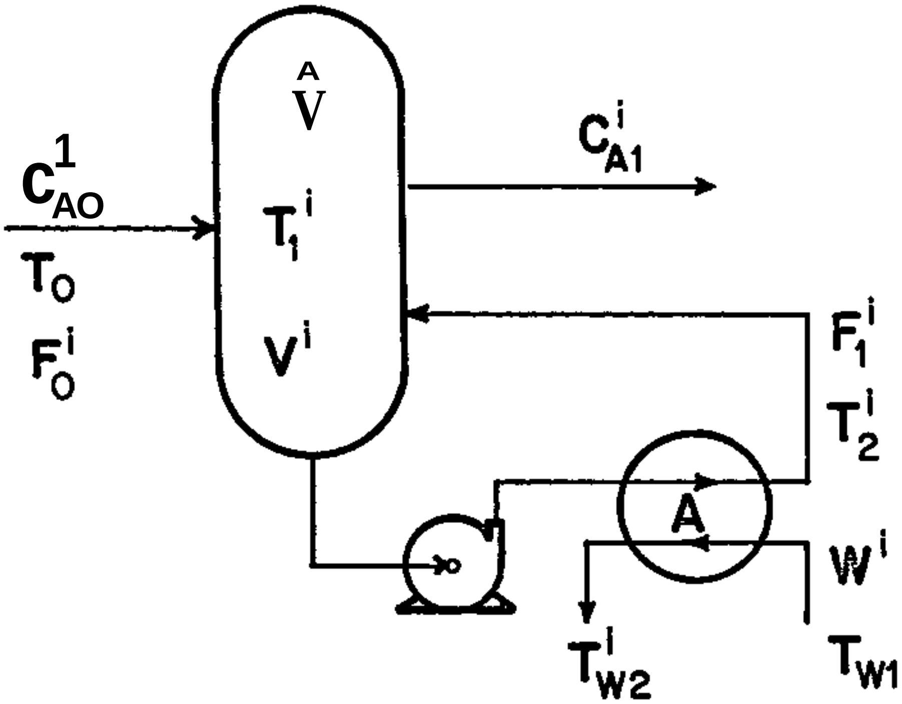
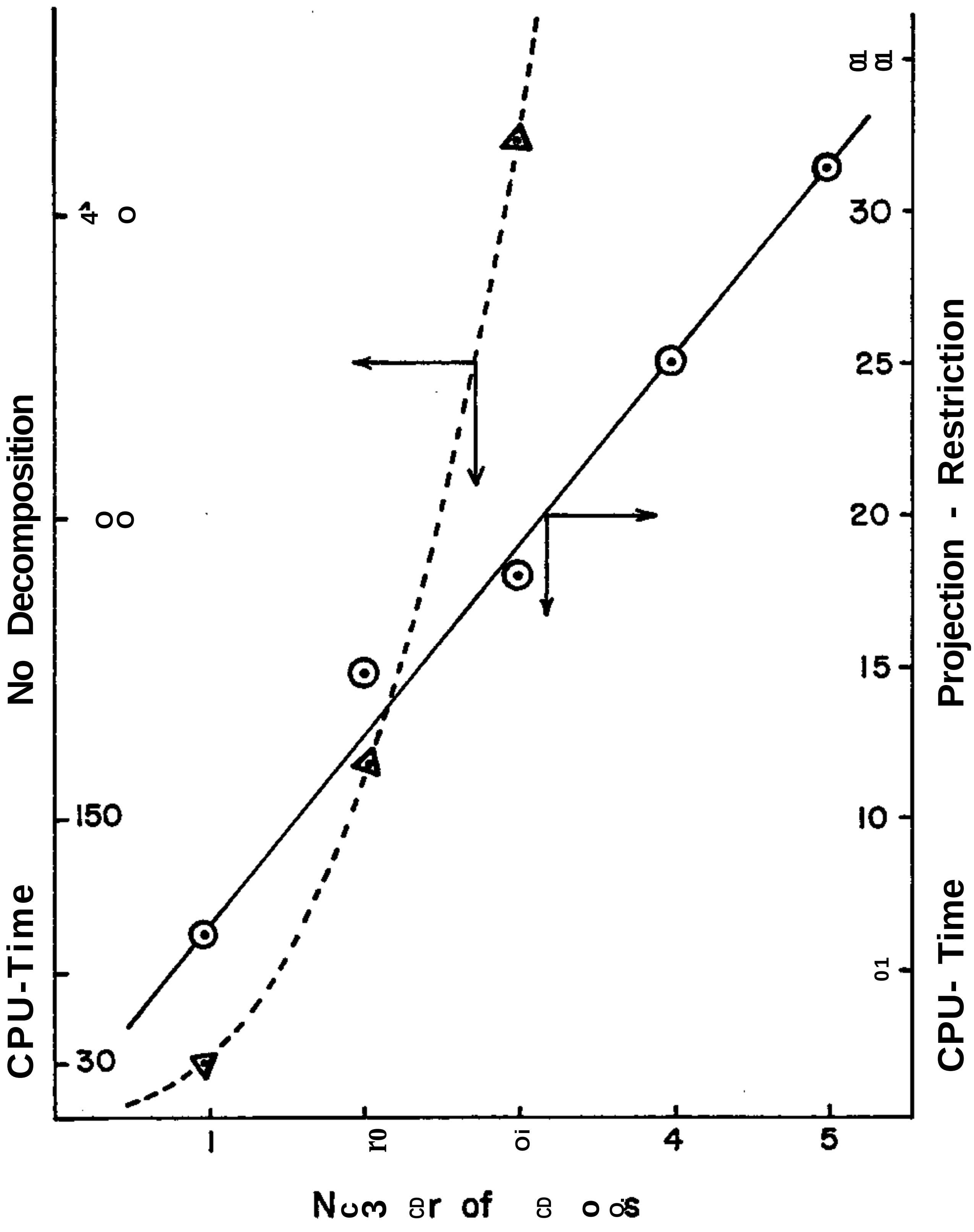


Fig. 4


CPU-Time
No Decomposition
30
150
00
40
CPU- Time
Projection - Restriction
01
10
15
20
25
30
1
r0
oi
4
5
Number of

Fig

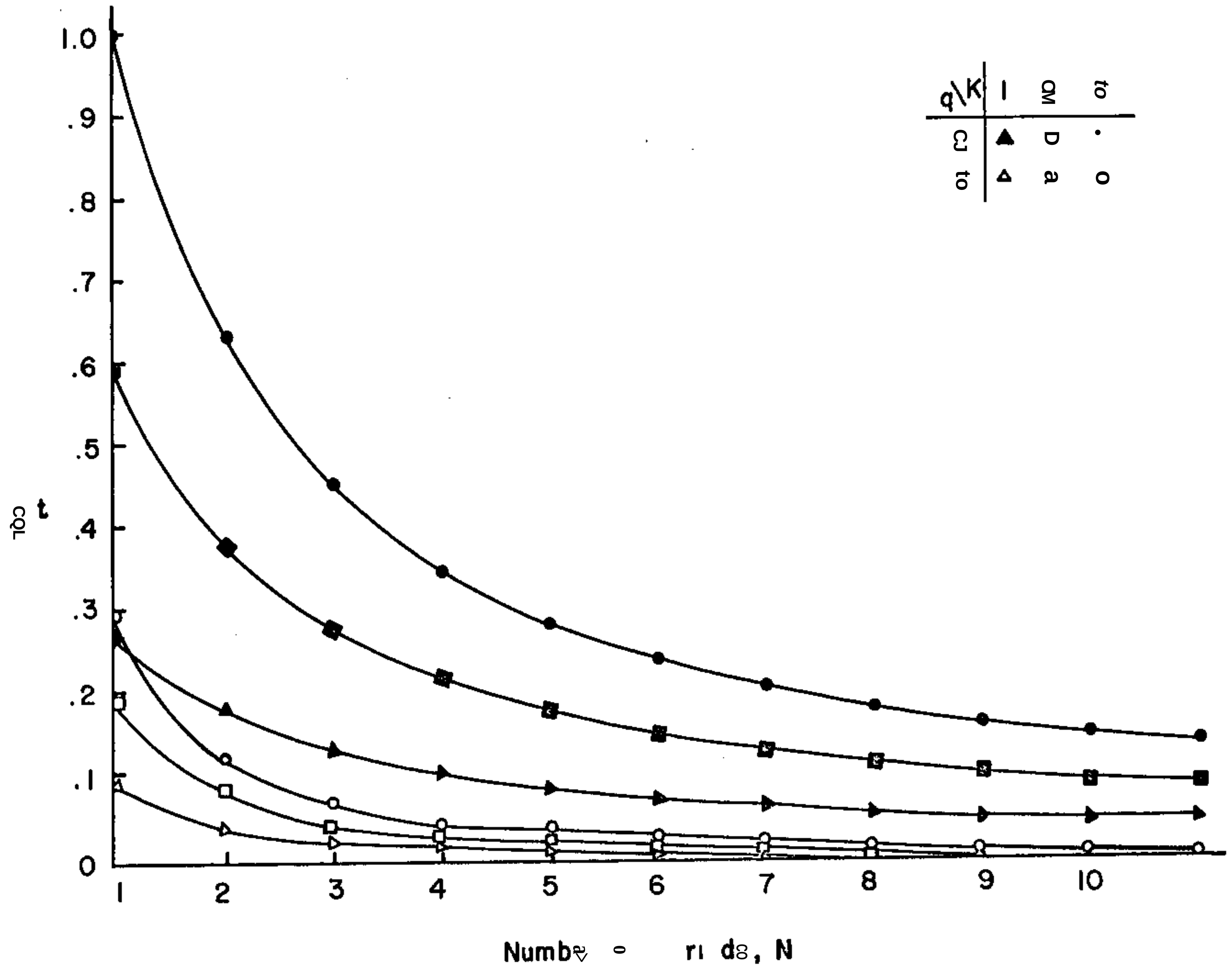

1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
t CQL
1
2
3
4
5
6
7
8
9
10
Numbe o rı dꝏ, N
q\K
I
CM
to
CJ
▲
D
•
to
△
a
o

Fig. Al