

OPTIMAL DESIGN USING DECOMPOSITION METHODS

by

Shapour Azarm & Wei-Chu Li

OPTIMAL DESIGN USING DECOMPOSITION METHODS

by

Shapour Azarm
Assistant Professor

and

Wei-Chu Li
Graduate Student

Department of Mechanical Engineering
and
Systems Research Center
The University of Maryland
College Park, MD 20742

October 1986

ABSTRACT

In this paper, a unified review of eight decomposition methods is presented. A two-level decomposition method is proposed, which is an extension of feasible model coordination methods. The method couples the global monotonicity analysis of the first-level subproblem(s) with an optimization method (single-level method) of the second-level. Three classes of problems are considered where in the first-level they have (1) one subproblem with one local variable, (2) several subproblems with one local variable, and (3) several subproblems with several local variables. Some test results have been presented which shows the substantially improved performance of the proposed approach over a single-level optimization method.

1. Introduction

Recently, optimization methods have found considerable applications to engineering design problems (Siddall, 1982; Reklaitis et al., 1983; Vanderplaats, 1984). Because of their nature, these problems are in general nonlinear and constrained, i.e. nonlinear programming (NLP) problems formulated in the following form :

Minimize $f(x)$

Subject to: (1.1)

$$h_k(x) = 0 \quad k = 1, \dots, K$$

$$g_j(x) \leq 0 \quad j = 1, \dots, J$$

where x is an n -vector of design variables and f, h_k , and g_j are the scalar objective and constraint functions. There exist now optimization algorithms which can solve small to medium-sized problems formulated in the above form quite efficiently (Lasdon and Waren, 1983; Crane et al., 1980). However, when the size of the problem becomes "large", which is the case for many engineering problems, then the solution process using the existing algorithms becomes expensive, if not impossible.

What is a "large-scale" NLP problem ? It is difficult to give an exact definition for a large-scale NLP problem. In nonlinear programming, the size of the problem is determined by the number of variables, the number and complexity of the constraint functions, and the complexity of the objective

function (Lasdon, 1970). It is perhaps safe to say that when the number of variables or constraints exceeds 50, we have encountered a large-scale NLP problem (Gabriel and Ragsdell, 1980).

There is a very large literature on the subject of large-scale mathematical programming methods. In general, these methods may be divided into two classes (Lasdon, 1970):

- (1) Direct methods which specialize an existing method to a particular class of problems to solve them directly. Examples of the direct methods include the Simplex method (Dantzig, 1963), more recently the Karmarkar's method (Karmarkar, 1984) for linear programming, and the projected Lagrangian method (Murtagh and Saunders, 1982) for nonlinear programming.
- (2) Indirect or decomposition methods which decompose the original problem into subproblems whose solutions in a prescribed manner will generate the solution of the original problem. It is this class of methods which is the subject of this paper.

The organization of the remainder of this paper is as follows. In Section 2, we present a unified review of eight decomposition methods for nonlinearly constrained design optimization problems. Our proposed monotonicity-based decomposition method which is an extension of the ones described in Section 2 together with some test results is presented in Section 3. Finally, we present in Section 4, the concluding remarks for this study.

2. Decomposition Methods

The idea of decomposition for solving large-scale nonlinear systems

was first proposed by Kron (1954). There, he indicated that "physical systems with a very large number of variables (say with tens of thousands) may be solved with available digital computers by tearing the system apart into a large number of small subdivisions". However, it was the publication of the Dantzig-Wolfe (1960) decomposition method which initiated the extensive work on large-scale mathematical programming. Dantzig-Wolfe method was developed for the decomposition of linear programming problems whose coefficient matrices have angular structure. In this method, the original program is decomposed into several linear subprograms and a "master (coordinating) program". At each iteration, the subprograms receive a set of parameters (simplex prices) from the master program. The subprograms then send their current solutions to the master program which in turn obtains a new set of prices to be sent again to the subprograms. The iterations continue until an optimal solution is obtained.

The basic steps followed by decomposition methods for NLP problems are very similar to the Dantzig-Wolfe method. These algorithms have a two-level structure and break down a problem having certain structure into several smaller subproblems. In the first level, these smaller subproblems are solved independently. Then, in the second level, the subproblem's solutions are coordinated to obtain the solution to the original problem. Decomposition methods fit very well with mechanical design problems. For example, Figure 1 shows a two level decomposition for a punch-press. A punch-press is a mechanical system composed of several components including flywheel, crankshaft, connecting rod, etc. Using decomposition, each component is optimized at the

first level and then the components' solutions are coordinated at the second level to obtain an optimum for the original problem; i.e., the punch-press.

Since the publication of the Dantzig-Wolfe method more than twenty five years ago, a great number of papers and several books have been published on the subject of decomposition. It is interesting to note that several of the major books on the subject were published during the period of 1970 to 1973 (Mesarovic et al., 1970; Lasdon, 1970; Wismer, 1971; Himmelblau, 1973). In the next section, we present a unified review of eight decomposition methods which are applicable to NLP problems. These are only a few of many important methods found in the literature.

2.1 Lagrangian Feasible Method (Wismer, 1971 and 1978)

Consider the NLP problem formulated in the following form :

Minimize $f(x)$

Subject to: (2.1)

$$g_i(x) \leq 0 \quad i = 1, \dots, n.$$

Note that the constraints have been organized into n (number of variable) functional constraints. We assume that the objective function can be written as the sum of single variable functions, i.e. $f(x)$ is an additively separable function:

$$\text{Minimize } f(x) = \sum_{i=1}^n f_i(x_i)$$

Subject to: (2.2)

$$g_i(x) \leq 0, \quad i = 1, \dots, n.$$

If we now form the Lagrangian for this problem, we have :

$$L(x, u) = \sum_{i=1}^n f_i(x_i) + \sum_{i=1}^n u_i^t g_i(x_1, \dots, x_n) \quad (2.3)$$

or

$$L(x, u) = \sum_{i=1}^n [f_i(x_i) + u_i^t g_i(x_1, \dots, x_n)]. \quad (2.4)$$

We then make the Lagrangian into an additively separable function. To do that, we define a new set of constraints:

$$s_j = x_j, \quad j = 1, \dots, n \quad (2.5)$$

which will be considered in the problem as (interconnection) equality constraints; therefore, after substitution we have:

$$L(x, v, u, s) = \sum_{i=1}^n [f_i(x_i) + u_i^t g_i(x_i; s_1, \dots, s_j, \dots, s_n)_{j \neq i} + v_i^t (x_i - s_i)] \quad (2.6)$$

note that $v_i = 0$, if $g_i(x_1, \dots, x_n)$ is not a function of x_i for all j not equal to i . In equation (2.6), assume that s_i is fixed to decompose that to the following subproblems:

$$L_i(x_i, v_i, u_i; s_i) = f_i(x_i) + u_i^t g_i(x_i; s_1, \dots, s_j, \dots, s_n)_{j \neq i} + v_i^t (x_i - s_i) \quad (2.7)$$

now, we can solve each subproblem independently. Let's write the KKT conditions for the subproblems :

$$\partial L_i / \partial x_i = \partial f_i / \partial x_i + (\partial g_i / \partial x_i)^t u_i + v_i = 0 \quad (2.8)$$

$$\partial L_i / \partial u_i = g_i(x_i; s_1, \dots, s_j, \dots, s_n)_{j \neq i} \leq 0 \quad (2.9)$$

$$\partial L_i / \partial v_i = x_i - s_i = 0 \quad (2.10)$$

$$u_i^t g_i = 0 \quad (2.11)$$

$$u_i \geq 0 \quad (2.12)$$

they can be solved for x_i , u_i , v_i as functions of s_i . This was the first-level part of optimization. The second-level part of optimization involves the selection of optimal values for the parameter s_i . Let's write the KKT conditions (this time for s_i as a variable):

$$\partial L_i / \partial s_i = 0 \quad (2.13)$$

thus

$$\partial L_i / \partial s_i = \sum_{j=1}^n (\partial g_j / \partial s_i)^t u_j - v_i, \quad i=1, \dots, n \quad (2.14)$$

by employing a gradient-type algorithm, we may find s_i to satisfy the equation (2.14). If we rewrite (2.14) as:

$$dL_i = \left[\sum_{j=1}^n (\partial g_j / \partial s_i)^t u_j - v_i \right]^t ds_i, \quad i=1, \dots, n \quad (2.15)$$

and choose ds_i in such a way to insure $dL < 0$, then it is obvious that:

$$ds_i = k \left[\sum_{j=1}^n (\partial g_j / \partial s_i)^t u_j - v_i \right] \quad i=1, \dots, n \quad (2.16)$$

where $k < 0$. Thus the second-level algorithm is:

$$s_i^{p+1} = s_i^p + \left[\sum_{j=1}^n (\partial g_j / \partial s_i)^t u_j - v_i \right]. \quad (2.17)$$

The mathematical structure for this method is demonstrated in Figure 2. Note that the interconnection constraints, equation (2.6), are always satisfied. Therefore, even if the second-level optimization terminates prematurely, the resulting point is feasible. The convergence condition for this method requires the existence of feasible solutions for the subproblems.

2.2 Lagrangian Dual Feasible Method (Wisner 1971 and 1978)

Consider the same problem as given by equation (2.2) together with its Lagrangian given by equation (2.3). However, this time we decompose the Lagrangian by assuming v as the fixed parameter:

$$s_{ij} = x_j \quad i, j = 1, \dots, n \quad i \neq j \quad (2.18)$$

then $v_{ij}^t (x_j - s_{ij}) = 0$, which if it is substituted in equation (2.3) we have:

$$L(x, v, u, s) = \sum_{i=1}^n [f_i(x_i) + u_i^t g_i(x_i, s_{i1}, \dots, s_{in})_{i \neq j} + \sum_{j=1}^n v_{ij}^t (x_j - s_{ij})] \quad (2.19)$$

now the subproblems are:

$$L_i(x_i, u_i, s_j; v_{ij}) = f_i(x_i) + u_i^t g_i(\cdot)_{i \neq j} + \sum_{j=1}^n v_{ji}^t x_i - \sum_{j=1}^n v_{ij}^t s_{ij} . \quad (2.20)$$

Here, in the first-level the KKT conditions are used to solve for x_i , u_i , s_{ij} as a function of v_{ij} , and the second-level of optimization determines v_{ij} . In this method, it can be shown that the second-level problem is in fact the dual of the first-level problem. The mathematical structure for this method is demonstrated in Figure 3. It should be mentioned that the convergence condition for this method also requires the existence of solution to the subproblems.

2.3 Lagrangian Dual-Feasible (Relaxation) Method (Wismer, 1978)

This is a combination of Lagrangian feasible and Lagrangian dual feasible methods which solves for both s_i and v_i in the second level. Consider again the same problem given by equation (2.2) together with its Lagrangian, equation (2.7) and the corresponding KKT conditions, equations (2.8) - (2.12). If we solve KKT conditions for $x_i(v, s)$, and $u_i(v, s)$. Then, by the second-level necessary conditions, we have:

$$s_i = x_i \quad (2.21)$$

$$v_i = \sum_{j=1}^n (\partial g_j / \partial s_i)^t u_j \quad (2.22)$$

we can then solve for x_i and u_i using the first-level subproblems. The mathematical structure of the method is shown in Figure 4.

2.4 Feasible Method (Kirsch, 1975 and 1981)

Consider the NLP problem formulated in the following form:

Minimize $f(z)$

Subject to (2.23)

$$h(z) = 0$$

$$g(z) \leq 0$$

where z is an N -vector of design variables, and f , h , g are the objective function and the vectors of equality and inequality constraint functions respectively. If we set

$$z = (y, x)^t \tag{2.24}$$

where y and x are the "interaction" (global) and the "noninteraction" (local) variables respectively. Then we can convert the problem into a two-level form where the first-level subproblems can be solved separately. To apply this method, the optimization problem should have a special structure:

(1) The objective function should be additively separable:

$$f(y, x) = \sum_{n=1}^N f_n(y, x_n) . \tag{2.25}$$

(2) The constraints should have the following structure:

$$h(y, x) = [h_1(y, x_1), \dots, h_n(y, x_n), \dots, h_N(y, x_N)]^t \tag{2.26}$$

$$g(y, x) = [g_1(y, x_1), \dots, g_n(y, x_n), \dots, g_N(y, x_N)]^t \tag{2.27}$$

where x_n , h_n , and g_n represent variables, equality constraints, and inequality constraints associated with the n th subproblem.

The two-level problem is formulated as follows:

(1) First-Level Problem - fix the interaction variable y by setting $y=y_0$.

Then the problem can be decomposed into N independent first-level problem:

$$\begin{aligned} H_n(x_n) &= \text{Minimize } f_n(y_0, x_n) \\ \text{Subject to: } & n = 1, \dots, N \end{aligned} \quad (2.28)$$

$$h_n(y_0, x_n) = 0$$

$$g_n(y_0, x_n) \leq 0 .$$

(2) Second-Level Problem - find y_0 such that:

$$\text{Minimize } H(y_0) = \sum_{n=1}^N H_n(y_0) . \quad (2.29)$$

The two-level structure of the method is shown in Figure 5.

2.5 Dual Feasible Method (Kirsch, 1975 and 1981)

Consider the same problem discussed in the previous Section:

$$\begin{aligned} \text{Minimize } f(z) &= \sum_{n=1}^N f_n(z_n) \\ \text{Subject to: } & \end{aligned} \quad (2.30)$$

$$h(z) = [h(z_1), \dots, h_n(z_n), \dots, h_N(z_N)]^t = 0$$

$$g(z) = [g_1(z_1), \dots, g_n(z_n), \dots, g_N(z_N)]^t \leq 0 .$$

Here, we decompose the problem in such a way that all links between subproblems are cut, i.e. no longer y_0 is the same between the subproblems. We set $z = (y, x)^t$, where y here is the "interconnection" variable. We then define a new objective function:

$$L(z, v) = \sum_{n=1}^N f_n(z_n) + \sum_{n=1}^{N-1} v_n^t (y_n - y_{n+1}) . \quad (2.31)$$

This is the original objective function with the added term, i.e. the penalty term. Note that at the optimum the "interconnection-balance conditions" should be satisfied, i.e.

$$y_n = y_{n+1} , \quad n=1, \dots, (N-1). \quad (2.32)$$

The two-level formulation for the problem is:

(1) First-Level Problem - fix the value of v , then decompose the original problem into N independent subproblems:

Subproblem 1:

$$\text{Minimize } L_1 = f_1(z_1) + v_1^t y_1 .$$

Subject to:

(2.33)

$$h_1(z_1) = 0$$

$$g_1(z_1) \leq 0$$

Subproblem $n = 2, \dots, (N-1)$:

$$\text{Minimize } L_n = f_n(z_n) - (v_{n-1}^t y_n - v_n^t y_n)$$

Subject to (2.34)

$$h_n(z_n) = 0$$

$$g_n(z_n) \leq 0$$

Subproblem N :

$$\text{Minimize } L_N = f_N(z_N) - v_{N-1}^t y_N$$

Subject to: (2.35)

$$h_N(z_N) = 0$$

$$g_N(z_N) \leq 0 \quad .$$

(2) Second-Level Problem - choose v such that

$$\text{Maximize } H(v) = \sum_{n=1}^N H_n(v) \quad (2.36)$$

and

$$y_n = y_{n+1} \quad (2.37)$$

where

$$H_n(v) = \text{Min } L_n(z_n, v) \quad . \quad (2.38)$$

The two-level structure of this method is shown in Figure 6.

2.6 Dual Decomposition for Separable Problems (Lasdon, 1968 and 1970)

Consider a NLP problem of the following form:

Minimize $f(x)$

Subject to (2.39)

$$g_i(x) \leq 0, \quad i = 1, \dots, m$$

$$x \in S$$

where x is an n -vector and S is an arbitrary subset of E^n . If we form the Lagrangian for this problem:

$$L(x, u) = f(x) + u^t g(x) \quad (2.40)$$

and solve the following problem:

Minimize $L(x, u)$

Subject to: (2.41)

$$x \in S, \quad u \geq 0$$

then, it can be shown that if $x(u)$ solves the above problem, then $x(u)$ solves the modified version of the original (primal) problem:

Minimize $f(x)$

Subject to: (2.42)

$$g_i(x) \leq g_i(x(u)), \quad i=1, \dots, m$$

$$x \in S.$$

The formulation given by equation (2.41) is an unconstrained one (except for $x \in S$ and $u \geq 0$). Therefore, if the primal problem is separable:

$$x = (x_1, \dots, x_p)^t \quad p \leq n \quad (2.43)$$

$$f(x) = \sum_{i=1}^p f_i(x_i) \quad (2.44)$$

$$g_i(x) = \sum_{k=1}^p g_{ik}(x_k) \quad (2.45)$$

then the Lagrangion is separable and can be decomposed into the following subproblems:

$$\text{Minimize } L_k = f_k(x_k) + \sum_{i=1}^m u_i g_{ik}(x_k), \quad k=1, \dots, p \quad (2.46)$$

$$x_k \in S_k.$$

If we now set:

$$h(u) = \text{Min.}_{x \in S} \sum_{k=1}^p L_k \quad (2.47)$$

then the function $h(u)$ is called the dual function which is defined for the following domain:

$$D = \{u : u \geq 0, \text{ min. } L(x, u) \text{ exists}\} . \quad (2.48)$$

We can now define a primal-dual pair of problems:

Primal:

$$\begin{aligned} \text{min. } f(x) &= \sum_{k=1}^p f_k(x_k) \\ \text{s.t.} \\ g_i(x) &= \sum_{k=1}^p g_{ik}(x_k) \leq 0 \end{aligned}$$

$$x \in S .$$

Dual:

$$\begin{aligned} \text{max } h(u) \\ \text{s.t.} \\ u \in D \end{aligned} \quad (2.49)$$

It can be shown that at least for Convex programming problems these two problems are equivalent. The two-level structure for this method is shown in Figure 7. Lasdon (1970) suggested the following steepest ascent algorithm:

Maximize $h(u)$:

$$u_{i+1} = u_i + \alpha_i S_i \quad (2.50)$$

where

$$S_i^k = \begin{cases} \partial h / \partial u^k |_{u_i} & \text{if } u_i^k \geq 0 \\ 0 & \text{if } u_i^k = 0 \end{cases}, \quad k=1, \dots, m \quad (2.51)$$

$$= \max \{0, \partial h / \partial u^k |_{u_i}\} \quad \text{if } u_i^k = 0.$$

The step size α_i is selected such that:

$$h(u_{i+1}) > h(u_i) \quad (2.52)$$

2.7 Johnson's (1984a) Two-Stage Decomposition Method

Again, consider the NLP problem formulated in the general form of equation (2.23). Johnson's method decomposes the problem into a Modular Component (MC) and an integrating System Component (SC). This is done by setting $z = (u, v)^t$, where u and v are vectors, such that the original problem is transformed into the following form:

Minimize $f(u, v)$

Subject to: (2.53)

$$h'(u,v) = 0$$

$$h''(v) = 0$$

$$g'(u,v) \leq 0$$

$$g''(v) \leq 0$$

where g' , g'' , h' , h'' are vectors. The problem is then decomposed into the following MC and SC subproblems:

MC Subproblem:

Minimize $f(u,v)$

Subject to: (2.54)

$$h'(u,v) = 0$$

$$g'(u,v) \leq 0$$

SC Subproblem:

Minimize $f(u,v)$

Subject to: (2.55)

$$h''(v) = 0$$

$$g''(v) \leq 0$$

An "exact" solution for the MC subproblem is then sought such that:

$$u_0 = f_0(v), \quad v \text{ is fixed.} \quad (2.56)$$

This solution is inserted into the SC subproblem to solve for v^* . Then from $u_0 = f_0(v)$, we can find u^* . The two-stage structure of the method is shown in Figure 8.

2.8 Johnson's (1984b) Multistage Decomposition Method

This is an extension of the Johnson's two-stage decomposition method described in the previous section. Consider the NLP problem of the previous Section and decompose it into m modular components (MC) and an integrating system component (SC). As in the two-stage method, the independent variables are resolved into u and v components (u and v are vectors). However, the elements of u are further resolved into sets of u_m (u_m is a vector) associated with the m th MC. Again, each MC must be exactly solvable, $u_{0m} = f_{0m}(v)$, furthermore MC subproblems are solved sequentially. On exit from the MC stage, we have u as a function of v , this solution is then inserted in the SC stage to solve for v^* and subsequently from the MC stages we obtain u^* . The structure of this method is shown in Figure 9.

2.9 Discussion on the Reviewed Methods

The decomposition methods reviewed in this section have a two-level hierarchical structure (Mesarovic et al., 1970), except for Johnson's (1984a, 1984b) method. They have an upper unit, called the second-level which controls or coordinates the units on the level below, called the first-level. Although there may be various ways of transforming a given nonlinear programming problem into a two-level form, they are all essentially combinations of two different approaches which are called the model coordination method and the goal coordination method (Wisner 1971). In the model coordination method, described in sections 2.1 and 2.4, the decomposition is made possible by adding a constraint to the mathematical model of the problem in

the form of fixing the interaction variables. For example, in section 2.1 a new set of variables are introduced into the problem, i.e. interconnection constraints, and they are fixed in such a way to make Lagrangian additively separable. Also, in section 2.4, the variable y is selected as the interaction variable to coordinate the activities of the subproblems. The model coordination method is also known as the feasible method due to feasibility of intermediate values of design variables. This method is particularly attractive from engineering design point of view, since the iteration process may be terminated whenever it is desired, with a feasible even though nonoptimal design. One major drawback of this method is possible lack of a feasible solution at the first-level for a given value of the interaction variable y of the second-level.

In the goal coordination method, described in sections 2.2, 2.5, and 2.6, the decomposition is made possible by modification of the objective (goal) of the subproblems while cutting the design variables links between subproblems. The goal coordination method is also known as the dual method since the second-level problem is the dual of the first-level subproblems. As might be expected, this method will not work for all classes of NLP problems unless the existence of a saddle point for the Lagrangian of the problem is guaranteed. The saddle point exists, loosely speaking, only for convex programming problems. However, useful results may be obtained for other classes of problems (Lasdon, 1970; Wismer, 1971). Another shortcoming of this method is the infeasibility of the intermediate values of design variables, thus the iteration process must proceed until the optimum is reached.

The two or multistage decomposition methods described in Sections 2.7 and 2.8 are applicable under the following requirements (Johnson and Benson 1984a and 1984b):

- (a) Constraint Inclusion Requirement, which requires that the solution to MC subproblem(s) be compatible with the SC subproblem.
- (b) Uniqueness Requirement, which requires uniqueness obtained from the MC subproblem(s).

3. A Monotonicity - Based Decomposition Method (MBDM)

In this section, we present a proposed two-level decomposition method. It is an extension of the model coordination method reviewed in Section 2.4 coupled with monotonicity analysis. In this method, the global (instead of local) monotonicity analysis (Wilde, 1975; Papalambros and Wilde 1979 and 1980) has been utilized in the first-level subproblem(s) to identify the active constraints. This information is then sent to the second-level problem which in turn finds a new point for an improved objective function. The new point is then sent back to the first-level subproblem(s) and the iteration process continues until an optimal solution is obtained.

In the next three sections we describe three class of problems, namely, one subproblem with one local variable to the more general case of several subproblems with several local variables, where we have applied the method. In all three classes, the first-level of subproblems are solved by global monotonicity analysis. The second-level problem may then be solved by any conventional nonlinearly constrained optimization method (a single-level method).

3.1 One Subproblem with One Local Variable

We consider a problem which may be decomposed into two levels with one subproblem which has only one local variable:

$$\text{Minimize } f(y;x) = f_0(y) + f_1(y; x)$$

Subject to: (3.1)

$$g_1(y) \leq 0 \qquad l = 1, \dots, L$$

$$g_j(y; x) \leq 0 \qquad j = 1, \dots, J$$

where y is the interaction (global) vector of design variables, and x is the local design variable in the subproblem. Thus, the subproblem is written in the following form, where y is fixed:

$$\text{Minimize } f_1(y; x)$$

Subject to: (3.2)

$$g_1(y; x) \leq 0$$

$$g_2(y; x) \leq 0$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$g_J(y; x) \leq 0 .$$

Based on the first rule of monotonicity, if the objective function of the subproblem is monotonic, e.g. increasing with respect to (w.r.t.) variable x ,

then at least one of the constraints in the subproblem must be active.

Suppose that the first J' constraints ($J' < J$) in the subproblem have opposite monotonicity, e.g. decreasing w.r.t. the variable x , and the rest of the constraints, that is $(J - J')$ constraints, have the same monotonicity w.r.t. the variable x . Also, suppose that the subproblem can be rewritten in the following form:

$$\begin{aligned}
 &\text{Minimize } f_1(y; x) \\
 &\text{Subject to:} \\
 &\quad x \geq g'_1(y) \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad x \geq g'_{J'}(y) \\
 &\quad \cdot \\
 &\quad x \leq g'_{(J' + 1)}(y) \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad x \leq g'_J(y)
 \end{aligned} \tag{3.3}$$

then the second-level problem can be written in the following form:

$$\begin{aligned}
 &\text{Minimize } f(y; x) = f_0(y) + f_1(y; x^*) \\
 &\text{Subject to:} \\
 &\quad g_l(y) \leq 0 \qquad l = 1, \dots, L
 \end{aligned} \tag{3.4}$$

where

$$x^* = \max \{g'_j : 1 \leq j \leq J'\},$$

the last equation may violate the rest of the constraints in the subproblem, in this case, the violated constraint should be transferred from the first to the second level problem. Note that similar argument may be made for the case where the objective function of the subproblem is monotonic decreasing

w.r.t. variable x .

Example 1:

This example is selected from Azarm and Papalambros (1984) where c and d are the design variables and K is the parameter:

$$\text{Minimize } f(d; c) = K_{01} c^{0.86} d^{-1.86}$$

Subject to: (3.5)

$$g_1: K_1 d^2 c^{-1} \leq 1$$

$$g_{25}: K_{25} c^{-1} \leq 1$$

$$g_3: K_3 c^3 d^{-1} \leq 1$$

$$g_4: K_4 c \leq 1$$

$$g_6: K_6 d^2 c^{-3} + L_6 d \leq 1$$

$$g_7: K_7 cd + L_7 d \leq 1$$

$$g_8: c^{-1} + K_8 c^{-1} d^{-1} \leq 1$$

$$g_9: K_9 d^{-1} \leq 1$$

$$g_{10}: K_{10} d \leq 1$$

$$g_{11}: K_{11} c_3 d^{-2} \leq 1$$

Here, we select d as the fixed variable and c as the local variable.

Thus the first-level problem is:

$$\text{Minimize } f(c) = K_{01} c^{0.86} d^{-1.86}$$

Subject to: (3.6)

$$\begin{aligned}
g_1: \quad c &\geq (K_1 d_2 = g'_1) \\
g_{25}: \quad c &\geq (K_{25} = g'_{25}) \\
g_3: \quad c &\leq (d/K_3 = g'_3) \\
g_4: \quad c &\leq (1/K_4 = g'_4) \\
g_6: \quad c &\geq ((L_6 d - 1)/(K_6 d^2))^{1/3} = g'_6) \\
g_7: \quad c &\leq (1 - L_7 d / (K_7 d) = g'_7) \\
g_8: \quad c &\geq (1 + K_8 / d = g'_8) \\
g_{11}: \quad c &\leq ((d^2 / K_{11})^{1/3} = g'_{11})
\end{aligned}$$

and the second level problem is:

$$\begin{aligned}
\text{Minimize } f(d) &= K_{01} c^* \frac{0.86}{d} - 1.86 \\
\text{Subject to:} & \\
K_9 &\leq d \leq 1/K_{10}
\end{aligned} \tag{3.7}$$

where

$$c^* = \max \{g'_1, g'_{25}, g'_6, g'_8\} .$$

Figure 10 shows the two-level structure for this example. Note that if for a given value of d^* in the first level, the rest of constraints are violated i.e. g_3, g_4, g_7, g_{11} , then the violated constraints should be transferred from the first to the second level problem.

3.2 Several Subproblems with One Local Variable

We consider here the problem which may be decomposed into two levels with several subproblems, each having one local variable:

$$\text{Minimize } f(y; x) = f_0(y) + \sum_{i=1}^I f_i(y; x_i)$$

Subject to: (3.8)

$$\begin{aligned} g_l(y) &\leq 0 & l &= 1, \dots, L \\ g_{i,j}(y; x_i) &\leq 0 & i &= 1, \dots, I \\ & & j &= 1, \dots, J \end{aligned}$$

index j corresponds to the number of constraints in subproblem i . The formulation for subproblem i with x_i as the only local variable is:

Minimize $f_i(y; x_i)$

Subject to: (3.9)

$$g_{i,j}(y; x_i) \leq 0 \quad j = 1, \dots, J.$$

Again suppose that the objective function is monotonic, e.g. increasing, w.r.t. variable x_i . Suppose that the first J' constraints ($J' < J$) have opposite monotonicity w.r.t. x_i , e.g. decreasing, which means that based on the first rule they are the candidate active constraints. Also suppose that the subproblem i ($1 \leq i \leq I$) can be rewritten in the following form:

Minimize $f_i(y; x_i)$

Subject to: (3.10)

$$\begin{aligned} x_i &\geq g'_{i,1}(y) \\ &\vdots \\ x_i &\geq g'_{i,J'}(y) \\ x_i &\leq g'_{i,(J'+1)}(y) \\ &\vdots \\ x_i &\leq g'_{i,J}(y) \end{aligned}$$

then the second level problem is written in the following form:

$$\begin{aligned} \text{Minimize } f(y;x) &= f_0(y) + \sum_{i=1}^n f_i(y;x_i^*) \\ \text{Subject to:} \end{aligned} \quad (3.11)$$

$$g_l(y) \leq 0 \quad l = 1, \dots, L$$

where

$$x_i^* = \max \{g'_{i,j} : 1 \leq j \leq J'\}.$$

As before, those constraints, $g_{i,j}(y; x_i)$, where $j > J'$, which are violated by the x_i^* should be transferred from the first level subproblem(s) to the second-level problem. Again, note that similar argument may be made for the case where the objective function of the subproblem i is monotonic decreasing w.r.t. variable x_i .

Example 2:

Consider the following example which has four variables and a total of ten constraints:

$$\begin{aligned} \text{Minimize } f(x) &= x_1 + x_2 + x_3 + x_3 x_4 + x_1 x_2 + x_2 x_4 \\ &\quad + x_1 x_2 x_4 + x_2 x_3 x_4 \end{aligned}$$

$$\text{Subject to:} \quad (3.12)$$

$$g_1: 8 + x_1 x_4 - 2 x_2 \geq 0$$

$$g_2: 12 + 4 x_1 - x_2 x_4 \geq 0$$

$$g_3: 12 + 3 x_1 x_4^2 - 4 x_2 \geq 0$$

$$g_4: 8 + 2 x_3 - x_2 x_4 \geq 0$$

$$g_5: 8 + x_3 x_2 - 2 x_4 \geq 0$$

$$g_6: 5 + x_3 - x_4 \geq 0$$

$$g_{(6+i)}: x_i \geq 0, \quad i = 1, 2, 3, 4.$$

Here we decompose the problem into two subproblems with x_2 and x_4 as the fixed variables for the first level, and x_1 and x_3 as the local variables for subproblem 1 and 2 respectively:

Subproblem 1:

$$\text{Minimize } f_1(x_2, x_4; x_1) = x_1 + x_2 + x_1 x_2 + x_1 x_2 x_4$$

Subject to: (3.13)

$$g_1: 8 + x_1 x_4 - 2 x_2 \geq 0$$

$$g_2: 12 + 4 x_1 - x_2 x_4 \geq 0$$

$$g_3: 12 + 3 x_1 x_4^2 - 4 x_2 \geq 0$$

$$g_7: x_1 \geq 0$$

Subproblem 2:

$$\text{Minimize } f(x_2, x_4; x_3) = x_3 + x_3 x_4 + x_2 x_3 x_4$$

Subject to: (3.14)

$$g_4: 8 + 2 x_3 - x_2 x_4 \geq 0$$

$$g_5: 8 + x_3 x_2 - 2 x_4 \geq 0$$

$$g_6: 5 + x_3 - x_4 \geq 0$$

$$g_8: x_3 \geq 0$$

In subproblem 1, w.r.t. variable x_1 , constraints g_1 , g_2 , g_3 and g_7 , are the candidate active constraints. Therefore:

$$x_1^* = \max \{ g'_{1,j} : j = 1, 2, 3, 7 \} \quad (3.15)$$

where

$$g'_{1,1} = (2x_2 - 8)/x_4 \quad (3.16)$$

$$g'_{1,2} = (x_2x_4 - 12)/12 \quad (3.17)$$

$$g'_{1,3} = (4x_2 - 12)/(3x_4^2) \quad (3.18)$$

$$g'_{1,7} = 0 \quad (3.19)$$

Similarly in subproblem 2 we have:

$$x_3^* = \max \{ g'_{2,j} : j = 4,5,6,8 \} \quad (3.20)$$

where

$$g'_{2,4} = (x_2x_4 - 8)/2 \quad (3.21)$$

$$g'_{2,5} = (2x_4 - 8)/x_2 \quad (3.22)$$

$$g'_{2,6} = (x_4 - 5) \quad (3.23)$$

$$g'_{2,8} = 0 \quad (3.24)$$

Therefore, the second level problem is written in the following form:

$$\begin{aligned} \text{Minimize } f(x_2, x_4; x_1^*, x_3^*) &= x_1^* + x_2 + x_3^* + x_3^*x_4 + x_1^*x_2 \\ &\quad + x_1x_4 + x_1^*x_2x_4 + x_2x_3^*x_4 \\ \text{Subject to:} &\quad (3.25) \end{aligned}$$

$$x_2 \geq 0, x_4 \geq 0$$

where:

$$x_1^* = \max \{ g'_{1,j} : j = 1,2,3,7 \}$$

$$x_3^* = \max \{ g'_{2,j} : j = 4,5,6,8 \}.$$

Figure 11, shows the two-level structure for this problem.

3.3 Several Subproblems with Several Local Variables

Here, we consider a problem which may be decomposed into two levels with several subproblems, each having several local variables:

$$\begin{aligned} &\text{Minimize } f(y;x) = f_0(y) + \sum_{i=1}^I f_i(y;X_i) \\ &\text{Subject to:} \end{aligned} \quad (3.26)$$

$$g_l(y) \leq 0 \quad l = 1, \dots, L$$

$$g_{i,j}(y; X_i) \leq 0 \quad j = 1, \dots, J$$

where i, j are the indices corresponding to the number of subproblems and number of constraints in each subproblem respectively. X_i is the vector of local design variables in subproblem i . The formulation for subproblem i is:

$$\begin{aligned} &\text{Minimize } f_i(y; X_i) \\ &\text{Subject to:} \end{aligned} \quad (3.27)$$

$$g_{ij}(y; X_i) \leq 0 \quad j = 1, \dots, J.$$

Here, we assume that the number of active constraints in each subproblem, i.e. inequalities which are satisfied in the form of equalities at the optimum, is the same as the number of local variables in that subproblem. Again, suppose that the monotonicity analysis is applicable to the subproblems, then following the same procedure as before, the second level problem is written in the following form:

$$\begin{aligned} &\text{Minimize } f(y; x) = f_0(y) + \sum_{i=1}^I f_i(y; X_i^*) \\ &\text{Subject to:} \end{aligned} \quad (3.28)$$

$$g_l(y) \leq 0 \quad l = 1, \dots, L$$

where X_i^* is found from the subproblem i , $i = 1, \dots, I$, as a function of the interreaction variables y using monotonicity analysis.

Example 3:

This example is selected from the collection of Hock and Schittkowski (1980). It has eight variables and a total of twenty two constraints:

$$\text{Minimize } f(x) = x_1 + x_2 + x_3$$

Subject to: (3.29)

$$\begin{aligned} g_1: & 1 - 0.0025 (x_4 + x_6) \geq 0 \\ g_2: & 1 - 0.0025 (x_5 + x_7 - x_4) \geq 0 \\ g_3: & 1 - 0.01 (x_8 - x_5) \geq 0 \\ g_4: & x_1 x_6 - 833.3325 x_4 - 100 x_1 + 83333.333 \geq 0 \\ g_5: & x_2 x_7 - 1250 x_5 - x_2 x_4 + 1250 x_4 \geq 0 \\ g_6: & x_3 x_8 - 1,250,000 - x_3 x_4 + 2500 x_5 \geq 0 \\ g_7: & 100 \leq x_1 \leq 10000 : & g_8 \\ g_{9,11}: & 1000 \leq x_i \leq 10000 : & g_{10,12} \quad i = 2,3 \\ g_{13}: & 10 \leq x_4 \leq 200 : & g_{14} \\ g_{15}: & 10 \leq x_5 \leq 300 : & g_{16} \\ g_{17}: & 100 \leq x_6 \leq 1000 : & g_{18} \\ g_{19}: & 200 \leq x_7 \leq 1000 : & g_{20} \\ g_{21}: & 300 \leq x_8 \leq 1000 : & g_{22} \end{aligned}$$

We select variables x_4 and x_5 as the fixed variables in the subproblems, and then decompose the problem into two levels with three subproblems in the first level:

Subproblem 1:

$$\text{Minimize } f_1(x_4, x_5; x_1, x_6) = x_1$$

Subject to: (3.30)

$$\begin{aligned}
 g_1: \quad & 1 - 0.0025 (x_4 + x_6) \geq 0 \\
 g_4: \quad & x_1 x_6 - 833.33252 x_4 - 100 x_1 + 83333.333 \geq 0 \\
 g_7: \quad & 100 \leq x_1 \leq 10,000 : \quad g_8 \\
 g_{17}: \quad & 100 \leq x_6 \leq 1000 : \quad g_{18}.
 \end{aligned}$$

Subproblem 2:

Minimize $f_2 (x_4, x_5; x_2, x_7) = x_2$

Subject to: (3.31)

$$\begin{aligned}
 g_2: \quad & 1 - 0.0025 (x_5 + x_7 - x_4) \geq 0 \\
 g_5: \quad & x_2 x_7 - 1250 x_5 - x_2 x_4 + 1250 x_4 \geq 0 \\
 g_9: \quad & 1000 \leq x_2 \leq 10000 : \quad g_{10} \\
 g_{19}: \quad & 200 \leq x_7 \leq 1000 : \quad g_{20}
 \end{aligned}$$

Subproblem 3:

Minimize $f_3(x_4, x_5; x_3, x_8) = x_3$

Subject to: (3.32)

$$\begin{aligned}
 g_3: \quad & 1 - 0.01 (x_8 - x_5) \geq 0 \\
 g_6: \quad & x_3 x_8 - 1,250,000 - x_3 x_5 + 2500 x_5 \geq 0 \\
 g_{11}: \quad & 1000 \leq x_3 \leq 10000 : \quad g_{12} \\
 g_{21}: \quad & 300 \leq x_8 \leq 1000 : \quad g_{22}
 \end{aligned}$$

The second-level problem is:

Minimize $f = f_1 + f_2 + f_3$

Subject to: (3.33)

$$\begin{aligned}
 g_{13}: \quad & 10 \leq x_4 \leq 200 : \quad g_{14} \\
 g_{15}: \quad & 10 \leq x_5 \leq 300 : \quad g_{16}.
 \end{aligned}$$

Now, let us apply monotonicity rules to the subproblems:

In subproblem 1, according to the first rule of monotonicity, w.r.t. variable x_1 constraints g_4 (if $x_6 > 100$) and g_7 are the candidate active constraints. Furthermore, based on the second rule of monotonicity, if g_4 is active, then w.r.t. variable x_6 constraints g_1 and g_{18} are also the candidate active constraints. Therefore, from subproblem 1 we have:

$$\text{where } x_1^* = \max \{g'_{1,4}, g'_{1,7}\} \quad (3.34)$$

$$g'_{1,4} = (833.3325x_4 - 83333.333)/(x_6 - 100) \quad (3.35)$$

$$g'_{1,7} = 100 \quad (3.36)$$

and If $x_1^* = g'_{1,4}$, Then:

$$x_6^* = \min \{g'_{1,1}, g'_{1,18}\} \quad (3.37)$$

where

$$g'_{1,1} = 1/0.0025 - x_4 \quad (3.38)$$

$$g'_{1,18} = 1000 \quad (3.39)$$

otherwise $100 \leq x_6^* \leq 1000$, as long as it is feasible.

Likewise in Subproblem 2:

$$x_2^* = \max \{g'_{2,5}, g'_{2,9}\} \quad (3.40)$$

where

$$g'_{2,5} = (1250 x_5 - 1250 x_4)/(x_7 - x_4) \quad (3.41)$$

$$g'_{2,9} = 1000 \quad (3.42)$$

and If $x_2^* = g'_{2,5}$ Then:

$$x_7^* = \min \{g'_{2,2}, g'_{2,20}\} \quad (3.43)$$

where

$$g'_{2,2} = 1/0.0025 + x_4 - x_5 \quad (3.44)$$

$$g'_{2,20} = 1000 \quad (3.45)$$

otherwise $200 \leq x_7^* \leq 1000$, as long as it is feasible.

Finally, in Subproblem 3:

$$x_3^* = \max \{ g'_{3,6}, g'_{3,11} \} \quad (3.46)$$

where

$$g'_{3,6} = (1,250,000 - 2,500 x_5)/(x_8 - x_5) \quad (3.47)$$

$$g'_{3,11} = 1000 \quad (3.48)$$

and If $x_3^* = g'_{3,6}$ Then:

$$x_8^* = \min \{ g'_{3,3}, g'_{3,22} \} \quad (3.49)$$

where

$$g'_{3,3} = 100 + x_5 \quad (3.50)$$

$$g'_{3,22} = 1000 \quad (3.51)$$

otherwise, x_8^* may be any value in the range of $300 \leq x_8^* \leq 1000$, as long as it is feasible. The two-level structure for this example is given in Figure 12.

3.4 Some Test Results

The methodology described in sections 3.1 - 3.3 has been combined with a sequential quadratic programming technique of the type suggested by Powell (1978). For each problem the first-level problem is solved by the global monotonicity analysis and the second-level one using the VMCON (Crane et al., 1980) program. We then selected a set of small to medium-sized problems to evaluate the performance of VMCON with and without using the monotonicity-

based decomposition. The results are shown in Table 1, where:

TP: Test Problem number

TP-[.]: Problem number in the reference [.], where

HS := Hock and Schittkowski (1983)

AP := Azarm and Papalambros (1984)

A := Azarm (1984)

NV: Number of the variable

NC: Number of constraints

NF/NG: Number of objective or constraint function evaluations

NDF/NDG: Number of gradient of objective or constraint functions evaluations

CPU: Central processing time in seconds

VM: VMCON program

MBD-VM: Combined Monotonicity-Based Decomposition and the VMCON program

The test results which are demonstrated in Table 1, indicates clearly the advantage of coupling the monotonicity-based decomposition into the VMCON program.

4. Concluding Remarks

The global monotonicity analysis has already been applied to many small to medium-sized engineering design problems within a single-level framework (see Papalambros and Li (1983), where further references can be found). In this paper we have demonstrated how the global monotonicity analysis can be applied within a two-level framework, and improve the performance of a conventional (single-level) optimization algorithm. However, for large and complex problems, the application of the global monotonicity to the subproblems may be

inhibited by the need for extensive algebraic manipulations. In that case, an extension of the method presented here should be considered where in the first-level subproblems the local (Azarm and Papalambros, 1984) rather than the global monotonicity analysis should be used.

5. Acknowledgement

The work of the first author was supported through NSF Grant DMC - 85 - 05113. The work of the second author was supported partially through NSF Grant DMC - 85 - 05113, and partially through NSF Grant CDR - 85 - 00108.

6. References

Azarm, S., 1984, "Local Monotonicity in Optimal Design," Ph.D. Thesis, The University of Michigan, Ann Arbor, Michigan.

Azarm, S., Papalambros, P., 1984, "An Automated Procedure for Local Monotonicity Analysis," ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 106, No. 1, pp. 82-89.

Crane, R. L., Hillstorm, K. E., Minkoff, M., 1980, "Solution of the General Nonlinear Programming Problems with Subroutine VMCON," Argonne National Lab.

Dantzig, G. B., 1963, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey.

Dantzig, G. B., Wolfe, P., 1960, "Decomposition Principle for Linear Programs," Operations Research 8, pp. 101-111.

Gabriele, G. A., Ragsdell, K. M., 1980, "Large Scale Nonlinear Programming Using The Generalized Reduced Gradient Method," ASME Trans., Journal of Mechanical Design, Vol. 102, July, pp. 566-573.

Himmelblau, D. M., 1973, Decomposition of Large-Scale Problems, North-Holland Publishing Company.

Hock, W., Schittkowski, K., 1980, Test Examples for Nonlinear Programming Codes, Springer-Verlag, New York.

Johnson, R. C., Benson, R. C., 1984a, "A Basic Two-Stage Decomposition Strategy for Design Optimization," ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 106, pp. 380-386.

Johnson, R. C., Benson, R. C., 1984b, "A Multi-Stage Decomposition Strategy for Design Optimization," ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 106, pp. 387-393.

Karmarker, N., 1984, "A New Polynomial-Time Algorithm for Linear Programming," Proceedings of the 16th Annual ACM Symposium on the Theory of Computing, pp. 302-311.

Kirsch, U., 1975, "Multilevel Approach to Optimum Structural Design," Journal of the Structural Division, Vol. 101, No. ST4, pp. 957-974.

Kirsch, U., 1981, Optimum Structural Design, McGraw-Hill Company, pp. 198-215.

Kron, G., 1954, "A Method of Solving a Very Large Physical Systems in Easy Stages," Proceedings of the Institute of Radio Engineers, Vol 42, No. 4, pp. 680-686.

Lasdon, L. S., 1968, "Duality and Decomposition in Mathematical Programming," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, pp. 86-100.

Lasdon, L. S., 1970, Optimization Theory for Large Systems, The Macmillan Company, London.

Lasdon, L. S., Waren, A. D., 1983, "GRG2 User's Guide."

Mesarovic, M. D., Macko, D., Takahara, Y., 1970, Theory of Hierarchical, Multilevel, Systems, Academic Press, New York and London.

Murtagh, B. A., Saunders, M. A., 1982, "A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints," Mathematical Programming Study 16, pp. 84-117.

Papalambros, P., Li, H.L., 1983, "Notes on the Operational Utility of Monotonicity in Optimization," ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design, Vol 105, No. 2, pp. 174-183.

Papalambros, P., Wilde, D. J., 1979, "Global Non-Iterative Design Optimization Using Monotonicity Analysis," ASME Trans., Journal of Mechanical Design, Vol. 101, Np. 4, pp. 643-649.

Papalambros, P., Wilde, D. J., 1980, "Regional Monotonicity in Optimum Design," ASME Trans., Journal of Mechanical Design, Vol. 104, No. 1, pp. 497-500.

Powell, M. J. D., 1978, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," Proceeding of the 1977 Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics, Vol. 630, Springer-Verlog, Berlin, pp. 144-157.

Reklaitis, G. V., Ravindran, A., Ragsdell, K. M., 1983, Engineering Optimization, John Wiley & Sons Inc.

Siddall, J. N., 1982, Optimal Engineering Design, Marcell-Dekker Inc.

Vanderplaatz, G. N., 1984, Numerical Optimization Techniques for Engineering Design, McGraw-Hill Inc.

Wilde, D. J., 1975, "Monotonicity and Dominance in Hydraulic Cylinder Design," ASME Trans., Journal of Engineering for Industry, Vol. 94, No. 4, pp. 1390-1394.

Wismer, D. A., 1971, Optimization Methods for Large-Scale Systems, McGraw-Hill Company.

Wismer, D. A., Chattergy, R., 1978, Introduction to Nonlinear Optimization, North-Holland, New York, pp. 325-349.

TABLE 1: SOME TEST RESULTS

TP	TP-[.]	NV	NC	NF/NG		NDF/NDG		CPU	
				VM	MBD-VM	VM	MBD-VM	VM	MB-VM
1	22-[HS]	2	2	25	16	8	15	1.56	1.43
2	15-[HS]	2	3	16	5	5	2	1.48	1.45
3	17-[HS]	2	5	34	15	11	7	2.20	1.49
4	20-[HS]	2	5	58	5	19	2	2.44	1.37
5	21-[HS]	2	5	13	7	4	3	1.53	1.36
6	18-[HS]	2	6	34	27	11	13	2.30	1.56
7	19-[HS]	2	6	19	21	6	10	1.55	1.53
8	23-[HS]	2	9	19	13	6	6	2.10	1.45
9	29-[HS]	3	1	45	43	11	14	2.22	1.52
10	35-[HS]	3	4	29	16	7	5	2.40	1.39
11	33-[HS]	3	6	21	22	5	7	2.30	1.43
12	30-[HS]	3	7	37	10	9	3	2.25	1.44
13	31-[HS]	3	7	37	28	9	9	2.26	2.13
14	36-[HS]	3	7	9	7	2	2	1.51	1.45
15	65-[HS]	3	7	29	28	7	9	2.33	2.15
16	34-[HS]	3	8	29	22	7	7	2.23	2.20
17	37-[HS]	3	8	37	28	9	9	2.31	2.20
18	66-[HS]	3	8	25	13	6	4	2.27	1.55
19	76-[HS]	4	7	26	33	5	8	2.34	2.16
20	73-[HS]	4	7	16	13	3	3	2.40	1.55
21	44-[HS]	4	10	31	7	6	2	2.47	1.45
22	71-[HS]	4	10	31	25	6	6	2.14	2.10
23	106-[HS]	8	22	325	25	36	8	39.22	2.15
24	113-[HS]	10	8	166	323	15	46	9.14	5.28
25	2-[AP]	2	10	19	11	6	5	2.8	1.43
26	--[A]	7	25	41	29	5	4	3.58	2.57

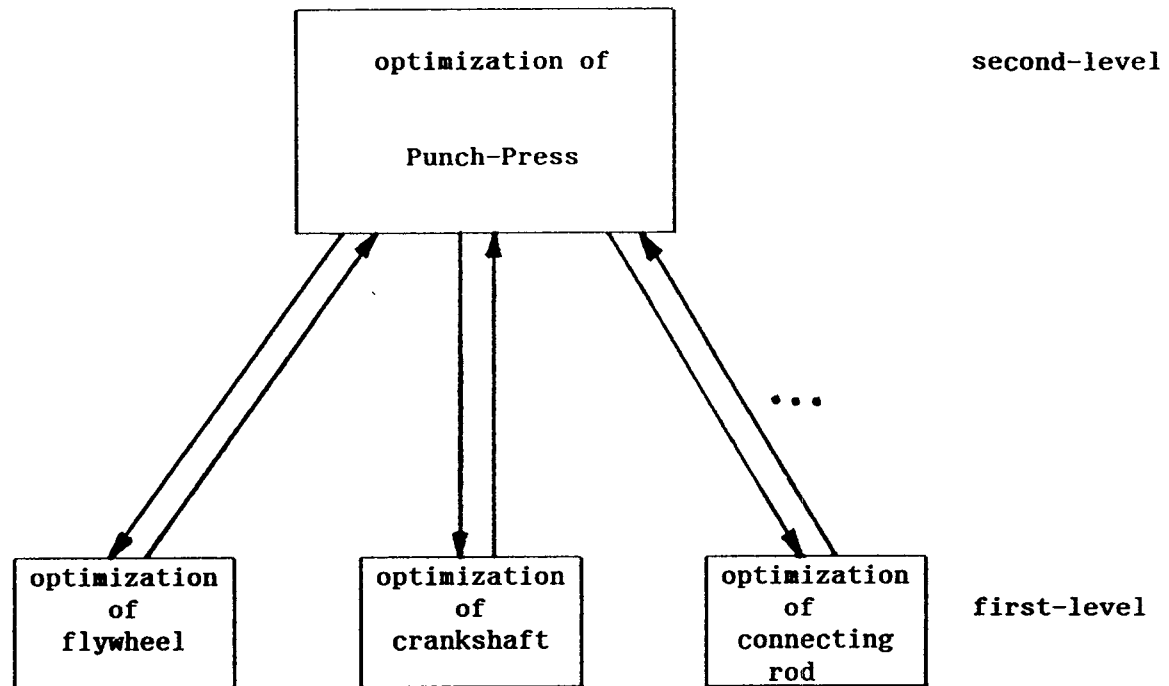


FIGURE 1 TWO-LEVEL OPTIMIZATION OF A PUNCH-PRESS

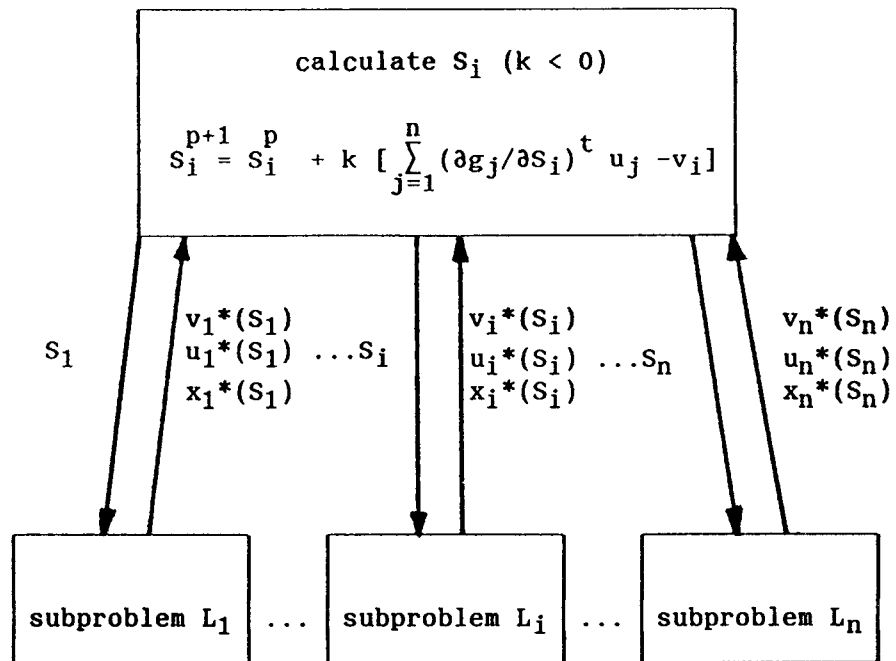


FIGURE 2 TWO-LEVEL STRUCTURE OF THE LAGRANGIAN FEASIBLE METHOD

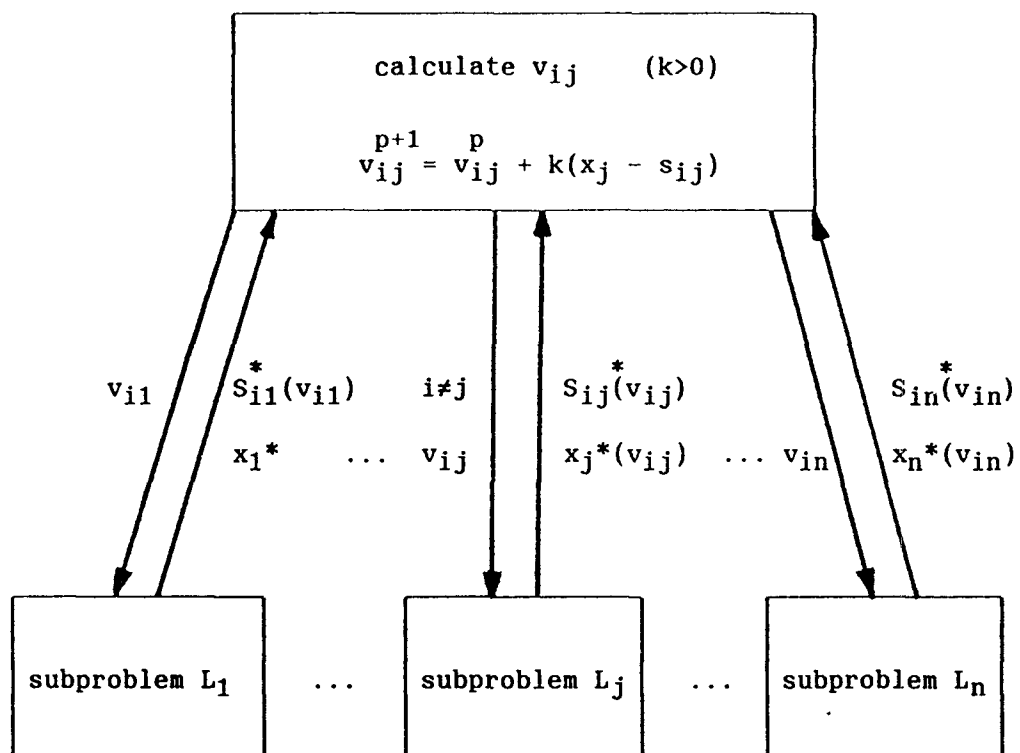


FIGURE 3 TWO-LEVEL STRUCTURE OF THE LAGRANGIAN DUAL FEASIBLE METHOD

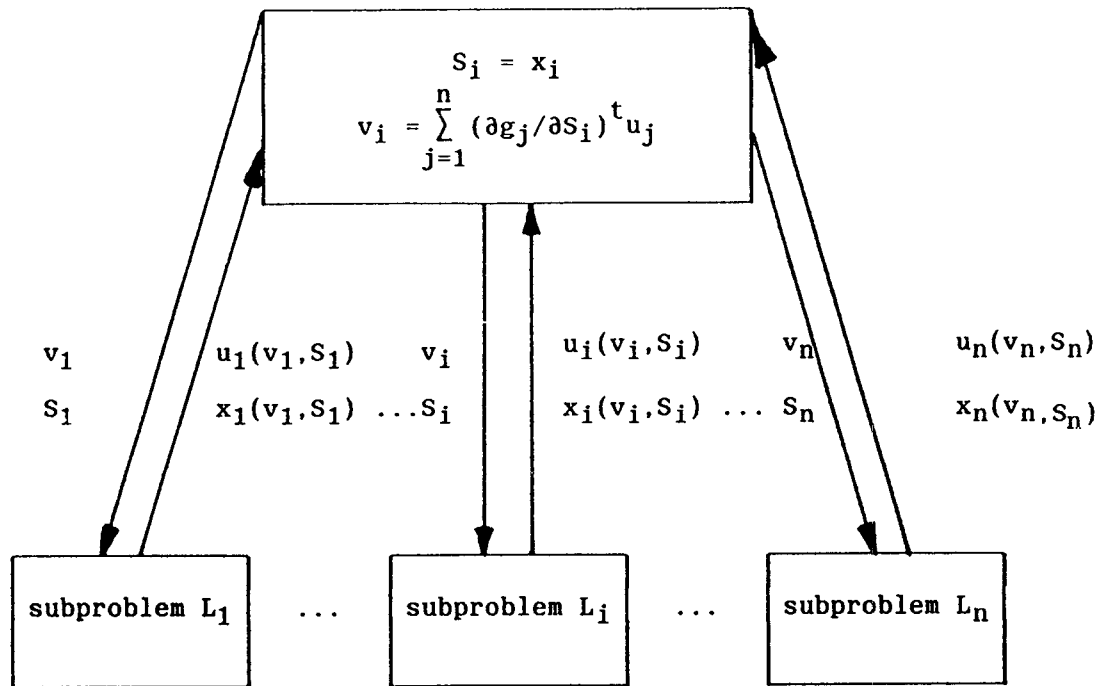


FIGURE 4 TWO-LEVEL STRUCTURE OF THE RELAXATION METHOD

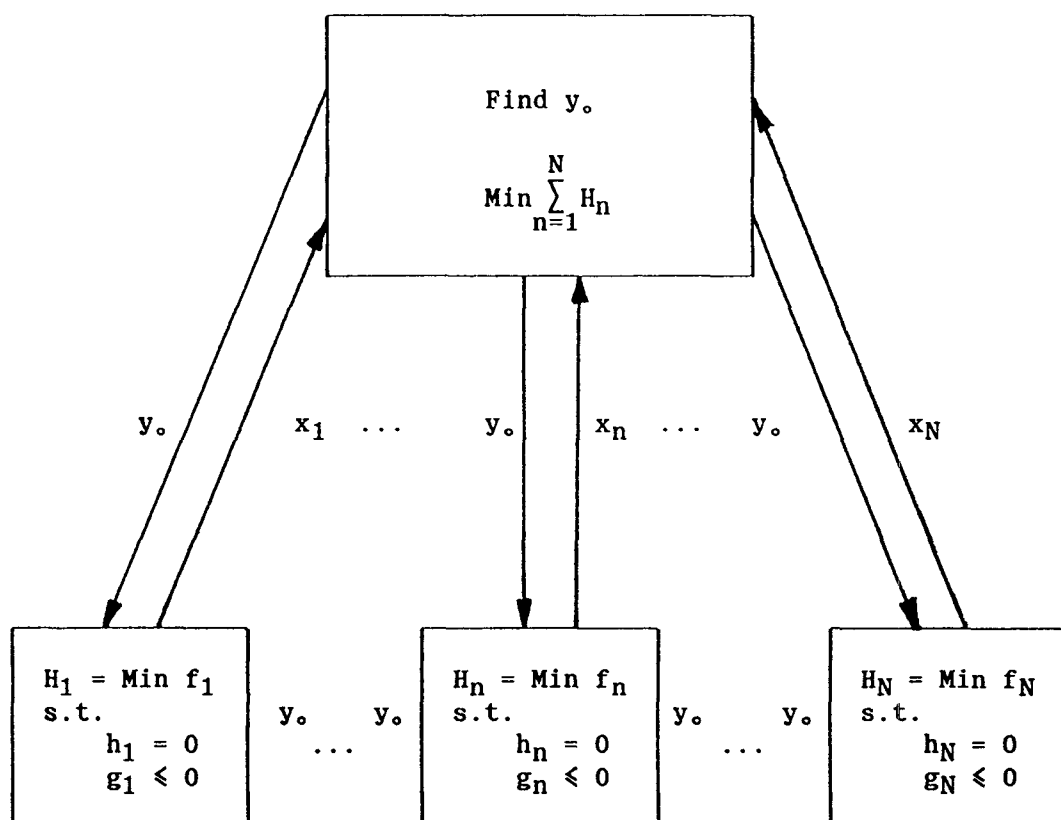


FIGURE 5 TWO-LEVEL STRUCTURE OF THE FEASIBLE METHOD

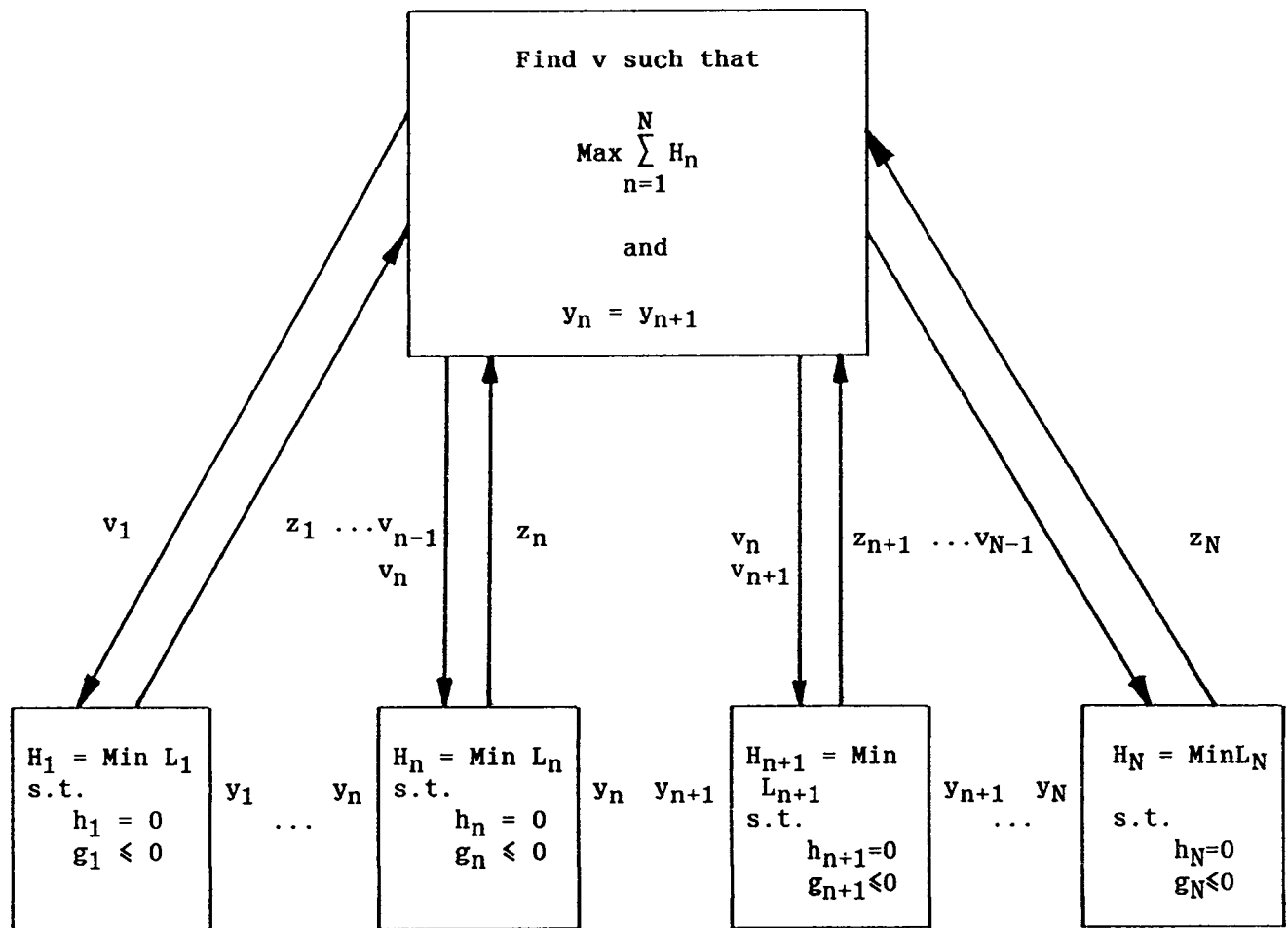


FIGURE 6 TWO-LEVEL STRUCTURE OF THE DUAL-FEASIBLE METHOD

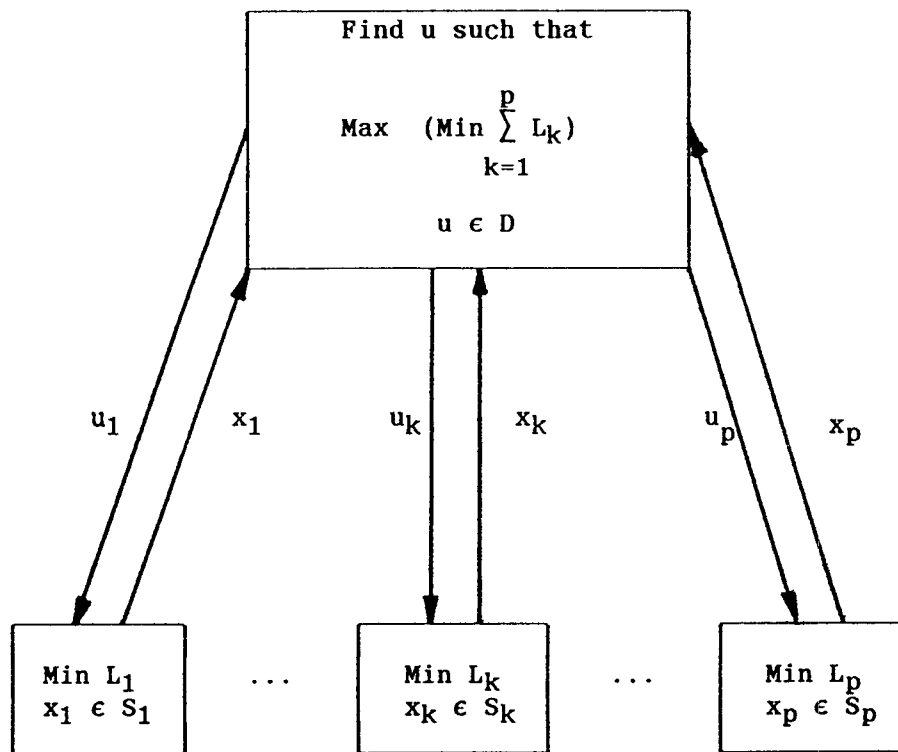


FIGURE 7 TWO-LEVEL STRUCTURE OF THE DUAL METHOD

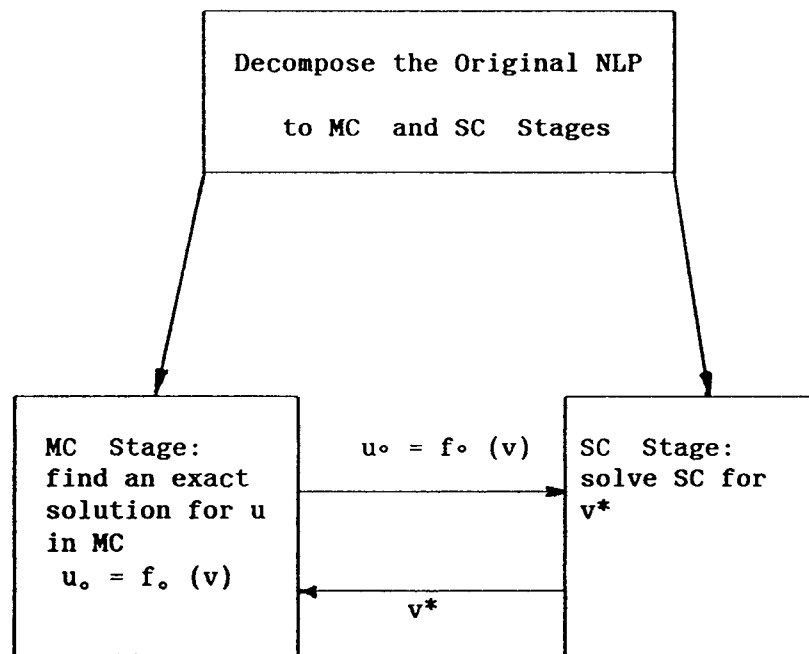


FIGURE 8 TWO-STAGE STRUCTURE OF THE JOHNSON'S METHOD

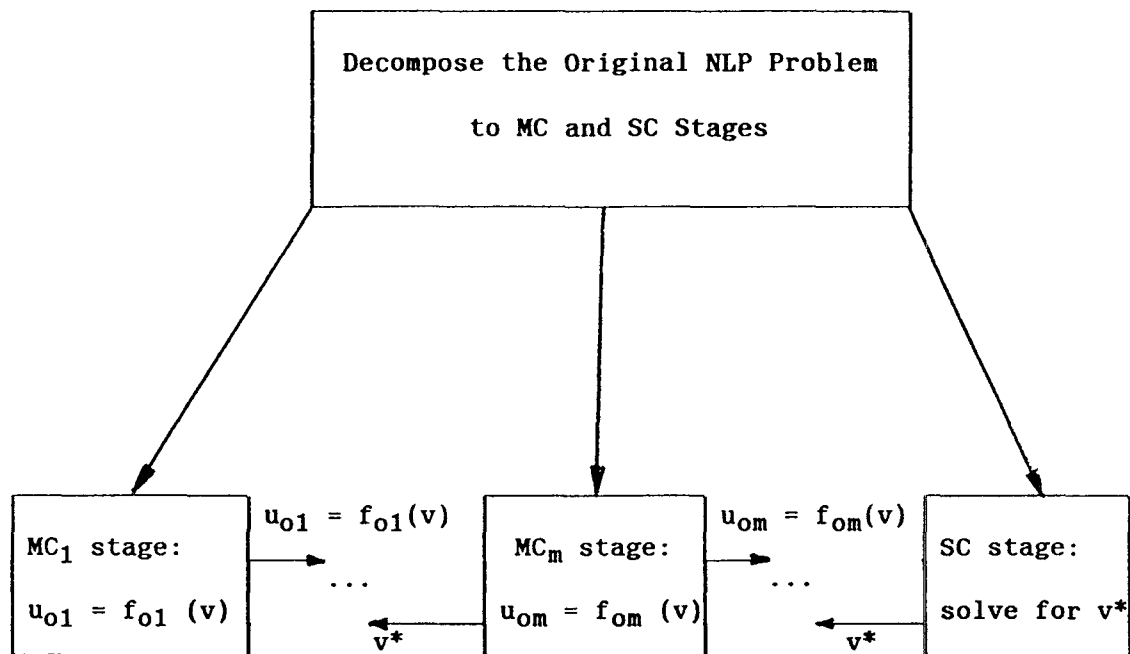


FIGURE 9 MULTISTAGE STRUCTURE OF JOHNSON'S METHOD

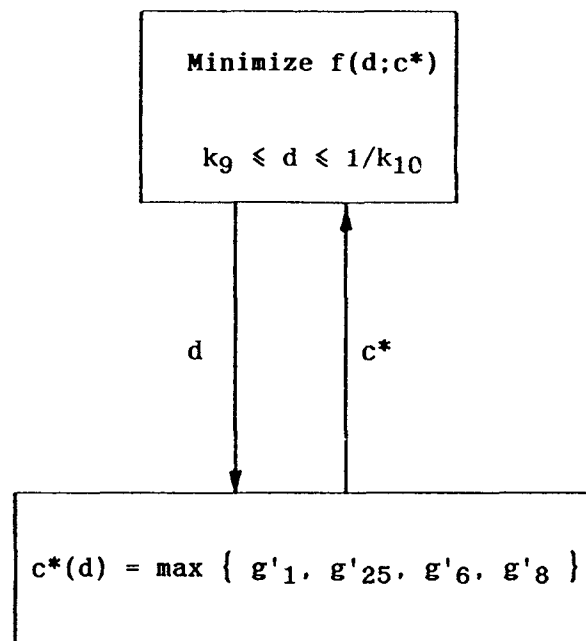


FIGURE 10 TWO-LEVEL STRUCTURE OF EXAMPLE 1

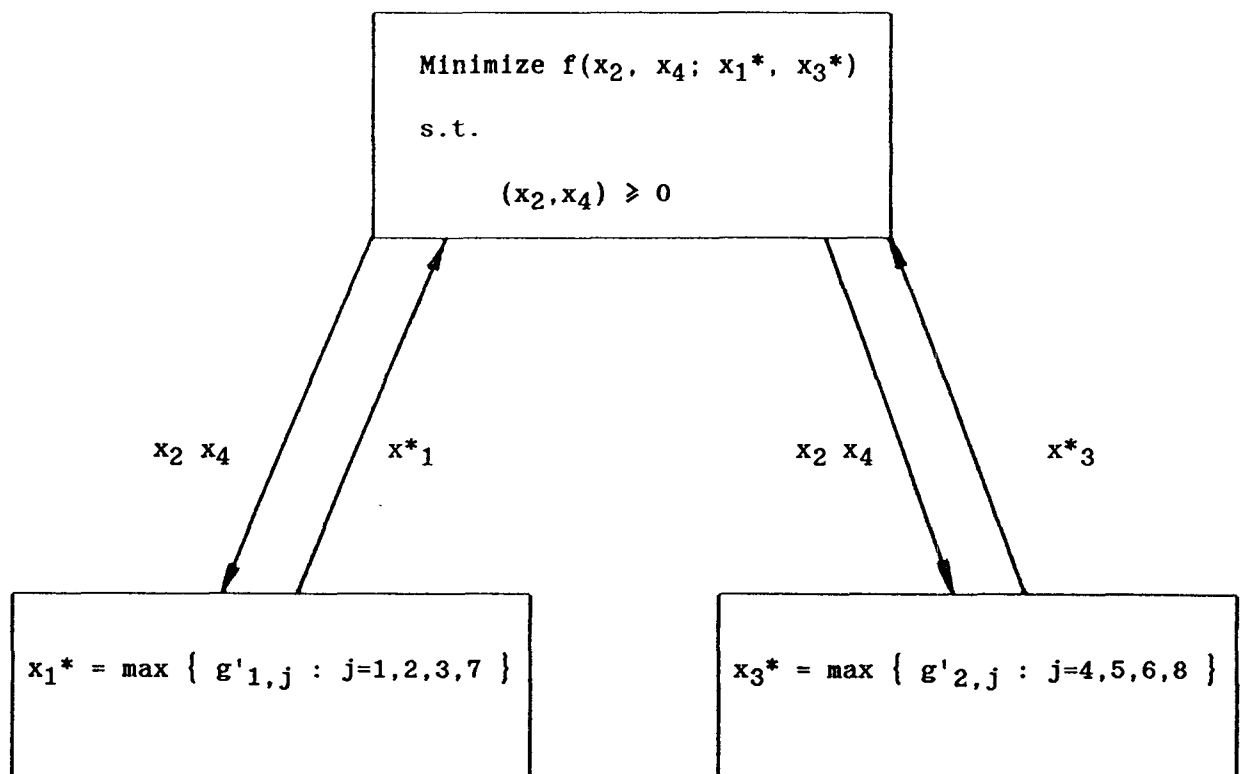


FIGURE 11 TWO-LEVEL STRUCTURE FOR EXAMPLE 2

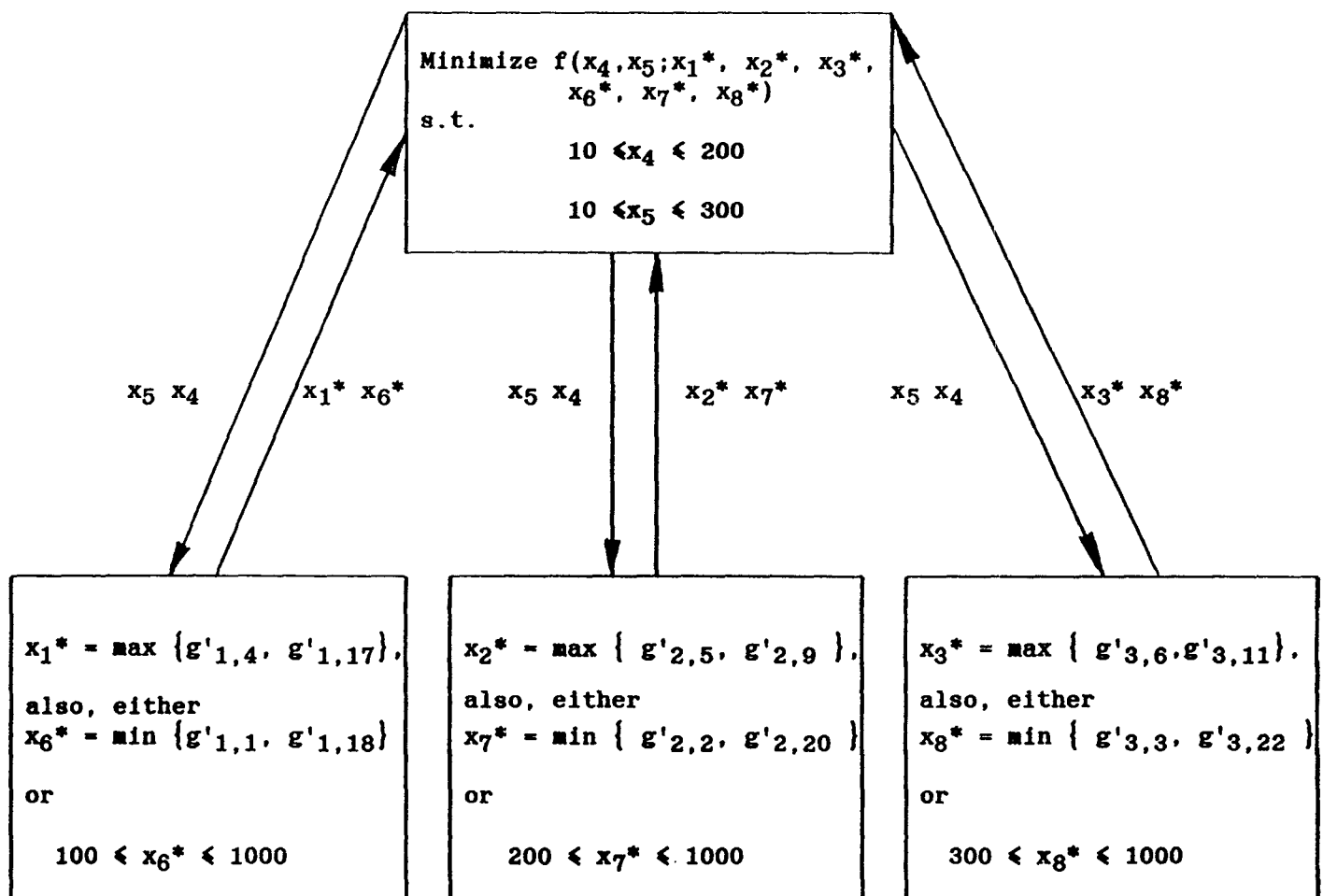


FIGURE 12 TWO-LEVEL STRUCTURE OF EXAMPLE 3