

**OPTIMAL HIERARCHICAL
DECOMPOSITION SYNTHESIS
USING INTEGER PROGRAMMING**

Ramprasad S. Krishnamachari
Graduate Student
and
Panos Y. Papalambros
Professor

Technical Report 95-17

November, 1995

OPTIMAL HIERARCHICAL DECOMPOSITION SYNTHESIS USING INTEGER PROGRAMMING

Ramprasad S. Krishnamachari

Graduate Student

and

Panos Y. Papalambros[†]

Professor

Design Laboratory
Department of Mechanical Engineering
and Applied Mechanics
The University of Michigan
Ann Arbor, Michigan

ABSTRACT

Decomposition synthesis in optimal design is the process of creating an optimal design model by selecting objectives and constraints so that it can be directly partitioned into an appropriate decomposed form. Such synthesis results are not unique since there may be many partitions that satisfy the decomposition requirements. Introducing suitable criteria an *optimal* decomposition synthesis process can be defined in a manner analogous to optimal partitioning formulations. The article presents an integer programming formulation and solution techniques for synthesizing hierarchically decomposed optimal design problems. Examples for designing a pressure vessel, an automotive caliper disc brake and a speed reducer are also presented.

November, 1995

[†] Corresponding author

INTRODUCTION

The general design problem (GDP) is defined formally as

$$\begin{aligned} \text{GDP:} \quad & \textit{find} \quad \mathbf{x} \in X \\ & \textit{subject to} \quad \mathbf{h}(\mathbf{x}, \mathbf{p}) = \mathbf{0} \\ & \quad \quad \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \end{aligned} \tag{1}$$

where \mathbf{h} , \mathbf{g} represent vectors of design criteria generally assumed to be nonlinear functions of the design variables \mathbf{x} and parameters \mathbf{p} , and X is the set constraint on the design variables, imposing additional restrictions on \mathbf{x} , such as discreteness. The GDP is transformed to an optimal design problem (ODP) by selecting one or more design criteria from Eq. (1) and composing a scalar objective as follows:

$$\begin{aligned} \text{ODP:} \quad & \textit{minimize} \quad f(\mathbf{x}, \mathbf{p}') = \sum_{i=1}^K f_i(\mathbf{x}_i, \mathbf{p}') \\ & \textit{subject to} \quad \mathbf{h}(\mathbf{x}, \mathbf{p}) = \mathbf{0} \\ & \quad \quad \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq \mathbf{0} \\ & \textit{and} \quad \quad \quad \mathbf{x} \in X \end{aligned} \tag{2}$$

where \mathbf{p}' is a vector of parameters that may include weights used in composing the scalar objective $f(\mathbf{x}, \mathbf{p}')$ to be minimized, \mathbf{x}_i is a *subvector* (any vector defined from the *components* of a given vector) of \mathbf{x} , and $f_i(\mathbf{x}_i, \mathbf{p}')$ are independent criteria selected from \mathbf{h} and \mathbf{g} to define the objective.

Synthesis of the ODP is based on a variety of subjective considerations, generally regarding (i) the type of knowledge available and/or desirable for the underlying design problem and (ii) expedience in solving the resulting mathematical optimization problem. Design of large complex engineering systems modeled as nonlinear programming problems can benefit from the use of decomposition strategies; see Papalambros (1995) for a review. Synthesizing ODP's that can be decomposed and solved in a systematic way is then desirable. A “hierarchical decomposition synthesis” methodology where a hierarchically decomposed ODP is obtained starting from a GDP was proposed by Krishnamachari and Papalambros (1995) — abbreviated as

K&P (1995) in the sequel. The ODP thus obtained can be partitioned and solved with a prescribed decomposition method.

Synthesis of a decomposable ODP is a natural extension of the *decomposition analysis* methodology developed for partitioning already formulated ODP's into appropriate decomposed forms (Wagner 1993, Papalambros 1995). Given an undecomposed GDP (or ODP for that matter) there are many ways of obtaining a decomposed ODP. It is then natural to attempt to formalize a process for selecting the "best" such ODP relative to some formal criteria. This process is termed *optimal decomposition synthesis* and it is analogous to the optimal partitioning methods developed for decomposition analysis (Michelena and Papalambros 1994, 1995).

The methodology proposed in K&P (1995) mainly focuses on synthesizing ODP's that can be solved by a primal hierarchical decomposition method (Wagner 1993). A block-angular structure is first identified in the GDP. An ODP is then created that can be hierarchically decomposed based on this structure. Formally, a GDP is first cast into the form

$$\begin{aligned} \mathbf{g}_0(\mathbf{x}_0) &\leq \mathbf{0} \\ \mathbf{h}_0(\mathbf{x}_0) &= \mathbf{0} \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{g}_i(\mathbf{x}_0, \mathbf{x}_i) &\leq \mathbf{0} & i = 1, \dots, K \\ \mathbf{h}_i(\mathbf{x}_0, \mathbf{x}_i) &= \mathbf{0} & i = 1, \dots, K \end{aligned} \tag{4}$$

that has a master problem and K subproblems with the block-angular structure of Fig. 1. A hierarchically decomposed ODP is synthesized by composing a weighted additive objective selecting criteria from Eq. (3) and (4) as shown in Eq. (5) and Fig. 2. As noted in K&P (1995) the synthesized objective may include components only from Eq. (3) or Eq. (4).

$$\begin{aligned} \underset{\mathbf{x}_0, \mathbf{x}_i}{\text{minimize } f} &= f_0(\mathbf{x}_0, w_0) + \sum_{i=1}^K f_i(\mathbf{x}_0, \mathbf{x}_i, w_i) \\ \text{subject to: } & \mathbf{g}_0(\mathbf{x}_0) \leq \mathbf{0} \\ & \mathbf{h}_0(\mathbf{x}_0) = \mathbf{0} \\ & \mathbf{g}_i(\mathbf{x}_0, \mathbf{x}_i) \leq \mathbf{0} \quad i = 1, \dots, K \\ & \mathbf{h}_i(\mathbf{x}_0, \mathbf{x}_i) = \mathbf{0} \quad i = 1, \dots, K \end{aligned} \tag{5}$$

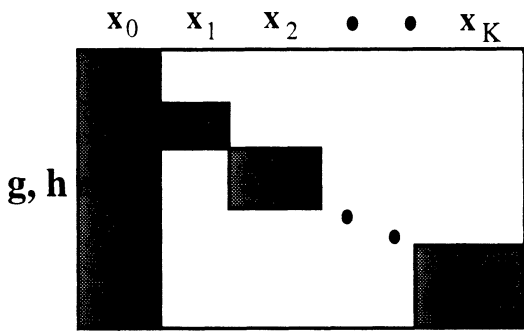


Fig. 1 GDP with block-angular structure

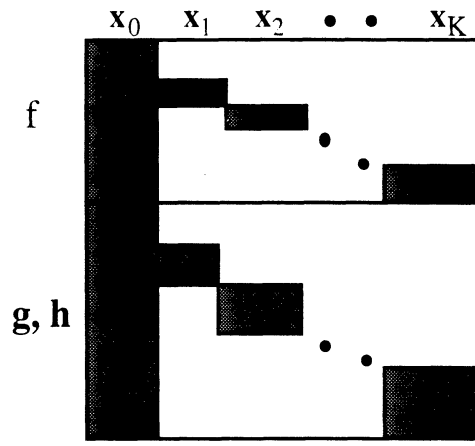
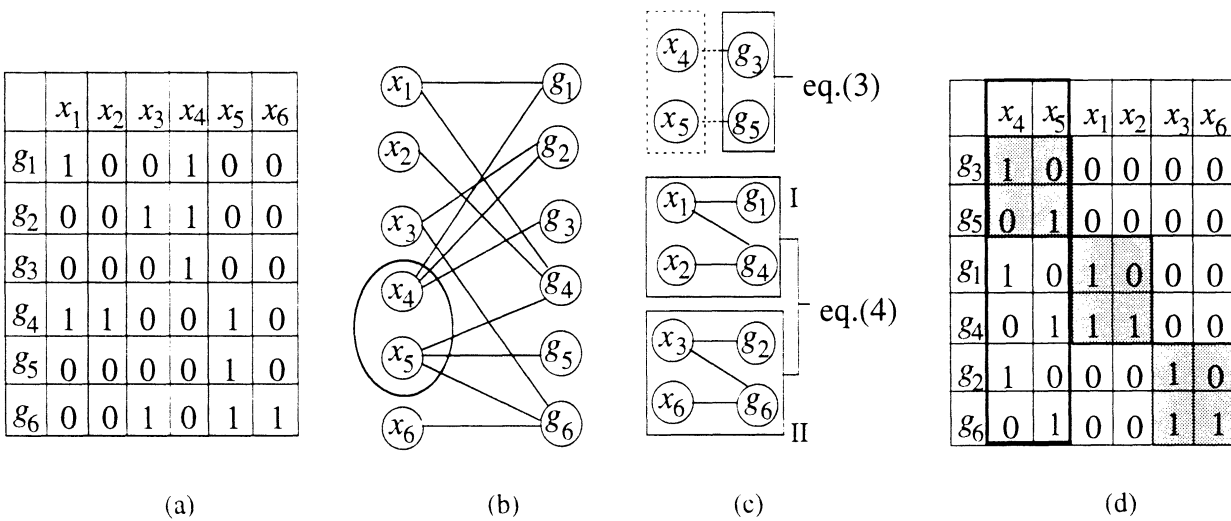


Fig. 2 Synthesized ODP

Identification of the block-angular structure in the GDP is performed using graph partitioning. This procedure proposed in K&P (1995) can be illustrated using the following example. Consider a GDP with a functional dependence table (FDT) as in Fig. 3(a). Linking variables $\{x_4, x_5\}$ are selected using the graph representation of the FDT, Fig. 3(b). The vertices corresponding to these variables are removed and the graph splits into pieces. The partitions corresponding to Eq. (3-4) are shown in Fig. 3(c). The partitioned GDP now has the FDT shown in Fig. 3(d).



(a)

(b)

(c)

(d)

Fig. 3 Steps in identifying block-angular structure in the GDP

Partitions I and II in Fig. 3(c) represented as the connected components (blocks) of the system in Fig. 3(d) is not the only way of creating a model structure in the form of Eq. (4). For example, a single larger cluster resulting from combining these two partitions would still satisfy Eq. (4). In general, selection of linking variables and block composition is not unique for a given desirable structure. If this selection can be guided by a formal objective criterion, a combinatorial optimization problem can be posed to find the “most suitable” structure, a process we termed optimal decomposition synthesis. The next section describes an integer programming model for this problem and a solution strategy when connectivity within each cluster is not strictly required. Direct graph partitioning methods that do account for connectivity are briefly discussed in the subsequent section. The methods are then demonstrated and compared in some small but illustrative design examples.

INTEGER PROGRAMMING FORMULATION

Considering the GDP in Eqs. (3) and (4), we refer to the linking variables \mathbf{x}_0 and associated functions $\mathbf{h}_0, \mathbf{g}_0$ as the *master cluster*, and the local variables \mathbf{x}_i and functions $\mathbf{h}_i, \mathbf{g}_i$ as the *subclusters* (for $i = 1, \dots, K$). Note that these clusters will correspond to the master problem and subproblems, respectively, in the hierarchically decomposed ODP to be synthesized.

To proceed with the formulation we assume that there are two desirable characteristics of the final ODP. The model blocks should be (i) relatively small in size to facilitate comprehension and computation, (ii) of approximately the same size to facilitate validation, parametric studies, and load balancing in case of parallel solution. The first characteristic leads to an objective function that attempts to minimize both the size of the master cluster and the average size of the subclusters: *minimize* w_m (*size of the master cluster*) + w_s (*average size of the subclusters*), where w_m, w_s are weights. The second characteristic imposes the constraint K_s (*size of the smallest subcluster*) \geq (*size of the largest subcluster*), where $K_s \geq 1$ is a size factor. The size of a cluster is defined as equal to the sum of the number of variables *and* the number of design criteria (functions) that it contains.

Note that this formulation does not account for changes in the cluster sizes resulting from the composition of the system objective. Also recall that block-angular structures such as that of Fig. 3 may include a subcluster (block) that is not a (single) connected component. This issue will be addressed in the next section.

In the formulation, each cluster is designated by its variables and functions. All variables and functions in the GDP must be assigned. The master cluster contains all design criteria that are functions exclusively of the linking variables. A local variable belongs to a subcluster if the function that depends on that variable is in the subcluster. Each function can belong to only one subcluster. Functions belonging to two different subclusters cannot have any common variables other than the linking ones. An integer linear programming (ILP) model is created whose zero-one variables indicate what cluster the design variables and functions are assigned to. The model is kept linear by assuming that the number of subclusters K is fixed during optimization. A post-optimal parametric study on K can be then conducted. It is assumed throughout $w_m = w_s = 1.0$. The ILP model is advantageous since it represents a difficult but well-studied optimization problem. The global optimum may be found using branch and bound or cutting plane methods, and a lower bound on the ILP solution can be always obtained by solving the relaxed continuous LP (see, e.g., Papadimitriou and Steiglitz 1982).

To proceed with the ILP model define the following:

i cluster index, $i = 0, \dots, K$ (zero corresponding to the master cluster)

j function (criterion) index in the GDP, $j = 1, \dots, T$

v variable index in the GDP, $v = 1, \dots, N$

d_j number of variables in function j

$$a_{jv} = \begin{cases} 1 & \text{if function } j \text{ contains variable } v \\ 0 & \text{otherwise} \end{cases}$$

$$e_{iv} = \begin{cases} 1 & \text{if cluster } i \text{ contains variable } v \\ 0 & \text{otherwise} \end{cases}$$

$$s_{ij} = \begin{cases} 1 & \text{if cluster } i \text{ contains function } j \\ 0 & \text{otherwise} \end{cases}$$

K_s cluster relative size constant, usually $3 \geq K_s \geq 1$

S_i size of cluster i

The mathematical model for the optimal synthesis problem is now stated as follows:

$$\text{minimize } \sum_{v=1}^N e_{0v} + \sum_{j=1}^T s_{0j} + (1/K) \sum_{i=1}^K S_i \quad (6)$$

subject to:

$$h_1: S_i = \sum_{j=1}^T s_{ij} + \sum_{v=1}^N e_{iv} \quad i = 1, \dots, K$$

$$h_2: \sum_{i=0}^K s_{ij} = 1 \quad j = 1, \dots, T$$

$$h_3: \sum_{i=0}^K e_{iv} = 1 \quad v = 1, \dots, N$$

$$g_1: K_s S_i \geq S_q, \quad q \neq i, \quad i, q = 1, 2, \dots, K$$

$$g_2: \sum_{v=1}^N a_{jv} e_{0v} \geq d_j s_{0j} \quad j = 1, \dots, T$$

$$g_3: \sum_{v=1}^N a_{jv} e_{0v} \leq d_j - \sum_{i=1}^K s_{ij} \quad j = 1, \dots, T$$

$$g_4: e_{iv} \geq \left(\sum_j a_{jv} s_{ij} / \sum_j a_{jv} \right) - e_{0v} \quad v = 1, \dots, N \text{ and } i = 1, \dots, K$$

$$g_5: e_{iv} \leq \sum_j a_{jv} s_{ij} \quad v = 1, \dots, N \text{ and } i = 1, \dots, K$$

$$g_6: \sum_{v=1}^N e_{iv} \geq 1 \quad i = 1, \dots, K$$

Constraint h_1 defines the size of a subcluster, while h_2 and h_3 enforce the requirement that each function and variable belong to one and only one cluster, respectively. Constraint g_1 restricts the relative sizes of subclusters; g_2 states that a function belonging to the master cluster must be

depend only the linking variables, and g_3 precludes such functions from being in any subcluster. Constraint g_4 says that if a function j depending on a variable v is in subcluster i then variable v is also in i , unless v is a linking variable; g_5 says that if the functions depending on variable v are not in a subcluster i then v does not belong to subcluster i . Finally, g_6 says that each subcluster must have at least one design variable that is a local variable.

The ILP model in Eq. (6) is NP-hard. Cutting plane methods and enumeration methods such as branch and bound will guarantee a global optimum for the ILP model in Eq. (6), *if* a solution is found. Other methods relying mostly on heuristics can provide solutions at various levels of confidence (Papadimitriou and Steiglitz, op. cit., Murty 1994). Here we use a branch and bound approach. Specifically, the model is first represented using AMPL (A Mathematical Programming Language, see Fourer 1993), and then solved using standard software from OSL (Optimization System Library, see IBM 1990). Use of standard software is of particular influence in selecting the solution method.

SYNTHESIS WITH CONNECTED SUBCLUSTERS

As mentioned already, the model in Eq. (6) does not guarantee solutions that have all subclusters corresponding to connected components. Such connectedness may be a design requirement for the synthesis process. In the ILP formulation connectivity can be enforced by adding constraints. There is a difficulty, however, since the number of constraints needed to impose the connectivity requirement (keeping the formulation linear) increases exponentially with the number of design criteria and variables in the GDP. Imposing connectivity in a cluster requires that any partition into two parts of the graph representing the cluster must have at least one edge common to both partitions. For a cluster with n vertices there are $2^n - 1$ possible partitions, and this must be repeated for all vertices of all subclusters. In addition, new intermediate discrete variables would be needed to keep the formulation linear. This approach is considered unprofitable as we would have a very large number of constraints in the ILP model for even medium size problems.

An alternative is to use a two phase approach for solving the graph partitioning problem directly using a greedy recursive graph partitioning technique followed by local search. In the first phase we try to achieve a ‘good’ feasible solution and in the second to improve on it. One could stop at the end of the first phase, if desired. Following the process exemplified in Fig. (3), the main task is to identify the “best” set of linking variables. This identifies the master and subclusters while all constraints in Eq. (6), except for the relative size constraints g_1 , are automatically satisfied. The goal then is to identify the linking variables that provide the lowest objective value without violating the relative size constraint in Eq. (6) — without exhaustive enumeration.

Recursive Partitioning

The starting point for the partitioning process is the GDP with N variables and T functions. The graph is assumed to be connected and it is partitioned recursively. Each recursion is called a stage and has the following steps

Step 1: Select cluster for partitioning. At each stage the largest subcluster (the one with the maximum number of vertices) is identified. A tie is broken by choosing the cluster containing the variable with the lowest index. The first such subcluster is the original graph.

Step 2: Selecting the best linking variable. Partitioning is effected by removing from the graph the vertices that correspond to each subcluster variable, one at a time, to test if the particular variable could serve as a linking variable. Each variable tested from the list of candidate linking variables is temporarily added to the list of *best linking variables*, b_l . The master cluster and subclusters are identified, and the model, Eq. (6) is evaluated. After collecting model values for each variable in the current largest subcluster, the best linking variable (the one whose feasible solution has the lowest objective value) is added to b_l permanently. A tie is broken by choosing the variable with the lowest index. If no feasible solution is obtained, the best variable is chosen as the one with the least violation of the relative size constraint. Ties are broken by choosing the

variable whose vertex has the maximum degree in the graph corresponding to the subcluster under consideration.

Step 3: Updating. The list b_l is updated until all variables in the GDP have been chosen as linking variables or some other termination criterion is met. Each updated solution can be further improved by a local search technique; otherwise, the solution chosen is the best feasible solution (if it exists) available at termination.

To illustrate consider the example GDP

$$\begin{array}{ll}
 g_1(x_1, x_2, x_3) \leq 0 & g_2(x_1, x_4) \leq 0 \\
 g_3(x_1, x_2) \leq 0 & g_4(x_1, x_4) \leq 0 \\
 g_5(x_1, x_3) \leq 0 & g_6(x_3) \leq 0
 \end{array} \tag{7}$$

Assuming $K_s = 2$ the master cluster and the subclusters identified by Phase 1 is as follows:

$$\begin{array}{l}
 \text{Update 1: MC} - \{x_1\}; \text{SC1} - \{g_1, g_3, g_5, g_6; x_2, x_3\}, \text{SC2} - \{g_2, g_4, x_4\}: \\
 \quad \text{Objective value} = 5.5 \\
 \text{Update 2: MC} - \{x_1, x_2; g_3\}; \text{SC1} - \{g_2, g_4, x_4\}, \text{SC2} - \{g_1, g_5, g_6, x_3\}: \\
 \quad \text{Objective value} = 6.5 \\
 \text{Update 3: MC} - \{x_1, x_2, x_3; g_1, g_3, g_5, g_6\}; \text{SC1} - \{g_2, g_4, x_4\}: \\
 \quad \text{Objective value} = 10.0 \\
 \text{Update 4: MC} - \{x_1, x_2, x_3, x_4; g_1, g_2, g_3, g_4, g_5, g_6\}: \\
 \quad \text{Objective value} = 10.0
 \end{array} \tag{8}$$

The best solution is obtained when a single linking variable x_1 is chosen, terminating Phase 1 after all variables have been considered.

Approximate complexity analysis of Phase 1 when T is of the same order as N indicates a worst case in the order of N^4 operations, if all operations are done sequentially. Of course, Phase 1 can be also implemented using a connected component identification algorithm commonly used in the area of graph theory (see, e.g., Deo 1990).

Local Search

Local search techniques are based on single or multiple exchanges of nodes between clusters (Kernighan and Lin 1970). The number of linking variables is assumed *fixed* during the search. Two ordinary local search and one variable depth search methods are used here based on standard approaches. The methods mainly differ in how a local neighborhood is defined and searched (see, Papadimitriou and Steiglitz, and Murty, op. cit). At each stage the local search goes through cycles that consist of one or more node exchanges. In the ordinary local search only exchanges that lead to an improvement are accepted during a cycle, a restriction not used in the variable depth search.

In the present implementation, a feasible solution is always preferred over an infeasible one. In case of a tie for the best feasible solution, the partition that contains the exchanged variable of the lowest index is chosen. If all solutions are infeasible then the best solution is selected as the one with the least relative size constraint violation. The variable depth search is more expensive than ordinary local search but tends to lead to better solutions.

DEMONSTRATION EXAMPLES

Application to three well-known optimal design examples are presented in this section: a pressure vessel, a caliper disc brake, and a speed reducer. Since we start from a GDP, the models used here are not exactly the same as those in the literature. Synthesis of ODP's from the GDP's is performed using (i) the integer programming model Eq. (6) and its branch and bound solution (ILP) — with no subcluster connectivity requirement, (ii) recursive graph partitioning without local search (RP), with local search (RPLS), and by complete enumeration or exhaustive search (ES) — where all combinations of linking variables, and subclusters are considered. Results for the different solution techniques are tabulated for three different values of the relative size constraint parameter K_3 : 2, 1.5, and 1; It is assumed $w_m = w_s = 1.0$ throughout. The solution to the relaxed LP is also reported as it provides a global lower bound.

Design of a Pressure Vessel

The GDP is based on a model first given in Wilde (1978), and subsequently used by several authors, including Papalambros and Wilde (1988) and Sandgren (1990). The pressure vessel is made of a cylindrical body with hemispherical heads welded at the two ends. The design variables x_1, \dots, x_4 correspond to cylinder and head radius, cylinder thickness, cylinder length, and head thickness, respectively.

Table 1 Pressure vessel general design problem

GDP - Functional Representation	Description of the Design Criteria	FDT Matrix Representation			
		x_1	x_2	x_3	x_4
$g_1(x_1, x_2, x_3) \leq 0$	Cylinder mass limits	1	1	1	0
$g_2(x_1, x_4) \leq 0$	Hemisphere mass limits	1	0	0	1
$g_3(x_1, x_2) \leq 0$	Stress limits in cylinder walls	1	1	0	0
$g_4(x_1, x_4) \leq 0$	Stress limits in hemispherical walls	1	0	0	1
$g_5(x_1, x_3) \leq 0$	Volume requirement	1	0	1	0
$g_6(x_3) \leq 0$	Limit on cylinder length	0	0	1	0

Table 1 summarizes the functional dependence table (FDT) assumed here. Note that detailed knowledge of the exact functional form is not required.

The value of the objective function for the optimal synthesis model for different solution techniques and parameters are shown in Table 2. The numbers in parentheses indicate the optimal number of subclusters in each case. The best GDP structure identified for $K_s = 2$ is shown in Fig. 4. No functions exclusively depending on x_1 exist in this case. The sub-clusters obtained using the ILP formulation were connected components even though no such requirement was explicitly imposed. The objective function values and the best linking variables (not shown in the table) were the same for all cases except for $K_s = 1$ with RP. That solution has only one subcluster, because no feasible one with two or more subclusters could be found using only the RP technique. In contrast, two subclusters are identifiable when the RP technique is combined with any of the local search techniques (RPLS). The best block-angular structures for $K_s = 1.0$ and $K_s = 1.5$ are given in Eq. (9) and (10) below. The two sub-clusters SC1 and SC2

correspond to subproblems related to hemispherical head and cylinder body, respectively. This is the case for $K_s = 2$ as well.

$$K_s = 1.0, \text{ objective value} = 7.0: \quad \text{MC} - \{x_1, x_3; g_5, g_6\}; \quad (9)$$

$$\text{SC1} - \{g_2, g_4; x_4\}, \text{SC2} - \{g_1, g_3; x_2\};$$

$$K_s = 1.5, \text{ objective value} = 6.5: \quad \text{MC} - \{x_1, x_2; g_3\}; \quad (10)$$

$$\text{SC1} - \{g_2, g_4; x_4\}, \text{SC2} - \{g_1, g_5, g_6; x_3\};$$

Table 2 Pressure Vessel Problem Results

Solution Techniques	$K_s = 2$	$K_s = 1.5$	$K_s = 1.0$
RP	5.5 (2)	6.5 (2)	10.0 (1)
RPLS	5.5 (2)	6.5 (2)	7.0 (2)
ES	5.5 (2)	6.5 (2)	7.0 (2)
ILP (B & B)	5.5 (2)	6.5 (2)	7.0 (2)
LP	5.0 (2)	5.0 (2)	5.0 (2)

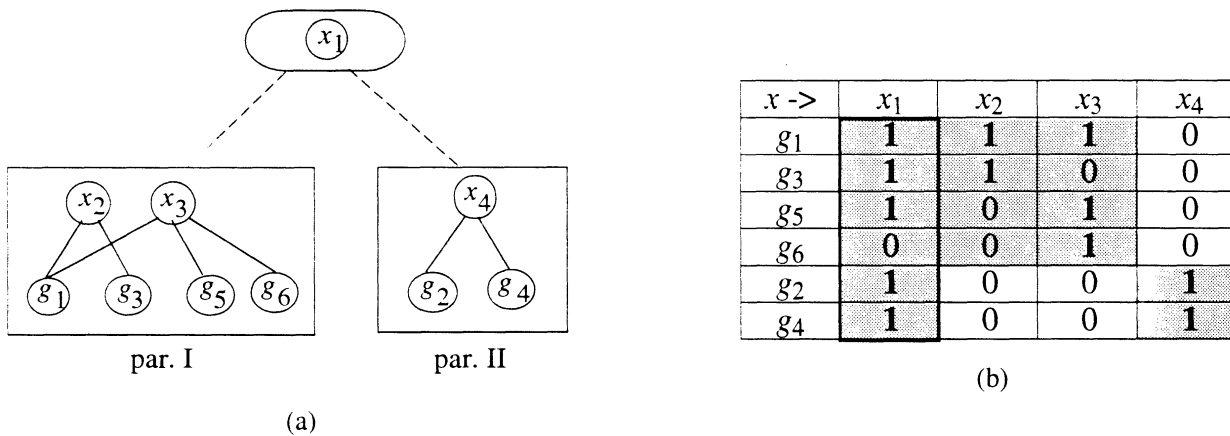


Fig. 4 Pressure vessel: optimal GDP partitioning for $K_s = 2$

To compose the ODP several ways are possible. For example, for $K_s = 2$, one can choose $\{g_1, g_2\}$ as the objective of each subcluster and compose an additive separable objective as discussed in K&P (1995). A hierarchically decomposed ODP can be similarly synthesized from Eq. (9, 10) for the cases of $K_s = 1, K_s = 1.5$.

Design of a Caliper Disk Brake

A disc brake for a passenger car based on the model proposed by Siddall (1982) is considered here. The brake has a caliper that holds a hydraulic cylinder and piston assembly. The GDP is shown in Table 3. The variables x_1, x_2, \dots, x_6 correspond to lining center line radius, lining diameter, piston diameter, disc thickness, oil pressure, and outside disk diameter, respectively. Optimal solutions are shown in Table 4. Again the ILP formulation generated connected components. The objective function value increases as the K_s value decreases, since the problem is being more severely restricted.

Table 3 General design problem for designing a caliper disc brake

	Design Criteria
$g_1(x_3, x_5, x_1, x_2) \leq 0$	limit on stopping time
$g_2(x_3, x_5, x_1, x_2) \leq 0$	limit on the braking force
$g_3(x_6, x_4) \leq 0$	volume of the disk
$g_4(x_5) \leq 0$	oil pressure limits
$g_5(x_3, x_5, x_1, x_2) \leq 0$	lining pressure limit
$g_6(x_6, x_4) \leq 0$	limit on maximum temperature
$g_7(x_6, x_1, x_2) \leq 0$	lining must not overhang disc
$g_8(x_1, x_2) \leq 0$	lining must not interfere with the hub
$g_9(x_1, x_3) \leq 0$	cylinder must not interfere with the hub

The best block-angular structure in the GDP is shown in Fig. 5 and described as follows:

$K_s = 2$ and 1.5, objective value = 9:

$$MC - \{x_1, x_2; g_8\}; \quad (11)$$

$$SC1 - \{g_3, g_6, g_7; x_4, x_6\}, \quad SC2 - \{g_1, g_2, g_4, g_5, g_9; x_3, x_5\};$$

Note that the criteria in SC1 relate to disc design, and in SC2 to the requirements for stopping the vehicle. From the clusters identified one could choose g_1 (stopping time) and g_3 (volume of the disc) and compose an additively separable objective in the ODP. Criterion g_8 that limits how close the lining can get to the hub may also be added to the objective with a suitable weight, if this is important for packaging.

Table 4 Caliper Disc Brake Problem Results

Solution Method	$K_s = 2$	$K_s = 1.5$	$K_s = 1.0$
All Techniques	9 (2)	9 (2)	10 (2)
LP	7.5 (2)	7.5 (2)	7.5 (2)

x->	x_1	x_2	x_3	x_5	x_4	x_6
g_8	1	1	0	0	0	0
g_1	1	1	1	1	0	0
g_2	1	1	1	1	0	0
g_4	0	0	0	1	0	0
g_5	1	1	1	1	0	0
g_9	1	0	1	0	0	0
g_3	0	0	0	0	1	1
g_6	0	0	0	0	1	1
g_7	1	1	0	0	0	1

Fig. 5 Optimal block-angular structure in the GDP

For $K_s = 1$ the solution is as follows:

$$K_s = 1, \text{ objective value} = 10: \quad \text{MC} - \{x_1, x_2, x_3; g_8, g_9\}; \quad (12)$$

$$\text{SC1} - \{g_3, g_6, g_7; x_4, x_6\}, \text{SC2} - \{g_1, g_2, g_4, g_5; x_5\};$$

The structure resembles closely those with the other values of K_s .

In all cases then the engineering interpretation for a meaningful ODP is that the assembly or packaging requirements are specified by the master problem and two subproblems deal with design requirements for the disc and for stopping the vehicle.

Design of a Speed Reducer

Design of a speed reducer based on a model originally proposed by Golinski (1970) and later modified by Lee (1977) and Azarm et al. (1989) is considered here. The reducer consists of a gear-pinion pair mounted on shafts 1 and 2 respectively. Each shaft is supported by one bearing at each end. The system includes gear, pinion, shafts, and bearings enclosed in a housing.

Table 5 General design problem for speed reducer

	Design Criteria
$g_1(x_1, x_2, x_3) \leq 0$	bending stress on the gear tooth
$g_2(x_1, x_2, x_3) \leq 0$	compressive stress on the gear tooth
$g_3(x_1, x_2, x_3, x_6) \leq 0$	weight of the pinion
$g_4(x_1, x_2, x_3, x_7) \leq 0$	weight of the gear
$g_5(x_2, x_3, x_4, x_6) \leq 0$	deflection of shaft 1
$g_6(x_2, x_3, x_5, x_7) \leq 0$	deflection of shaft 2
$g_7(x_2, x_3, x_4, x_6) \leq 0$	stress in shaft 1
$g_8(x_2, x_3, x_5, x_7) \leq 0$	stress in shaft 2
$g_9(x_6, x_4) \leq 0$	limit on the diameter of the shaft 1
$g_{10}(x_7, x_5) \leq 0$	limit on the diameter of the shaft 2
$g_{11}(x_4, x_6) \leq 0$	weight of shaft 1
$g_{12}(x_5, x_7) \leq 0$	weight of shaft 2
$g_{13}(x_2, x_3) \leq 0$	sum of the diameters of the gear and pinion
$g_{14}(x_1, x_2) \leq 0$	upper bound on face width
$g_{15}(x_1, x_2) \leq 0$	lower bound on face width
$g_{16}(x_1) \leq 0; g_{17}(x_1) \leq 0;$	bounds
$g_{18}(x_2) \leq 0; g_{19}(x_2) \leq 0;$	bounds
$g_{20}(x_3) \leq 0; g_{21}(x_3) \leq 0;$	bounds
$g_{22}(x_4) \leq 0; g_{23}(x_4) \leq 0;$	bounds
$g_{24}(x_5) \leq 0; g_{25}(x_5) \leq 0;$	bounds
$g_{26}(x_6) \leq 0; g_{27}(x_6) \leq 0;$	bounds
$g_{28}(x_7) \leq 0; g_{29}(x_7) \leq 0;$	bounds

The GDP shown in Table 5 follows the model version by Lee (1977). The variables x_1, x_2, \dots, x_7 are gear face width, gear tooth module, number of teeth on the pinion, distance between gear shaft bearings, distance between pinion shaft bearings, and shaft 1 and 2 diameters.

Table 6 Speed Reducer Problem Results

Solution Techniques	$K_s = 2$	$K_s = 1.5$	$K_s = 1.0$
RP	25.0 (2)	25.0 (2)	25.0 (2)
RPLS	25.0 (2)	25.0 (2)	25.0 (2)
ES	20.66 (3)	20.66 (3)	25.0 (2)
ILP (B & B)	20.66 (3)	20.66 (3)	25.0 (2)
LP	12.0 (3)	12.0 (3)	18.0 (2)

Partitioning results are shown in Table 6. For $K_s = 2$ the best structure obtained by RP and RPLS are shown in Fig. 6(a) and by ES and ILP in Fig. 6(b).

	x_1	x_2	x_3	x_5	x_7	x_4	x_6
g_1	1	1	1	0	0	0	0
g_2	1	1	1	0	0	0	0
g_{13}	0	1	1	0	0	0	0
g_{14}	1	1	0	0	0	0	0
g_{15}	1	1	0	0	0	0	0
g_{16}	1	0	0	0	0	0	0
g_{17}	1	0	0	0	0	0	0
g_{18}	0	1	0	0	0	0	0
g_{19}	0	1	0	0	0	0	0
g_{20}	0	0	1	0	0	0	0
g_{21}	0	0	1	0	0	0	0
g_4	1	1	1	0	1	0	0
g_6	0	1	1	1	1	0	0
g_8	0	1	1	1	1	0	0
g_{10}	0	0	0	1	1	0	0
g_{12}	0	0	0	1	1	0	0
g_{24}	0	0	0	1	0	0	0
g_{25}	0	0	0	1	0	0	0
g_{28}	0	0	0	0	1	0	0
g_{29}	0	0	0	0	1	0	0
g_3	1	1	1	0	0	0	1
g_5	0	1	1	0	0	1	1
g_7	0	1	1	0	0	1	1
g_9	0	0	0	0	0	1	1
g_{11}	0	0	0	0	0	1	1
g_{22}	0	0	0	0	0	1	0
g_{23}	0	0	0	0	0	1	0
g_{26}	0	0	0	0	0	0	1
g_{27}	0	0	0	0	0	0	1

(a)

	x_2	x_3	x_6	x_7	x_4	x_1	x_5
g_{13}	1	1	0	0	0	0	0
g_{18}	1	0	0	0	0	0	0
g_{19}	1	0	0	0	0	0	0
g_{20}	0	1	0	0	0	0	0
g_{21}	0	1	0	0	0	0	0
g_{26}	0	0	1	0	0	0	0
g_{27}	0	0	1	0	0	0	0
g_{28}	0	0	0	1	0	0	0
g_{29}	0	0	0	1	0	0	0
g_5	1	1	1	0	1	0	0
g_7	1	1	1	0	1	0	0
g_9	0	0	1	0	1	0	0
g_{10}	0	0	0	1	0	0	1
g_{11}	0	0	1	0	1	0	0
g_{22}	0	0	0	0	1	0	0
g_{23}	0	0	0	0	1	0	0
g_1	1	1	0	0	0	1	0
g_2	1	1	0	0	0	1	0
g_3	1	1	1	0	0	1	0
g_4	1	1	0	1	0	1	0
g_{14}	1	0	0	0	0	1	0
g_{15}	1	0	0	0	0	1	0
g_{16}	0	0	0	0	0	1	0
g_{17}	0	0	0	0	0	1	0
g_6	1	1	0	1	0	0	1
g_8	1	1	0	1	0	0	1
g_{12}	0	0	0	1	0	0	1
g_{24}	0	0	0	0	0	0	1
g_{25}	0	0	0	0	0	0	1

(b)

Fig. 6 Optimal block-angular structure in the speed reducer GDP: (a) RP result (b) ES result

The subclusters obtained using ILP were again connected components. No feasible solution could be found for a number of subclusters higher than shown in Table 6.

The block-angular structure in Fig. 6(a) represents the partition

$$K_S = 2, \text{ objective value} = 25.0: \text{MC} - \{x_1, x_2, x_3; g_1, g_2, g_{13}, \dots, g_{21}\}; \quad (13)$$

$$\text{SC1} - \{g_3, g_5, g_7, g_9, g_{11}, g_{22}, g_{23}, g_{26}, g_{27}; x_4, x_6\},$$

$$\text{SC2} - \{g_4, g_6, g_8, g_{10}, g_{12}, g_{24}, g_{25}, g_{28}, g_{29}; x_5, x_7\};$$

One may choose g_3 (pinion weight), g_4 (gear weight), and g_{13} (gear box width) to compose the objective. The criteria in SC1 and SC2 relate to pinion and sub-assemblies, respectively. Since

it is not necessary to pick only one criterion from each cluster, one could compose an additively separable objective using $\{g_3, g_{11}\}$ from SC1, $\{g_4, g_{12}\}$ from SC2, and g_{13} from MC.

The structure in Fig. 6(b) obtained using ILP represents the partition:

$$\begin{aligned}
K_s = 2, \text{ objective value} = 20.66: \quad & \text{MC} - \{x_2, x_3, x_6, x_7; g_{13}, g_{18}, \dots, g_{21}, g_{26}, \dots, g_{29}\}; \\
& \text{SC1} - \{g_5, g_7, g_9, g_{11}, g_{22}, g_{23}; x_4\}, \\
& \text{SC2} - \{g_1, \dots, g_4, g_{14}, \dots, g_{17}; x_1\}, \\
& \text{SC3} - \{g_6, g_8, g_{10}, g_{12}, g_{24}, g_{25}; x_5\};
\end{aligned} \tag{14}$$

Subclusters SC1, SC2, and SC3 correspond to design of shafts 1, the gear-pinion pair, and shaft 2, respectively. A meaningful additively separable objective can be composed choosing g_{11} from SC1, g_{12} from SC3, $\{g_3, g_4\}$ from SC2, and g_{13} from MC. This is the same objective as one of those constructed from Eq. (13) above, but the decomposed ODP has one additional subproblem. The ILP branch and bound method found a better solution than RP or RPLS, but a meaningful synthesis of a decomposed ODP can be achieved with all methods.

From the results obtained so far one might wonder whether the ILP formulation does result in subproblem clusters that are not connected. That is indeed the case. The ILP solution obtained when two subclusters were sought with $K_s = 2$ is

$$\begin{aligned}
K_s = 2, \text{ objective value} = 24.5: \quad & \text{MC} - \{x_2, x_3, x_6, x_7; g_{13}, g_{18}, \dots, g_{21}, g_{26}, \dots, g_{29}\}; \\
& \text{SC1} - \{g_5, \dots, g_{12}, g_{22}, \dots, g_{25}; x_4, x_5\}, \\
& \text{SC2} - \{g_1, \dots, g_4, g_{14}, \dots, g_{17}; x_1\};
\end{aligned} \tag{15}$$

The graph representing SC1 consists of two connected components. As expected, the objective value in this case is lower than the best solution obtained (with two subclusters) when the subclusters were required to be connected (see Table 6).

DISCUSSION

Given the combinatorial nature of the optimal synthesis formulation, the techniques presented seek the best possible solution with a reasonable amount of computation. The nature of the clustering procedure requirements makes it difficult to use partitioning methods, such as those in

VLSI circuit design (Bryan et al. 1988). Seeking a "good" rather than an optimal solution is an appropriate goal. Indeed, the optimization model in Eq. (6) has limitations and subjective judgment is still needed. In this context, the following discussion points are in order.

Synthesis of a decomposed ODP for a large system would in general require evaluation of the optimally decomposed ODPs generated for different values of K_s and of the weights w_m, w_s . Multicriteria optimization may prove useful, particularly if the designer does not have a good grasp of how to rank relative importance of the criteria. Weights could be assigned to different criteria and/or variables in situations where the design functions have to be evaluated using extensive simulations. Extensions to the present formulation and solution techniques will be required.

The model presented cannot account for changes in the size of the clusters when composing the objective. Also, the best structure identified for synthesis from a given GDP is based on the associated functional dependence table (FDT) representing the system. The FDT is not unique, as manipulations and/or functional rearrangements of the GDP model can modify the FDT. The procedure is then sensitive to changes in the FDT. However, several FDT's for the same design problem could be "averaged" to a single one using weights. The FDT need be stable rather than precise, since the synthesis process ultimately aims at organizing the design activity rather than at a computational advantage on the margin.

In solving the ILP the number of subclusters K must be specified. The maximum value of K for which a feasible solution to the ILP exists is not known a priori. If the continuous LP is infeasible for a particular value of K then the associated ILP is infeasible. The Branch and Bound method used for solving the ILP can also discover infeasibility but the number of nodes in the branch and bound tree could become large. A good starting guess for the maximum value of K can be easier to obtain using the RP or RPLS techniques. The branch and bound solution provides a lower bound for RP and RPLS, and the continuous LP solution provides a lower bound for the ILP.

Finally, the integer programming formulation may not be always solved using branch and bound in a reasonable time. Branch and cut methods (Papadimitriou and Steiglitz 1982) may be

proven more effective in larger problems. A discussion of alternate methods that may not guarantee a global optimum is available in Murty (1994).

CONCLUSION

A formulation for optimal decomposition synthesis and some general solution techniques have been presented. The designer can use the proposed methods to synthesize a suitable decomposed ODP. The integer programming formulation can be readily adapted to different user needs.

Only two level primal decomposition synthesis has been discussed in this paper. In many problems it may be preferable to obtain a multi-level decomposed ODP instead of a two-level one or to use dual methods instead of primal ones. Developing formulations and solution techniques to address these requires further study.

The ideas proposed here have been tested using small to medium size design problems. Their true value will be ascertained only after treating problems of a size sufficiently large to defy obvious intuitive problem partitions, and to test their relative merits. Coordinated solution of the decomposed problems will pose its own challenges.

ACKNOWLEDGMENT

This research has been partially supported by the Automotive Research Center at the University of Michigan, a US Army Center of Excellence in Modeling and Simulation of Ground Vehicles, under Contract No. DAAE07-94-C-R094. This support is gratefully acknowledged. The second author is also grateful to the Universities of Michigan and Patras for supporting his sabbatical leave for this research.

REFERENCES

Azarm, S., and Li, W., 1989, "Multi-level Design Optimization Using Global Monotonicity Analysis," *ASME Journal of Mechanisms, Transmission, and Automation in Design*, Vol. 113, No. 111, pp. 211-224.

- Bryan, T., and Lorenzetti, M., (eds.), 1988, *Physical Design Automation of VLSI Systems*, Benjamin/Cummings, San Francisco.
- Deo, N., 1990, *Graph Theory With Applications to Engineering and Computer Science*, Prentice-Hall of India, New Delhi.
- Fourer, R., Gay, D., and Kernighan, B., 1993, *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, South San Francisco.
- Golinski, J., 1970, "Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods," *Journal of Mechanisms*, Vol. 5, pp. 287-309.
- IBM, 1990, "Optimization Subroutine Library Guide and Reference," IBM Corporation, N.Y.
- Kernighan, B., and Lin, S., 1970, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell Systems Journal*, Vol. 29, pp. 291-307.
- Krishnamachari, R., and Papalambros, P., 1995, "Hierarchical Decomposition Synthesis in Optimal Systems Design," *ASME Journal of Mechanical Design*, (in review).
- Lee, T., 1977, "Weight Minimization of a Speed Reducer," *ASME Paper No. 77-DET 163*.
- Michelena, N., and Papalambros, P., 1994. "A Network Reliability Approach to Optimal Decomposition of Design Problems," *ASME Journal of Mechanical Design*, Vol. 117, No. 3, pp. 433-440.
- Michelena, N., and Papalambros, P., 1995. "Optimal Model-Based Partitioning of Powertrain System Design," S. Azarm et al. (eds.), *Advances in Design Automation*, Boston, Vol. 1, pp. 165-192.
- Murty, K., 1994, *Operations Research: Deterministic Models*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Papadimitriou, C., and Steiglitz, K., 1982, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Papalambros, P., 1995, "Optimal Design of Mechanical Components and Systems," *ASME Journal of Mechanical Design*, 50th Anniversary of the Design Engineering Division Issue, Vol. 117, pp. 55-62.
- Papalambros, P., and Wilde, D., 1988, *Principles of Optimal Design: Modeling and Computation*, Cambridge U. Press, New York.

- Sandgren, E., 1990, "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *ASME Journal of Mechanical Design*, Vol. 112, pp. 223-229.
- Siddall, 1982, *Optimal Engineering Design*, Marcel Dekker, Inc., New York.
- Wagner, T., 1993, *A General Decomposition Methodology for Optimal Systems Design*, Doctoral Dissertation, Department of Mechanical Engineering and Applied Mechanics, The University of Michigan, Ann Arbor, MI.
- Wilde, D., 1978, *Globally Optimal Design*, Wiley, New York.