AIAA JOURNAL
Vol. 41, No. 6, June 2003

System Design Through Subsystem Selection Using Physical Programming

Mitul Patel* and Kemper E. Lewis†
State University of New York at Buffalo, Buffalo, New York 14206
and
Aniela Maria‡ and Achille Messac§
Rensselaer Polytechnic Institute, Troy, New York 12180-3590

The design of complex systems can involve the selection of several subsystem designs. We investigate the problem of selecting discrete concepts from multiple, coupled subsystems. This problem is one where measures of merit for both subsystem (local) and system (global) levels are present. An approach is developed to obtain the sets of preferred subsystem design concepts. Graph theory is used to represent the coupled selection problem where the nodes of the graph are the subsystem design choices and the arcs connecting the nodes indicate the relationships between the subsystems. Optimization techniques from graph theory and physical programming are combined to form an approach to model and solve this problem. This approach can be used to identify a given number of successful, or feasible, subsystem combinations that represent design alternatives. Once the promising subsystem designs are obtained at the conceptual design stage, focus can be restricted to these chosen design alternatives for further testing and refinement at a later embodiment design stage. Although the examples presented in this paper involve conceptual design, the presented approach can be used with any coupled discrete selection problem.

Nomenclature

A_f	=	wetted area of fuselage, m^2
A_R	=	wing aspect ratio
b	=	wing span, m
C	=	specific fuel consumption of engine, kg/Ns
C_L	=	wing lift coefficient
$(L/D)_C$	=	actual cruise lift-to-drag ratio
S_{ref}	=	reference wing area, m^2
T_{RC}	=	engine cruise thrust, kg
T_{RTO}	=	engine takeoff thrust, kg
W_e	=	dry weight of engine, kg
W_f	=	weight of fuselage, kg
W_p	=	weight of payload, kg
W_w	=	weight of wing, kg

I. Introduction

THE design of complex systems can involve the selection of several subsystem designs, each of which might involve complex analysis routines. It is critically important that the subsystem designs selected be mutually compatible in order for the overall system to function efficiently. Considerable literature exists in the area of concept selection at various stages of the design process. In the later stages of design, assemblies and modules of a design configuration must be selected in order to meet functional and geometrical interface constraints. In this class of combinatorial selection problems, the number of possible solutions can only be estimated using approximations, as the number of possibilities is too large to enumerate explicitly.[1]

In the earlier stages of design, the combinatorial selection problem is one of choosing one of the subsystem concepts that meets functional and performance constraints. Ulrich and Eppinger[2] use the Pugh selection method in comparing designs at the concept screening stage with the help of decision matrices. At the concept scoring stage the designer weighs the various concepts based on the customer preferences. The concept scores are determined by the weighted sum of the rating. Hazelrigg[3] also details the use of utility theory and von Neumann–Morgenstern lotteries to make a choice among a given number of design options. Yang and Sen[4] use a hierarchical evaluation process for multiattribute design selection with uncertainty. They use hierarchical factor structure for evaluation and quantification of a qualitative attribute. See and Lewis[5] use a set of hypothetical alternatives to elicit a set of stated preferences from the decision maker in order to make a multiattribute selection decision. However, these and most other methods, although able to address multiattribute selection problems, do not address the selection of a design from multiple coupled subsystems that have conflicting objectives. In this paper coupled subsystems refer to a set of subsystems whose selected design depends on those of other subsystems. In other words, the subsystems are not independent and cannot be analyzed and solved in isolation.

Also, many methods identify a single best concept. On the other hand, it can be argued that at the conceptual design stage identifying a manageably small set of good designs could prove to be a more promising product development strategy. Therefore, we present an approach to identify a number of good subsystem design combinations, based on local and global measures of merit. This approach can be used for any discrete coupled selection problem, whether it is the selection of subsystems within a given design or the selection of specific geometrical modules and assemblies within a subsystem. In either case the problem is a discrete coupled selection problem that requires 1) modeling compatibility constraints, 2) local and global measures of merit, and 3) an efficient solution approach.

The discrete subsystem selection problem under consideration is conceptually represented in Fig. 1, where each of the three subsystems (the columns) has three different alternatives. Figure 1 represents a simple example, but the discrete nature of such problems also makes them explode computationally because the number of subsystem combinations increases combinatorially with the number

Received 30 January 2001; revision received 30 November 2002; accepted for publication 20 January 2003.

*Research Assistant, Department of Mechanical and Aerospace Engineering. Student Member AIAA.

†Associate Professor, Department of Mechanical and Aerospace Engineering; kelewis@eng.buffalo.edu. Member AIAA.

‡Research Assistant, Mechanical, Aerospace, and Nuclear Engineering Department. Student Member AIAA.

§Associate Professor, Mechanical, Aerospace, and Nuclear Engineering Department, JEC-2049; messac@rpi.edu. Associate Fellow AIAA.

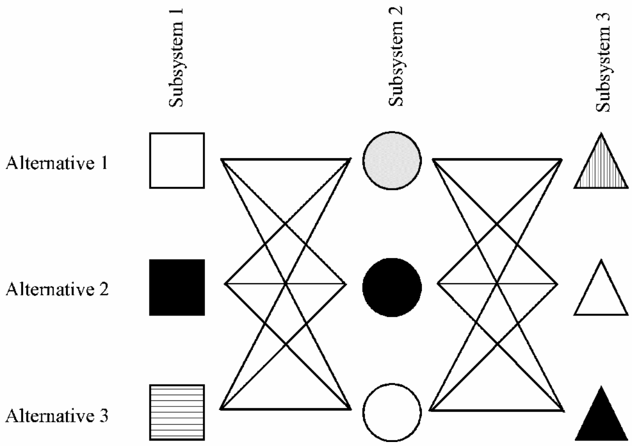


Fig. 1 Discrete design selection problem.

of subsystems and the number of concept alternatives per subsystem. For example, in the simple example of Fig. 1, the total number of combinations is $3 \times 3 \times 3 = 27$.

The important and difficult task of the system designer is to identify the combinations of subsystem designs that satisfy the system objectives. The coupled selection problem of Fig. 1 can be represented by the generic 0–1 selection optimization problem, which is given in Eq. (1):

$$\min F(x) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} w_{ij} x_{ij} \tag{1a}$$

subject to:

$$c_i(x_{ij}) \leq b_i \tag{1b}$$

$$x_{ij} = 0 \quad \text{or} \quad 1 \tag{1c}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} = n \tag{1d}$$

where n is the number of subsystems, m_i is the number of alternatives for subsystem i, x_{ij} are the design variables (one if subsystem is selected, zero if subsystem is not selected), and w_{ij} are the weights associated with the design variables (value of choosing that subsystem).

Concepts from the physical programming method developed by Messac[6] and graph theory form the basis for an algorithm that addresses the discrete coupled selection problem and are used in this paper. Physical programming is used to model the designer's preferences for both the local measures of merit (characteristic of a given subsystem) and the global measures of merit (characteristic of the entire system). Because the design variables are limited to either 0 or 1, a discrete programming algorithm, called the kth shortest path algorithm, is used with physical programming to find the best subsystem design combinations.

In this paper we focus on three aspects of this problem: 1) determining how to integrate and balance local and global measures of merit, 2) handling the discrete nature of the problem, and 3) handling the multidisciplinary nature of the problem, as each subsystem's problem is dependent on the choices made on other subsystems—from different disciplines.

The integration of the local and global measures of merit is a difficult task. In particular, we address the question, Does a subsystem design need to meet subsystem objectives, or system objectives, or both? Physical programming is used to model both sets of objectives. In previous work[7] a modified form of Taguchi's quality loss function is integrated with graph theory to model the sets of objectives. However, physical programming has been shown to capture a designer's preference accurately and is strongly practical in a multidisciplinary design environment.[8–10]

Some assumptions about the complex systems under consideration are the following:

1) The system consists of several subsystems, where each offers the possibility of different discrete designs from which to choose. The overall design problem is decomposed into representative subsystem problems.

2) The selection is made from off-the-shelf subsystem designs that are available. Therefore, the capability of the overall system depends on the capabilities of the individual subsystems.

3) The physics, or performance, of the subsystem is known, which allows for establishing compatibility issues between subsystem designs. Compatibility implies that all constraints between the various subsystem designs are clearly known.

4) The subsystems characteristics are measurable, allowing the designs to be quantitatively compared to make selections.

5) The system-level problem is governed by information provided from the subsystem-level design problem.

II. Technology Base

In this section we explain the primary technology base for this paper: graph theory and physical programming.

A. Graph Theory

We represent a coupled selection problem as in Fig. 1 as a network problem and solve it using graph theoretic techniques. A graph, or network, is defined by two sets of symbols: nodes and arcs.[11] Consider the simple network of Fig. 2. The nodes are represented as circles, arcs are represented as lines connecting nodes, and the arrows represent the direction of motion. Each arc has a weight (or length) associated with it, which is a measure of nodal attribute. For example, consider the nodes in Fig. 2 to be cities on a map. Each arc represents the distance between two cities, following a certain path. The problem of finding the shortest path from node A to any other node is known as the shortest path problem.

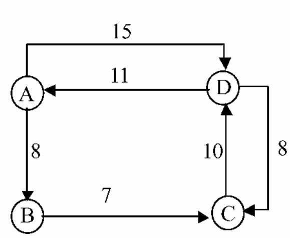
A typical discrete design selection problem might involve the integration of several subsystems, each having a number of design alternatives. To represent such a situation as a graph, we can consider the nodes to be subsystem alternatives and the arcs between any pair of nodes to be the compatibility relationship between them. The weights of each arc could describe the strength of the compatibility relationship between the nodes (subsystem alternatives) or the performance characteristic of the node.

In the present work we identify the candidate subsystem combinations using the so-called kth shortest path, which we describe as follows. Consider cities A–D in Fig. 2. Assume that the shortest path to reach D from A has been broken and that we must find the next shortest path. Such a problem is known as kth shortest path problem, where the k shortest paths are found. In this paper we use Dijkstra's kth shortest path algorithm[12] to obtain the shortest path between two nodes on a graph. The arc weights are obtained by representing the characteristics of the subsystem designs (nodes) using physical programming, a synopsis of which is provided in the next section. Arc weights represent the local measures of merit defined by the subsystem designers. More explanation about this method is presented in Sec. III. The local and global measures of merit are modeled using physical programming, which is a method based on modeling a designer's preferences.

B. Physical Programming

Designers innately have certain preference ranges for their design metrics. For example, an aircraft designer might have the following

Fig. 2 Graph G: a graph or network.



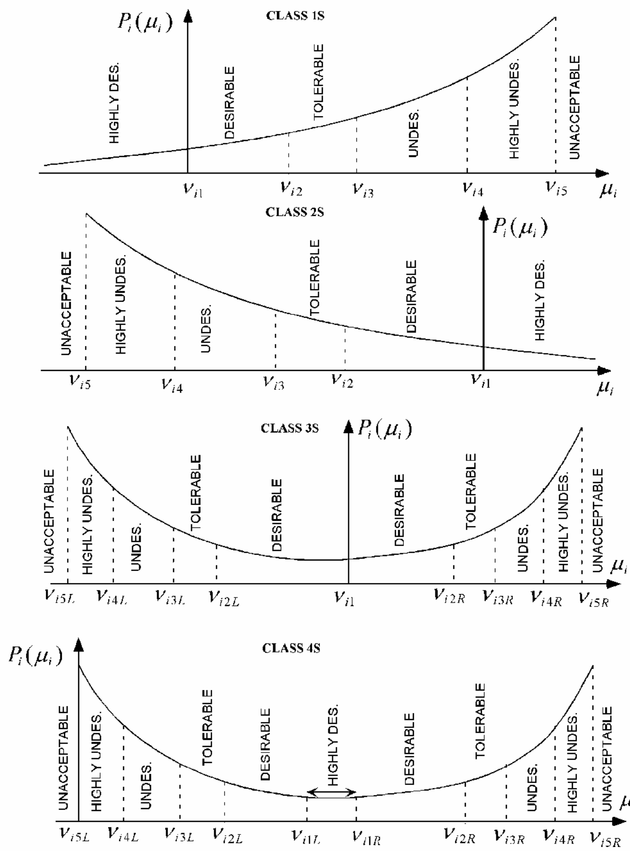


Fig. 3 Soft class functions for physical programming.

preferences for the range of an aircraft: 10,000 km or more is highly desirable, between 8000 and 10,000 is desirable, below 5000 km is not acceptable. Physical programming helps designers express their preferences with respect to each of their metrics or criteria using four different classes. Reference 6 provides detailed information regarding physical programming. We provide the following brief synopsis. The subsystem and system measures of merit will be appropriately defined using the four soft and hard classes of nonlinear physical programming, shown in Fig. 3, which is given as follows: the soft classes are 1S, smaller-is-better; 2S, larger-is-better; 3S, value-is-better; and 4S, range-is-better. The hard classes are 1H, must be smaller, $\mu_i \le \mu_{i,\max}$; 2H, must be larger, $\mu_i \ge \mu_{i,\min}$; 3H, must be equal, $\mu_i = \mu_{i,\max}$; and 4H, must be in range, $\mu_{i,\min} \le \mu_i \le \mu_{i,\max}$. The variable μ_i is the ith generic criterion expressed in terms of the problem's decision variable. In Fig. 3 the value of the criterion μ_i is on the horizontal axis, and the physical programming preference function value P_i for the pertaining criterion is on the vertical axis. Consider the first segment of the curve for class 1S in Fig. 3. We observe that when the value of the criterion μ_i is less than ν_{i1} (highly desirable range) the value of the preference function is small and requires little further minimization. On the other hand, when the value of the criterion ν_i, is between ν_{i4} and ν_{i5} (highly undesirable range) the value of the preference function is large, and significant minimization is necessary. Stated simply, the value of the preference function for each design metric governs the optimization path in objective space.

The preference function value for each of the system and subsystem objectives is determined using the four classes. For each of the design measures of merit (local or global), regions describing their degree of desirability (i.e., highly desirable, desirable, tolerable, undesirable, highly undesirable, and unacceptable) are defined. These ranges are defined by the individual designers and reflect his/her preferences.

As an illustration, the ranges for class 1S can be defined as follows:

1) The highly desirable range ($\mu_i \le \nu_{i1}$; range-1) is an acceptable range over which the improvement that results from further reduction of the performance objective is desired but is of minimal additional value.

2) The desirable range ($\nu_{i1} \le \mu_i \le \nu_{i2}$; range-2) is a range that is desirable and acceptable to the designer.

3) The tolerable range ($\nu_{i2} \le \mu_i \le \nu_{i3}$; range-3) is a range that is tolerable and acceptable to the designer.

4) The undesirable range ($\nu_{i3} \le \mu_i \le \nu_{i4}$; range-4) is a range that is undesirable but is still acceptable to the designer.

5) The highly undesirable range ($\nu_{i4} \le \mu_i \le \nu_{i5}$; range-5) is a range that is highly undesirable but is acceptable to the designer.

6) The unacceptable range ($\mu_i \ge \nu_{i5}$; range-6) is a range beyond ν_{i5} that is unacceptable to the designer.

In the preceding definitions the values ν_{i1} through ν_{i5} are physically meaningful values of subsystem designs or system requirements that are to be specified by the respective designers. For example, an aircraft designer might define his preferences for the range of the aircraft using class 2S with $R \ge 10{,}000$ km as the highly desirable range on the physical programming lexicon scale; $8000 \le R \le 10{,}000$ as a desirable range; $6000 \le R \le 8000$ as a tolerable range; and so on. The next section presents the integration of graph theory and physical programming to model the subsystem selection problem.

III. Subsystem Selection Problem Setup

The subsystems that constitute a complex system have several properties that affect the performance of the system. Consider, for example, the design of an automotive system in which a radiator and an engine are the two subsystems. Such a complex system can be modeled as a network and is shown in Fig. 4. The subsystem designs are the nodes of the graph, and the relationships between the designs are the arcs of the graphs. The arc weights reflect the properties of the subsystems. These weights are developed using the physical programming preference functions. Hence, the lower the weight, the better it is. The initial weights on the arcs of the graph are developed based on the subsystem-level objective functions. Thus, the initial weights represent the local measures of merit. The weights are applied to the arcs that connect a particular node to those of the preceding subsystem. From Fig. 4 the weight associated with node E_1 in subsystem 2 is applied to the arc connecting that node to any node in subsystem 1 and is denoted by w_{e1} and w_{e2}. Because the local measure of merit for a given node will remain the same irrespective of the node to which it is connected, the arcs connecting $R_1 - E_1$ and $R_2 - E_1$ will have the same weight (i.e., $w_{e1} = w_{e2}$).

For this example we assume that the radiator designer has two radiators to choose from, R_1 and R_2, and that the engine designer has two engines to choose from, E_1 and E_2. We further establish that

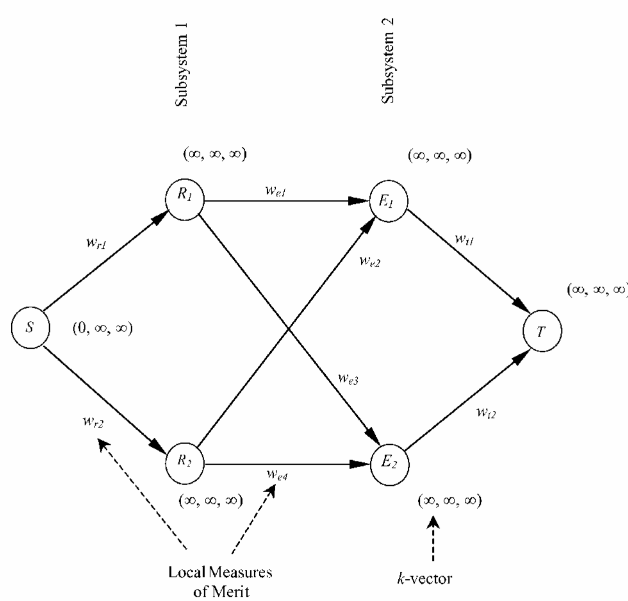


Fig. 4 Subsystem selection problem setup.

the radiator with higher efficiency and the engine with higher thrust are ranked higher by the radiator and engine designer, respectively. The objective of the system designer is to select an engine–radiator combination that occupies the least volume under the hood. The network model for the problem is shown in Fig. 4.

The model is a graph in which the set of nodes is given as

$$V \text{ (nodes)} = \{R_1, R_2, E_1, E_2\}$$

and the set of arcs weights is denoted as

$$W \text{ (weight)} = \{w_{r1}, w_{r2}, w_{e1}, w_{e2}, w_{e3}, w_{e4}, w_{t1}, w_{t2}\}$$

In Fig. 4 the directed arcs (arcs with arrows) connect the elements of the first set of subsystems R_i $(i = 1, 2)$ to the elements of the second set of subsystems E_i $(i = 1, 2)$ and not within the individual subsystem elements themselves. This connection arrangement occurs because of the absence of a relationship between alternatives within each subsystem.

If the graph were undirected (having arcs without arrows), it would mean unrestricted motion in any direction along the arcs. In such a case more than one subsystem alternative could be selected, which does not apply here. Thus, representing a network as a directed graph is appropriate when only one node from each node set is to be chosen.

The nodes S and T, respectively, represent the start and the terminal nodes of the graph and are not subsystem designs but are dummy nodes. These are used to show that the iteration of the graph problem begins at S and ends at T with the motion of control along the directed arcs. Also, the arcs pointing into the terminal node of the graph T do have dummy null weights. This is because the node T is a representation of the end of the graph and not a subsystem design.

Initially, we assume that the radiator designer rates his/her designs based on efficiency and that the resulting preference function value attributed to their designs R_1 and R_2 are 0.5 and 0.7, respectively. (These are example preference function values and do not represent any actual study.) We also assume that the engine designer specifies preference ranges that result in preference function values of 0.8 and 0.4 (example preference function values) for their engine designs E_1 and E_2, respectively, based on thrust rating. These values are used as the weights on the graph and are all measured on the same scale. In graph theoretic terms these weights can be represented as

$$|w_{r1}| = 0.5, \qquad |w_{r2}| = 0.7, \qquad |w_{e1}| = |w_{e2}| = 0.8$$

$$|w_{e3}| = |w_{e4}| = 0.4, \qquad |w_{t1}| = |w_{t2}| = 0$$

From the preference function values of the designs, we see that the radiator designer prefers R_1 to R_2, and the engine designer prefers E_2 to E_1. The individual preference function values are the initial weights on the graph and are based only on the local performance information. However, the radiator that is more efficient could be larger in size and could occupy most of the volume under the hood, forcing the designer to choose a smaller engine, which might be less powerful. The same can be said if the most powerful and possibly largest engine is chosen, thereby restricting the radiator size. These are the factors that the system designer considers, as she/he chooses a combination of subsystems that meets the system objectives, using the information provided by the subsystem engineers.

In this problem, although the radiator designer rates the subsystem designs based on the properties most preferred, a highly rated radiator might not necessarily function effectively when combined with the best engine design chosen by the engine designer. Therefore, the system manager has to search for the appropriate choice of radiator–engine combination based on the information (weights) obtained from the individual subsystem designers.

The shortest path algorithm finds the shortest path (best individual subsystem design concepts) between the start (S) and end (T) nodes. Often the subsystem design concepts chosen are not compatible. Hence, there is a need to choose the second shortest path that will have a different combination of subsystem concepts and then the third shortest path and so on. This solution technique is incorporated using a kth shortest path algorithm, where k shortest paths are identified. Previous studies have shown that using the dynamic programming algorithm[7] might not be as effective as this approach because if any two subsystem combinations are incompatible then a penalty is applied across all of the arcs that define that combination. This could leads to elimination of any of those individual subsystems during future iterations.

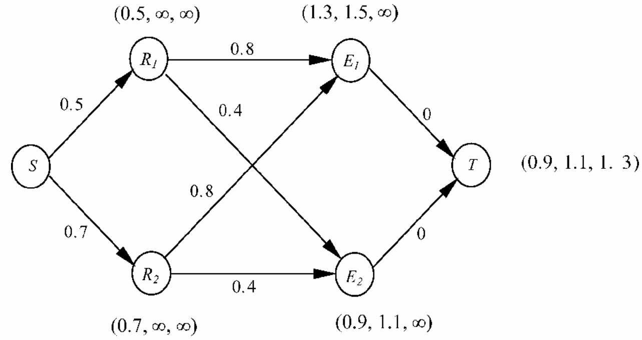


Fig. 5 Solution: engine–radiator problem (for $k = 3$, number of shortest path).

The approach being presented in this paper follows these steps:

1) Model the subsystem selection problem using nodes and arcs where the nodes are subsystem alternatives and the arc weights are the preference function values of the local measures of merit (obtained with physical programming). An example of this model is shown in Fig. 4.

2) Define the value of k, the number of desired paths (subsystem combinations). The value of k depends on the number of feasible solutions a designer wishes to identify for further investigation and evaluation using the system-level objectives. In Fig. 4 the value of k is set to three. A vector of k elements, called a k vector, is created for each node and initialized to $(\infty, \infty, \ldots, \infty)$, as shown in Fig. 4.

3) Invoke the kth shortest path algorithm. A brief description of the algorithm is given here, and full details are given by Lawler.[12] The algorithm is implemented for nonnegative weights and noncyclic networks. Both of these conditions are met by using physical programming and the network model already described. To each node is associated a k vector. This vector will record the k shortest paths to reach that node from a start node in the form of weights. The lowest value in a given vector will be the best path to reach that node. The weights (paths) are stored in ascending order in each vector.

As an example, we refer to Fig. 5, where the two nodes associated with the first subsystem R_1 and R_2 each have only one possible path entering the node. The shortest path is simply the value of the arc weight leading into the node (obtained with physical programming). In the second subsystem there are two possible paths for each node, and the lengths of these paths are sorted in ascending order in the k vector associated with each node. Because there are only two possible paths for each node, the third element in the k vector is still ∞. At the terminal node there are four possible paths, and the shortest three are kept and sorted in the three-element vector.

After the best k paths for each node are calculated, the feasibility between nodes from different subsystems is checked using compatibility relationships. If the subsystem combination is not feasible, then the combination is not considered and is discarded from the k vector. At the terminal node only feasible solutions are present; which number might be less than k, depending on the problem.

4) The resulting feasible solutions contained in the k vector of the end node of the graph represent the best solutions. The number of feasible solutions will typically be less than k, as shown in the case studies in Sec. IV. Given the set of feasible subsystem combinations, system-level objectives are then used to select the best solution. Physical programming is also used to model the preferences of the system-level objectives.

We conclude this section by emphasizing three salient focus areas of this paper:

1) Balancing local and global measures of merit is the first area. We use physical programming and determine the best set of feasible solutions based on local subsystem measures of merit, followed by

the determination of the overall optimal solution from this set using global system-level merits.

2) Handling the discrete optimization problem is the second area. A kth shortest path algorithm is used to determine the k shortest paths and adapted to discrete path planning problems. Physical programming is then used to evaluate the resulting feasible solutions to determine the best overall solution.

3) Handling the multidisciplinary nature of the problem is the final area. The coupled subsystem selection problem is modeled using concepts from graph theory, namely, sets of nodes and arcs that represent the subsystem options and subsystem measures of merit, respectively.

These developments are illustrated using the study conducted in the next section. The case study is a three-discipline problem for a passenger aircraft, requiring the selection of wings, engines, and fuselages from existing designs in order to develop a preliminary conceptual configuration.

IV. Case Studies

In Sec. IV.A the design of a passenger aircraft involving the selection of design concepts for fuselage, wing, and engine subsystems is presented. Although the total number of possible combinations for the problem does not present significant computational challenge, this aircraft problem entails an adequate level of complexity to illustrate the proposed approach. A larger problem is presented in Sec. IV.C to demonstrate the effectiveness of the approach.

A. Design of a Passenger Aircraft

In this case study the overall system objectives are to design a passenger aircraft capable of achieving a cruise range between 3000 and 5000 km and a takeoff weight between 120,000 and 230,000 kg. The design of the passenger aircraft consists of three subsystems design teams: fuselage, wing, and engine. Each of these subsystems has five alternatives, as shown in Table 1. The final aircraft design consists of the best combination of subsystem alternatives, taking into account both of the subsystem-level objectives, the compatibility constraints, and the system-level objectives. The fuselage designer evaluates his/her design alternatives based on the payload, sets a desirable payload of 70,000 kg, and optimizes the design alternatives by using class 2S of physical programming. The wing designer prefers a wing with the highest lift-to-drag ratio, sets a target value of 18, and uses class 2S preference function. The engine designer evaluates the design alternatives based on cruise thrust and prefers a cruise thrust greater than or equal to 50,000 kN. Class 2S preference function is also used for the engine weight evaluation.

A single physical constraint imposed on the problem is that the maximum number of engines that can be mounted on the aircraft is four. The preference function values for the subsystem objectives form the weights of the arcs of the graph. The kth shortest path algorithm is then used to find the feasible solutions. The number of iterations will depend on the value of k. This number is decided by the designer based on the size of the problem and computation and time resources.

The system objectives that are used to evaluate the final aircraft designs are range and takeoff weight. The range of the aircraft consists of only the cruising distance and does not include the distance covered during climb and loiter. The takeoff weight of the aircraft is the sum of the weights of the fuselage, wings, engines, payload, and fuel. The stress and other detailed aerodynamic calculations of the resulting concepts would be evaluated during the detailed design stage of the design process in order for structural integration. This technique provides feasible design combinations from among the available subsystem designs for further analysis in the detailed design phase.

In this case study we consider seven segments in the aircraft mission. At the end of each segment, there is a reduction in the weight of the aircraft as a result of fuel usage. The seven segments[13,14] are as follows: segment 1, engine start/warm-up; segment 2, taxi; segment 3, takeoff; segment 4, climb; segment 5, cruise; segment 6, descent; and segment 7, engine shutdown.

Table 1 lists a subset of the characteristics of the aircraft subsystem alternatives.[15] The subsystem alternatives are components of existing aircraft. The subsystem alternatives with the same design number belong to the same aircraft. Thus, fuselage 3, wing 3, and engine 3 are all from an existing aircraft such as a DC-10. Full details of this model are given in previous works.[7,16] Atmospheric properties used in the calculation of cruise range and other parameters are obtained from U.S. standard atmosphere data.[17] Based on aircraft equations,[13,14,18] such properties as cruise range and takeoff weight are calculated and used in the solution process for the aircraft problem.

Table 1 Engine, fuselage, and wing designs

Design	Alternative 1	2	3	4	5
		Fuselage			
W_f	103,688	64,747	64,353	26,594	48,782
A_f	2,056	1,133	1,194	579	876
W_p	67,360	37,784	44,678	18,144	31,260
		Wing			
W_w	51,844	32,373	32,176	13,297	24,391
S_{ref}	511	322	330	153	286
b	59.64	47.34	47.34	32.92	47.57
A_R	6.96	6.95	6.8	7.07	7.9
C_L	1.1065	1.107	1.094	1.115	1.179
$(L/D)_C$	13.02	13.01	12.88	13.12	13.88
		Engine			
W_e	3,982	4,171	3,582	1,520	4,029
T_{RC}	47.5	43.9	41.1	18.2	50
T_{RTO}	208.3	187	182.4	64.5	213.5
C	18.16	17.7	9.91	22.86	17.99

B. Results and Discussion

Using the kth shortest path algorithm, a set of best feasible designs is found. Using the subsystem objectives of payload, lift-to-drag, and cruise thrust, as well as a set of compatibility constraints, the top seven feasible solutions are found and are shown in Table 2 along with the results from a previous penalty-based approach.[7] The design combinations in this table are presented in the order of fuselage-wing-engine (F-W-E). Also in Table 2, the values of the system-level objectives, cruise range and takeoff weight, for each combination are shown along with the resulting preference function value from physical programming (using the preference ranges

Table 2 Aircraft subsystem combinations

kth shortest path					Penalty-based approach[16]			
Combo F-W-E	Cruise range, km	Takeoff weight, kg	No. of engines	Preference function (p_i)	Combo F-W-E	Cruise range, km	Takeoff weight, kg	No. of engines
5-5-3	4,170.44	138,768	3	12.87	5-5-3	4,170.4	138,770	3
5-2-5	4,625.00	172,129	3	14.22	5-2-1	4,587.3	171,990	3
5-2-2	4,683.64	172,555	3	14.29	2-3-5	4,423.9	201,220	3
5-2-1	4,587.26	171,988	3	14.37	2-2-2	3,778.5	199,220	4
5-4-3	3,920.80	125,309	3	58.88	5-4-3	3,920.8	125,310	3
2-2-5	3,851.12	194,619	3	280.01	3-3-2	4,151.3	212,320	4
2-2-1	3,819.24	194,478	3	287.52	3-2-1	3,527	204,960	4

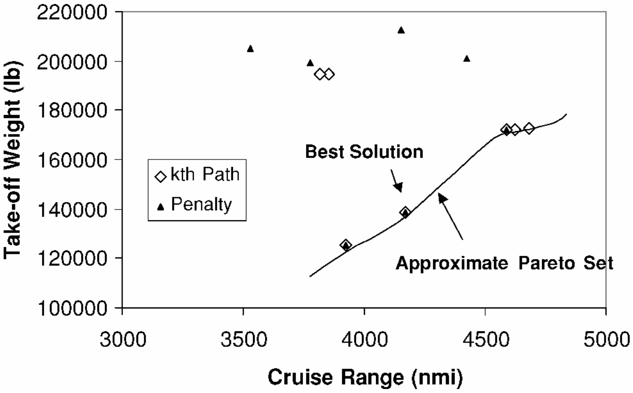


Fig. 6 Pareto plot of *k*th path solutions and penalty solutions.

given in Sec. IV.A). This value is an aggregation of the subsystem objectives and is to be minimized. The combinations are quite different from each other and can lead to new aircraft configurations not previously considered. In fact, none of the combinations obtained include a wing, fuselage, and engine from the same aircraft type. These seven designs can be taken into more detailed embodiment design to assess further feasibility and effectiveness. An important feature of using the *k*th shortest path algorithm is its ability to explore those solutions that the penalty-based approach[7] is not able to find. This is revealed by the results that are obtained using this algorithm. Also, note that none of the *k*th shortest path designs are existing aircraft.

In Fig. 6 a Pareto frontier is shown that compares the results from both the *k*th path solution and the earlier penalty-based approach. The top five solutions found by the *k*th path approach are all Pareto optimal and lie along the approximated Pareto frontier. The penalty-based approach yielded only three of the five Pareto solutions. Because this problem is a relatively small one (125 combinations), it is straightforward to evaluate all 125 combinations. When this is performed, it is found that there are only five Pareto optimal solutions of the 125 possible solutions, all of which are obtained with the *k*th path algorithm. Therefore, the *k*th path algorithm identified all five of these solutions without having to evaluate the exhaustive combinatorial possibilities. The overall best solution from physical programming is one of the Pareto solutions, as shown in Fig. 6. Because the solution is a set of feasible designs, there is sufficient flexibility to choose the best design using the preferences on the system-level objectives.

C. Pressure Vessel

In light of the absence of a benchmark problem involving two or more subsystems having a number of design alternatives, the design of a pressure vessel involving discrete variables is chosen to explore further the capability of the algorithm. This is a benchmark discrete optimization problem involving mixed discrete and continuous variables, which is modified to represent a design situation wherein the two subsystems (volume and weight) control two sets of design variables each.[19] The cylindrical pressure vessel is capped at both ends by hemispherical heads as shown in Fig. 7. The goal of this design problem is to determine appropriate dimensions for the pressure vessel based on several constraints, including a volume constraint, so as to obtain the total cost of manufacturing the pressure vessel within a user-defined range. The problem consists of four design variables: R and L, which are, respectively, the inner radius and length of the cylindrical section; and T_S and T_h, respectively, the thickness of the shell and head. The design variables are given in inches. The problem is as given here:

Find

System variables:

R, L, T_S, T_h

Satisfy

Constraints:

c_1, minimum shell wall thickness:

$$0.0193R \leq T_S$$

c_2, minimum head wall thickness:

$$0.00954R \leq T_h$$

c_3, maximum length:

$$L \leq 240.0$$

c_4, minimum volume of tank:

$$\frac{-(4/3)\pi R^3 + 1{,}296{,}000}{\pi R^2 L} \leq 1.0$$

Goal

$$\text{Cost} = 0.6224 T_S R L + 1.7781 T_h R^2 + 3.1661 T_S^2 L + 19.84 T_S^2 R$$

$$\text{Target cost} = 5000 \leq \text{Cost}_{\text{TV}} \leq 7000$$

where Cost_{TV} is the target total cost value for manufacturing the pressure vessel.

Bounds

$$R_{\text{LB}} \leq R \leq R_{\text{UB}}$$

$$L_{\text{LB}} \leq L \leq L_{\text{UB}}$$

$$T_{S\,\text{LB}} \leq T_S \leq T_{S\,\text{UB}}$$

$$T_{h\,\text{LB}} \leq T_h \leq T_{h\,\text{UB}}$$

The bounds for the design variables are $R_{\text{LB}} = 25$ in., $R_{\text{UB}} = 150$ in., $L_{\text{LB}} = 25$ in., $L_{\text{UB}} = 250$ in., $T_{S\,\text{LB}} = 0.05$ in., $T_{S\,\text{UB}} = 1.25$ in., $T_{h\,\text{LB}} = 0.05$ in., and $T_{h\,\text{UB}} = 1.25$ in. Previous studies of the problem have the target value for cost of $\text{Cost}_{\text{TV}} = \$5000$ (Ref. 19).

In the pressure vessel problem there are two subsystems: volume and weight of the vessel. Each of the variable sets represents a column or node set in the problem setup, that is, R, L, T_S, and T_h are the four node sets in the problem. R and L are integral multiples of 5 within [25, 150] and [25, 250], respectively. T_S and T_h are integral multiples 0.05 within the upper and lower bounds already provided. The number of possible variable combinations is $26 \times 46 \times 25 \times 25 = 747{,}500$. R and L values dictate the volume of the vessel and are controlled by the volume design team whose objective is to maximize the volume. Therefore, the R and L values are modeled using the class 2S preference function or maximization. T_S and T_h play a key role in deciding the weight of the vessel

Table 3 Pressure vessel design results

	Solution				
Parameter	1	2	3	4	5
R, in.	40	40	40	40	40
L, in.	205	210	205	215	210
T_S, in.	0.80	0.80	0.80	0.80	0.80
T_h, in.	0.40	0.40	0.45	0.40	0.45
Cost, $	6144.22	6253.94	6286.47	6363.66	6396.19

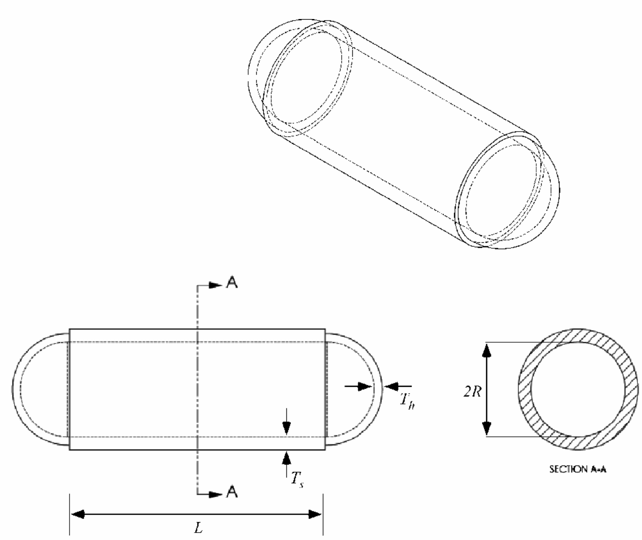


Fig. 7 Pressure vessel.

Table 4 Comparison of pressure vessel results

Parameter	kth path/PPro	Penalty[7,16]	Lewis and Mistree[19]	Hsu et al.[20]	Lin et al.[21]	Kannan and Kramer[22]	Sandgren[23]
R, in.	40	40	38.76	51.81	N/A	58.29	47.7
L, in.	205	210	223.3	101.85	N/A	43.69	117.7
T_S, in.	0.80	0.80	0.75	1.00	N/A	1.125	1.125
T_h, in.	0.40	0.55	0.375	0.5	N/A	0.625	0.625
Cost, $	6144.22	6680.70	5869.5	7021.67	7197.7	7198.2	8129.8

and are controlled by the weight design team, whose objective is to minimize the weight of the pressure vessel. Therefore, class 1S preference function, or minimization, is chosen to model T_S and T_h. Preference functions are constructed for the design variables R, L, T_S, and T_h, which are used as the subsystem-level objectives. The constraints c_1, c_2, and c_3 correspond to American Society of Mechanical Engineers limits on the geometry, whereas c_4 corresponds to minimum volume limit.

The system-level objective is the cost of the pressure vessel, and because the target cost is a range it is modeled by class 4S preference function. The preference function for cost is the global objective. Therefore, for the set of k feasible solutions generated using the subsystem-level objectives the five best solutions based on the overall cost are shown in Table 3. All of the constraint values are feasible, and solution 1 denotes the best solution obtained. In Table 4 the best solution obtained from this algorithm is compared with those obtained from previous studies.[7,16,19–23] The solution obtained using the kth path algorithm with physical programming found a much better solution than all of the previous approaches except for the study in Lewis and Mistree.[19] However, the pressure vessel model in the Lewis study modeled the radius R and the length L as continuous variables, allowing for detailed refinement of the solution. The design variable values of these two solutions are very similar. Both solutions seem to be located in the neighborhood of the global optimal solution. We expect that an even better solution could be obtained by using the kth path algorithm if a finer discretization were allowed.

A better solution is found in this study compared to the previous penalty approach[7,16] using the same discretization because the penalty approach as noted in Sec. III potentially eliminates portions of the design space where optimal solutions might lie. This characteristic of the penalty approach is evident from the results, as it was not able to yield the solution found in this study. In addition, the solution found in this study is better than the other previous studies,[20–23] even though the previous studies modeled the problem as a continuous optimization problem. This is because of the ability of the proposed approach to avoid being confined to local optimal solution regions. The previous solution approaches were apparently searching in local optimal solution regions.

V. Conclusions

In industry, product development is performed to develop new subsystem components so that the effectiveness of an existing complex system can be increased. Under such circumstances designers might wish to assess the performance of these newly developed subsystem design components quickly when integrated with the other subsystems. The approach presented in this paper can be used effectively to obtain sets of subsystem combinations that would perform successfully with the newly developed components. The aircraft case study presented in this paper suggests that the method is capable of identifying new effective designs, when the subsystem alternatives are chosen from existing aircraft models. Given the required properties of the aircraft, the developed approach generated combinations of subsystem designs from different aircraft models. The pressure vessel example suggests that the method is capable of avoiding local optimal solution regions. In addition, the use of physical programming provides an intuitive method to model preferences of local and global objectives, making this approach applicable to a wider set of both single and multiobjective selection problems.

This approach might help to reduce the time and cost in product development by identifying promising subsystem design combinations, so that focus could be restricted to these designs for further testing and development. As a potential extension, the use of Dijkstra's algorithm could be implemented to problems that are cyclic in nature. A cyclic loop would represent the coupling of different subsystems more effectively, as two subsystems typically involve mutual dependencies. In such a case, there could be a cyclic loop introduced between two subsystems, and the algorithm can be used to find the values of the arcs through some numerical convergence. In addition, combining discrete off-the-shelf selections with adjustable, continuous parameter selection would help increase the applicability of this approach to more systems.

Acknowledgments

This multidisciplinary research project is a joint effort between the State University of New York at Buffalo and Rensselaer Polytechnic Institute, Troy, New York. We gratefully acknowledge the support of the National Science Foundation through Grant DMI-9875607 for Kemper Lewis and Grant DMI-0196243 for Achille Messac.

References

[1] Siddique, Z., and Rosen, D., "On Combinatorial Design Spaces for the Configuration Design of Product Families," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 15, No. 2, 2001, pp. 91–108.

[2] Ulrich, K. T., and Eppinger, S. D., *Product Design and Development*, McGraw–Hill, New York, 1995, Chap. 6.

[3] Hazelrigg, G. A., *Systems Engineering: An Approach to Information-Based Design*, Prentice–Hall, Upper Saddle River, NJ, 1995, pp. 155–174.

[4] Yang, J., and Sen, P., "A Hierarchical Evaluation Process for Multiple Attribute Design Selection with Uncertainty," *Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-93)*, edited by P. W. H. Chung, Gordon and Breach, New York, 1993, pp. 484–493.

[5] See, T. K., and Lewis, K., "Multi-Attribute Decision Making Using Hypothetical Equivalents," American Society of Mechanical Engineers, Paper DETC02/DAC-02030, Sept.–Oct. 2002.

[6] Messac, A., "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 149–158.

[7] Ramaswamy, V., and Lewis, K., "Conceptual Design of a Complex Engineering System Through Coupled Selection," AIAA Paper 98-4882, Sept. 1998.

[8] Messac, A., Melachrinoudis, E., and Sukam, C. P., "Mathematical and Pragmatic Perspectives of Physical Programming," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 885–893.

[9] Messac, A., "From the Dubious Construction of Objective Functions to the Application of Physical Programming," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 155–163.

[10] Tappeta, R. V., Renaud, J. E., Messac, A., and Sundararaj, G. J., "Interactive Physical Programming: Tradeoff Analysis and Decision Making in Multidisciplinary Optimization," *AIAA Journal*, Vol. 38, No. 5, 2000, pp. 917–926.

[11] Winston, W. L., *Introduction to Mathematical Programming: Applications and Algorithms*, Duxbury Press, Belmont, CA, 1995, pp. 405–412.

[12] Lawler, E. S., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, Philadelphia, 1976, pp. 70–75.

[13] Roskam, J., *Airplane Design Part 1: Preliminary Sizing of Airplanes*, Roskam Aviation and Engineering Corp., Ottawa, 1989, Chaps. 2 and 3.

[14] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA, Washington, DC, 1992.

[15] *Jane's All the World's Aircraft 1981–1982*, Jane's Publishing Co., London, 1981.

[16]Ramaswamy, V., "Conceptual Design of Complex Engineering Systems Through Coupled Selection," M.S. Thesis, Dept. of Mechanical and Aerospace Engineering, State Univ. of New York, Buffalo, NY, Dec. 1998.

[17]*U.S. Standard Atmosphere, 1962*, U.S. Government Printing Office, Washington, DC, 1962.

[18]White, F. M., *Fluid Mechanics*, McGraw–Hill, New York, 1994, Chap. 7.

[19]Lewis, K., and Mistree, F., "FALP: Foraging-Directed Adaptive Linear Programming, A Hybrid Algorithm for Discrete/Continuous Design Problems," *Engineering Optimization*, Vol. 32, No. 2, 1999, pp. 191–218.

[20]Hsu, Y.-L., Sun, T.-L., and Leu, L.-H., "A Two-Stage Sequential Approximation Method for Non-Linear Discrete Variable Optimization," *American Society of Mechanical Engineers Design Engineering Technical Conference*, Vol. 2, edited by S. Azarm, D. Dutta, H. Eschenauer, B. J. Gilmore, M. McCarthy, and M. Yoshimura, American Society of Mechanical Engineers, New York, 1995, pp. 197–202.

[21]Lin, S.-S., Zhang, C., and Wang, H.-P., "On Mixed-Discrete Nonlinear Optimization Problems: A Comparative Study," *Engineering Optimization*, Vol. 23, No. 4, 1995, pp. 287–300.

[22]Kannan, B. K., and Kramer, S. N., "An Augmented Lagrange Multiplier Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design," *Journal of Mechanical Design*, Vol. 116, No. 2, 1994, pp. 405–411.

[23]Sandgren, E., "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design*, Vol. 112, No. 2, 1990, pp. 223–229.

E. Livne
Associate Editor