DeMAID/GA
AN ENHANCED DESIGN MANAGER'S AID FOR INTELLIGENT DECOMPOSITION

J. L. Rogers*
NASA Langley Research Center

Abstract

Many companies are looking for new tools and techniques to aid a design manager in making decisions that can reduce the time and cost of a design cycle. One tool is the Design Manager's Aid for Intelligent Decomposition (DeMAID). Since the initial public release of DeMAID in 1989, much research has been done in the areas of decomposition, concurrent engineering, parallel processing, and process management; many new tools and techniques have emerged. Based on these recent research and development efforts, numerous enhancements have been added to DeMAID to further aid the design manager in saving both cost and time in a design cycle. The key enhancement, a genetic algorithm (GA), will be available in the next public release called DeMAID/GA. The GA sequences the design processes to minimize the cost and time in converging a solution. The major enhancements in the upgrade of DeMAID to DeMAID/GA are discussed in this paper. A sample conceptual design project is used to show how these enhancements can be applied to improve the design cycle.

Introduction

In today's competitive environment, companies are under enormous pressure to reduce the time and cost of their design cycle. One method for reducing both time and cost is to develop an understanding of the flow of the design processes and the effects of the iterative subcycles that are often found in complex design projects. Once these are understood, the design manager can make decisions that take advantage of decomposition, concurrent engineering, and parallel processing techniques to reduce the total time and total cost of the design cycle. One software tool that can aid in this decision-making process is the Design Manager's Aid for Intelligent Decomposition (DeMAID). The original version of DeMAID was released to the public in 1989. This version is a knowledge-based software tool for minimizing the feedback couplings; sequencing the design processes; grouping processes into iterative subcycles; decomposing these subcycles into a hierarchical, multilevel structure for a design cycle; and displaying the sequence of processes in a design structure matrix (DSM) format[1, 2].

Feedback couplings between design processes create iterative subcycles. Once the iterative subcycles have been determined, the real benefits of producing the best design in the least time and at a minimum cost are obtained from sequencing the processes within these subcycles. In the original version of DeMAID, the processes within a subcycle are sequenced by using a knowledge base. The knowledge base, however, only sequences the processes to minimize the number of feedback couplings. No significant effort was made with DeMAID to determine the best sequence in regard to saving both time and cost in an iterative subcycle.

Since the initial public release of DeMAID in 1989, extensive research has been done in the areas of decomposition, concurrent engineering, parallel processing, and process management. A number of new tools and techniques have been developed. Based on these recent efforts, numerous enhancements have been incrementally added to DeMAID. These enhancements, including a genetic algorithm (GA), have been documented in several publications and will be available in the next public release of DeMAID/GA[3, 4, 5]. This paper begins with an overview of the DSM; and then provides a description of the enhancements made to DeMAID/GA, and shows how they can be applied to reduce the time and cost of the design cycle.

Design Structure Matrix

Any nontrivial project consists of numerous processes that are dependent on one another. This interdependency can become quite complicated. To reduce this complexity, numerous approaches were developed to aid in understanding and managing these

* Senior Computer Scientist.

processes. These approaches include the Program Evaluation and Review Technique (PERT) and the Critical Path Method (CPM)[6]. Unfortunately, these tools are only applicable to sequential and parallel activities and cannot handle the iterative subcycles that are often found in complex design problems. A PERT-related tool, called the General Evaluation Review Technique (GERT), can handle iterative subcycles but is only effective for simple problems[7]. A new tool was developed to manage and display the sequence of design processes in a complex design project and to handle the iterative subcycles that are commonly found in these types of projects. This tool, called the design structure matrix (DSM), was originally formulated by Steward[8]. A sample DSM is shown in figure 1.

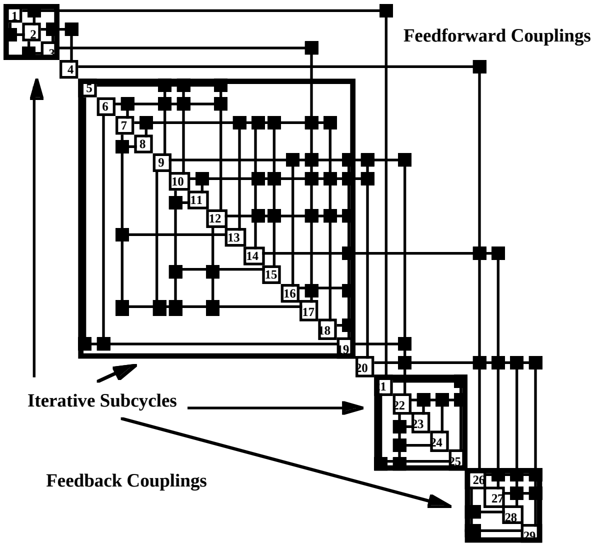

Figure 1. Design structure matrix.

The DSM is derived from graph theory and is used to display the design process sequence in DeMAID. In the DSM in figure 1, the design processes are shown as numbered boxes on the diagonal. Output from a process is shown as a horizontal line that exits a numbered box, and input to a process is shown as a vertical line that enters a box. The off-diagonal squares that connect the horizontal and vertical lines represent couplings between two processes. Squares in the upper triangle of the DSM represent feedforward couplings; squares in the lower triangle of the matrix represent feedback couplings. Feedback couplings imply iterations in which initial data estimates must be made. Feedback couplings should be eliminated if possible; however, in many cases, not all of the feedback couplings can be eliminated. If certain feedback couplings cannot be eliminated, the processes are grouped into iterative subcycles. In figure 1, processes 1-3, 5-19, 21-25, and 26-29 are grouped into iterative subcycles.

The primary advantage of the DSM format over earlier display tools is the capability to group and display the iterative subcycles commonly found in a design project. After the iterative subcycles have been determined, their processes should be sequenced in such a way to produce the best design in the least time at a minimum cost. The total cost and time to complete a design project are dependent on the sequencing of the processes in the iterative subcycles[5]. A large, iterative subcycle, such as the one in figure 1 that contains processes 5-19 can be very expensive to converge because the iterations contained in feedback loops defined by the feedback couplings are nested. In this figure, there is a feedback loop containing processes 12-15 coupled by a feedback from process 15 to process 12. This loop is nested within another feedback loop containing processes 12-17 coupled by a feedback from process 17 to process 12. These nested feedback loops may require numerous executions of potentially expensive processes. Any software tool that is based on the DSM must have the capability to rapidly examine many different sequences of processes within an iterative subcycle and select the optimal sequence based on cost, time, and iteration requirements.

DeMAID/GA Enhancements

Many features in the original version of DeMAID have been incorporated into DeMAID/GA. These features include minimizing feedbacks, grouping processes into iterative subcycles, determining the hierarchical decomposition, and forming a dependency matrix.

In addition, DeMAID/GA has many new enhancements not available in the original version of DeMAID. Some of these enhancements have not been reported in previous papers including pull-down menus; an on-line help function; windows; and a function for the user to define rules. The DSM can now also be displayed in reverse format (Steward's original format) with the feedforward couplings shown in the lower triangle and the feedback couplings shown in the upper triangle[8]. Furthermore, a new interface function has been added to DeMAID/GA to take advantage of other process management tools. This interface function allows the user to save a file that can be input to other programs that create spreadsheets and PERT charts. A detailed description of these enhancements can be found in the DeMAID/GA user's guide[9].

The enhancements discussed in this section pertain to cost and time reductions in the design cycle. These enhancements include: quantifying coupling strengths, computing the cost and time of an iterative subcycle, optimizing the sequence of design processes within an iterative subcycle with a GA, identifying parallel processing opportunities within an iterative subcycle, expanding the detail of output values, dividing large iterative subcycles, and tracing changes in the design cycle. A sample conceptual design project of a representative airplane is used in this discussion. The process flow for this conceptual design project is shown in figure 2.

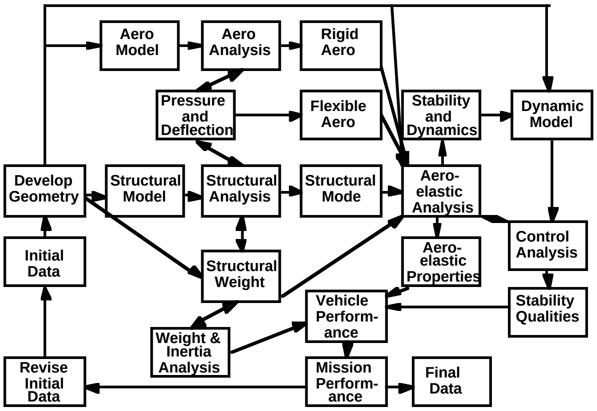

Figure 2. Process flow for a sample conceptual design project.

The problems with this representation are the difficulties in visualizing the iterative subcycles, identifying the feedback loops within these subcycles, and determining the optimum sequence of the processes to complete the design cycle.

The DSM in figure 3 shows the process sequence for the project shown in figure 2 after the iterative subcycle has been determined but before the processes in the iterative subcycle have been sequenced by the GA in DeMAID/GA. The initial sequence, times, and costs associated with the design processes in this figure were arbitrarily selected. Typically, a large design project will have several large iterative subcycles, as shown in figure 1[10]. The example here is limited to a single iterative subcycle containing numerous nested iterations (feedback loops) within the subcycle to illustrate the effectiveness of the GA in optimizing a sequence of design processes.

	Process	Time	Cost
1	DYNMODL	30	30
2	STABDYN	40	20
3	STRMODL	10	50
4	STBQUAL	10	50
5	STRMODE	10	50
6	GEOMDEV	50	10
7	AEPROP	40	20
8	AEANALY	50	10
9	CTANALY	20	40
10	FLXAERO	20	40
11	INITDAT	40	20
12	RVSEDAT	30	30
13	MISPERF	30	30
14	VEHPERF	20	40
15	RGDAERO	30	30
16	AEROANL	20	40
17	PRESDEF	30	30
18	STRANAL	40	20
19	STRCTWT	50	10
20	WIANALY	40	20
21	AEROMDL	20	40
22	FINLDAT	20	40

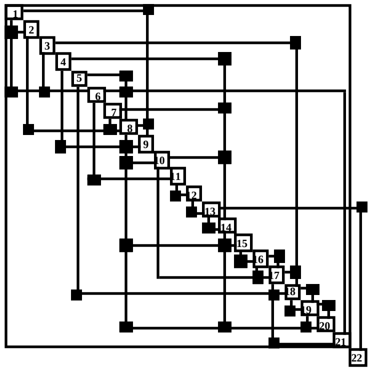

Figure 3. Unsequenced DSM for sample conceptual design project.

In a large iterative subcycle such as the one in this figure, there are numerous possible sequences in which to execute the design processes. Many sequences, including this one, are probably meaningless to a design manager. The design manager needs a meaningful sequence, or at least a reasonable attempt at a meaningful sequence that will minimize both cost and time of an iterative subcycle. The knowledge-based minimization of feedback couplings while very effective in finding the iterative subcycles; was not effective in sequencing the design processes contained within them. Therefore, a decision was made to add a GA to sequence the design processes within the iterative subcycles. Two other enhancements, quantifying coupling strengths and determining the "goodness" of a sequence based on the cost and time of an iterative subcycle, preceded the addition of the GA and were incorporated into it. These two enhancements will be discussed first, then the GA, followed by the other cost and time saving enhancements.

Quantifying Coupling Strengths

Much of the work regarding coupling strengths and the incorporation of these strengths into DeMAID has been previously reported[4, 11]. This approach relies on sensitivity analysis to determine the strength of a coupling. Seven levels of coupling strengths are available in DeMAID/GA and range from extremely weak to extremely strong. These strengths are color coded for easy recognition in the DSM.

A recent addition to this enhancement is the postulation of a relation between the coupling strengths and the number of iterations (iteration factor) required to converge processes joined by a feedback coupling. The defaults for the iteration factors were arbitrarily selected and range from two iterations for an extremely weak coupling to eight iterations for an extremely strong coupling. These defaults can be overridden by the user as necessary. Currently, no formal approach exists to directly relate coupling strengths and the number of iterations. For some problems, past experience may serve as a guide; for others, some experimentation may be required. The iteration factors are used to help the GA determine the best sequence of the processes within an iterative subcycle. If no coupling strengths are available, a total of five iterations (the default for a nominal coupling) is assumed for computing purposes.

Computing Cost and Time

The original version of DeMAID has times associated with the processes but they are not used to optimize the sequence. The only sequence optimization is based on the minimization of feedback couplings. A cost can now be associated with the processes in DeMAID/GA. The user determines the units for cost and time.

A new feature was added to DeMAID/GA to sum the time and cost of each process contained in a feedback loop and multiply those sums by the iteration factor for the feedback coupling to determine the time and cost for each feedback loop in an iterative subcycle. For example, in figure 3, there is a weak feedback coupling process 6 to process 1. The times and costs of processes 1-6 are summed and then multiplied by 4, the default number of iterations to converge a weak coupling. The times and costs of each feedback loop are summed to obtain the total cost and time to converge an iterative subcycle. This method may be misleading because many of the possible sequences are probably meaningless to a design manager. However, this method appears to work well in eliminating the meaningless sequences and converging to a meaningful sequence of processes that minimize the time and cost of the iterative subcycle.

For example, the time (21,340 units) and cost (19,640 units) to converge the unsequenced iterative subcycle in figure 3 are shown in table 1. The values in the time and cost columns of the table represent the sum of the time and cost of all processes in the particular feedback loop multiplied by the iteration factor obtained from the feedback coupling strength.

Table 1. Time and Cost for Unsequenced Sample Conceptual Design Project.

To	From	Iterations	Time	Cost
1	2	8	560	400
1	6	4	600	840
2	8	8	1680	1680
3	6	2	160	320
4	9	7	1260	1260
5	18	6	2580	2460
6	11	8	1760	1120
7	8	6	540	180
8	9	2	140	100
8	10	8	720	720
8	15	4	960	960
8	20	7	2940	2520
10	17	8	1760	2080
11	12	5	350	250
12	13	3	180	180
13	14	6	300	420
14	15	8	400	560
14	20	4	920	760
15	16	6	300	420
16	17	7	350	490
16	21	8	1600	1280
17	18	8	560	400
18	19	6	540	180
19	20	2	180	60
Totals			21,340	19,640

Optimizing the Sequence of Design Processes with a GA

The major enhancement to DeMAID/GA that sets it apart from the original version of DeMAID is the addition of the GA to optimize the sequence of processes within an iterative subcycles[5]. The use of GA's has been instrumental in achieving acceptable solutions to discrete optimization problems such as the sequencing problem found in DeMAID[12, 13]. The addition of the GA capability allows DeMAID/GA to examine most, if not all, possible combinations of process sequences within in an iterative subcycle and converge to the optimal sequence. This enhancement was derived from the work at the State University of New York at Buffalo and Stanford University[14, 15, 16]. The method for computing the cost and time of a feedback loop discussed in the previous section composes the fitness function of the GA. The user can weight the importance of each factor in the fitness function and can minimize only cost, only time, or both cost and time as necessary. The GA examines and sequences the processes of one iterative subcycle at a time. The processes within a subcycle are sequenced independent of processes in other subcycles. Currently, nothing is being done to optimize a sequence of processes in one iterative subcycle in relation to the sequences in the other iterative subcycles. This would, in effect, provide an optimum sequence for the entire design cycle.

The population of the GA is a set of strings made from the concatenation of integers representing valid process numbers[16]. Successive populations are produced primarily by the operations of selection, crossover, and mutation. The GA operators have been modified to handle integers.

The selection operator determines those members of the population that survive to participate in the production of members of the next population. Selection is based on the value of a fitness function, such that members with greater fitness levels tend to survive. Selection in DeMAID/GA is accomplished by the "tournament selection" operator. The mating pool is filled by randomly selecting two strings from the parent pool, and comparing their fitness functions. The string with the greater fitness is included in the mating pool.

Crossover is the recombination of traits of selected members from the mating pool, in the hope of producing a child with a better fitness level than its parents. Crossover is accomplished by swapping parts of the string into which the sequence design processes have been coded. Crossover in DeMAID/GA is accomplished by "position-based" crossover in which several process numbers are chosen from the first parent string and placed in the same positions in the child string. Then, the process numbers that were not taken from the first parent string are taken from the second parent string to fill the holes in the child string in the order in which they appear in the second parent string. The result is a complete string with one copy of each process number.

The final operation, mutation, prevents the search for the optimum sequence from becoming too narrow. After the production of a child population, the mutation operator randomizes small parts of the resulting strings, with a very low probability that any given string position will be affected. Mutation in DeMAID/GA is accomplished through the "order-based" mutation operator, in which each string position is polled; if a given string position is selected to undergo mutation, then its content is swapped with another randomly selected position in the same string.

The sequence of processes in the iterative subcycle for the DSM shown in figure 4 has been optimized with the GA. The time and cost for converging the initial arbitrary ordering of the project (table 1) have been significantly reduced as shown in table 2. The total time has been reduced from 21,340 to 3,800 units and the total cost has been reduced from 19,640 to 3,220 units.

Process	Time	Cost
1 RVSEDAT	30	30
2 INITDAT	40	20
3 GEOMDEV	50	10
4 STRMODL	10	50
5 AEROMDL	20	40
6 AEROANL	20	40
7 PRESDEF	30	30
8 STRANAL	40	20
9 STRCTWT	50	10
10 WIANALY	40	20
11 STRMODE	10	50
12 RGDAERO	30	30
13 FLXAERO	20	40
14 AEANALY	50	10
15 STABDYN	40	20
16 DYNMODL	30	30
17 CTANALY	20	40
18 STBQUAL	10	50
19 AEPROP	40	20
20 VEHPERF	20	40
21 MISPERF	30	30
22 FINLDAT	20	40

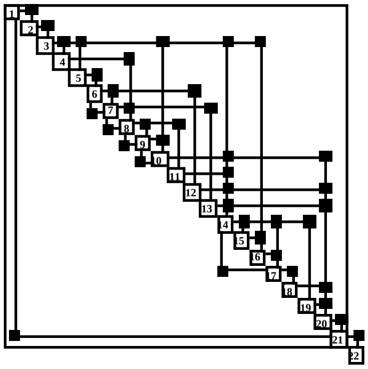

Figure 4. Sequenced DSM for sample conceptual design project.

Table 2. Time and Cost for Sequenced Sample Conceptual Design Project.

To	From	Iterations	Time	Cost
1	21	3	1890	1890
6	7	7	350	490
7	8	8	560	400
8	9	6	540	180
9	10	2	180	60
14	17	2	280	200
Totals			3,800	3,220

The current GA fitness function is only one method for determining the optimum sequence of design processes in an iterative subcycle. Other methods need to be researched to expand the robustness of the fitness function. These methods include tradeoffs between minimizing feedbacks and maximizing parallel processing opportunities (discussed in the next section) within an iterative subcycle; other algorithms such as the one proposed by Smith for computing coupling strengths; and optimizing the sequence of processes in all iterative subcycles at once rather than just optimizing one iterative subcycle at a time[17].

Identifying Parallel Processing Opportunities Within an Iterative Subcycle

Many iterative subcycles are large, and significant savings in terms of cost and time can be realized by executing some of the processes in parallel. To identify the parallel processing opportunities within an iterative subcycle, DeMAID/GA assumes that all feedback coupling data are available as estimates. By using this assumption, the processes in an iterative subcycle can be divided into a hierarchy in which any processes in the same level of the hierarchy can be executed in parallel. For example, prior to resequencing, the processes in the iterative subcycle of the DSM in figure 3 decompose into the levels shown in figure 5. In this sequence, many of the processes, particularly in level 1, can be executed in parallel.

Level 1	1 2 3 4 5 6 7 10 11 12 13 15 16
Level 2	8 14 17 21
Level 3	9 18
Level 4	19
Level 5	20

Figure 5. Levels of processes in conceptual design project prior to resequencing.

After resequencing, the processes in the iterative subcycle of the DSM in figure 4 decompose into the levels shown in figure 6.

Level 1	1
Level 2	2
Level 3	3
Level 4	4 5
Level 5	6
Level 6	7 12
Level 7	8 11
Level 8	9 13
Level 9	10
Level 10	14
Level 11	15 18
Level 12	16
Level 13	17
Level 14	19
Level 15	20
Level 16	21

Figure 6. Levels of processes in conceptual design project after resequencing.

In this sequence, even though the sequence has fewer feedback couplings, only a small number of the processes can be executed in parallel. Thus, tradeoffs exist between maximizing parallel processing opportunities and minimizing the number of feedback couplings. These tradeoffs need to be considered as a part of the fitness function of the GA.

Expanding the Detail of Output

To enable the design manager to track more detailed information, a change was made in DeMAID/GA in the way output is represented in the format of a process definition [4]. In the original version of DeMAID, the output field represents all output produced by a process, regardless of whether it is a single piece of data, a string of data, a vector, a matrix, or any combination. Design managers expressed their need for more detailed output than was allowed in the original version.

For example, in figure 7, process 1 represents a finite-element analysis program that computes stresses and displacements. In the original format, all output was combined under a single name (e.g. feout). In DeMAID/GA, this output name can be expanded into its various components. In this figure, process 1 couples to processes 2, 3, and 4. Hence, processes 2-4 require “feout” as input. Suppose, however, that process 2 really requires only stresses as input to compute stress constraints; process 3 requires only displacements to compute displacement constraints; and process 4 requires stresses, stress constraints, displacements, and displacement constraints. The DSM would appear the same; however if a change occurred in process 1 that affected only the stresses, then process 3 (shaded) would not have to be reexecuted because it does not require stress data as input.

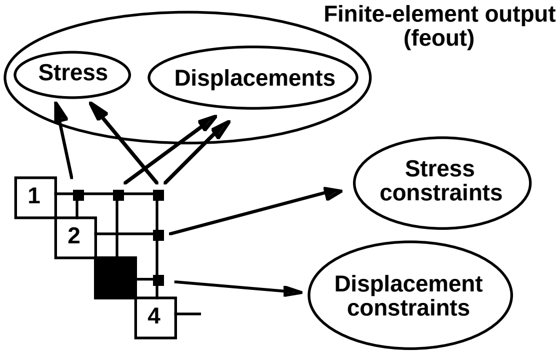

Figure 7. Processes with multiple outputs.

Dividing Large Iterative Subcycles

Similar to Steward's method of tearing, DeMAID/GA makes recommendations on the removal or temporary suspension of certain processes and couplings based on the strength of the coupling[4,8]. By following the recommendations, large iterative subcycles such as the one in figure 4 can be divided into smaller subcycles (figure 8) by removing couplings and/or processes.

	Process	Time	Cost
1	INITDAT	40	20
2	DEVGEOM	50	10
3	STRMODL	10	50
4	AEROMDL	20	40
5	AEROANL	20	40
6	PRESDEF	30	30
7	STRANAL	40	20
8	STRCTWT	50	10
9	WIANALY	40	20
10	STRMODE	10	50
11	RGDAERO	30	30
12	FLXAERO	20	40
13	AEANALY	50	10
14	STABDYN	40	20
15	DYNMODL	30	30
16	CTANALY	20	40
17	STBQUAL	10	50
18	AEPROP	40	20
19	VEHPERF	20	40
20	MISPERF	30	30
21	FINLDAT	20	40

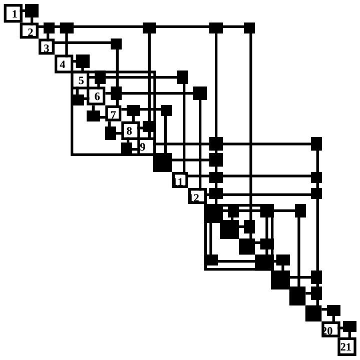

Figure 8. DSM of modified sample conceptual design project.

In this example, process RVSEDAT (Revise Data) was removed only to demonstrate how a large iterative subcycle might be divided into smaller subcycles by the removal of a single process. This could also have been accomplished by removing the coupling from the MISPERF (Mission Performance) process to the RVSEDAT process.

Tracing Design Changes

Changes are frequently made during the design cycle. Simply because a change is made does not necessarily mean that all design processes must be reexecuted. Another new capability in DeMAID/GA identifies the processes affected by a design change[4]. For example, in figure 8 the DSM displays those processes (shaded) that must be reexecuted to determine the effects of a change in STRMODE (Structural Mode) on VEHPERF (Vehicle Performance). In this simple example, RGDAERO (Rigid Aero) and FLXAERO (Flexible Aero) are not affected by the change in STRMODE. If either or both of these processes are expensive or use a critical resource, then a substantial savings may be realized.

Concluding Remarks

Much of the cost and time involved to complete a complex multidisciplinary design project is attributable to expensive iterative subcycles. New tools and techniques have been researched and developed to aid the design manager in reducing these costs. Significant enhancements based on these developments have been added to a software tool known as the Design Manager's Aid for Intelligent Decomposition (DeMAID/GA). The major enhancement is the addition of a genetic algorithm (GA). After decomposing the design project into iterative subcycles and, where possible, quantifying the coupling strengths between the design processes, the GA sequences the processes within each iterative subcycle to minimize the time and cost to converge to the design solution. Other added capabilities enable the design manager to identify opportunities at which to take advantage of executing design processes in parallel within a subcycle; expand the detail of the output; divide large iterative subcycles; and trace the effects of changes in the design cycle. A sample conceptual design of a representative aircraft demonstrates the gains that can be made by applying these enhancements to a complex design project.

References

1. Rogers, J. L., "A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem, " NASA TP-2903, 1989.

2. Rogers, J. L., "DeMAID - A Design Manager's Aid for Intelligent Decomposition User's Guide," NASA TM-101575, 1989.

3. Rogers, J. L. and Barthelemy, J.-F. M., "Enhancements to the Design Manager's Aid for Intelligent Decomposition (DeMAID)," AIAA Paper No. 92-4809, 1992.

4. Rogers, J. L., and Bloebaum, C. L., "Ordering Design Tasks Based on Coupling Strengths," AIAA Paper No. 94-4326, 1994.

5. Rogers, J. L., McCulley, C. M., and Bloebaum, C. L., Integrating a Genetic Algorithm into a Knowledge-Based System for ordering Complex Design Processes, Fourth International Conference of Artificial Intelligence in Design, Stanford University, CA, June 1996.

6. Kerzner, H., Project Management. Van Nostrand Reinhold, New York, 1989.

7. Neumann, K., and Steinhardt, U., GERT Networks and the Time-Oriented Evaluation of Projects, Springer-Verlag, New York, 1979.

8. Steward, D. V., Systems Analysis and Management, Structure, Strategy and Design, Petrocelli Books, Inc., New York, 1981.

9. Rogers, J. L., DeMAID/GA User's Guide-Design Manager's Aid for Intelligent Decomposition With a Genetic Algorithm, TM-110241, April 1996.

10 Rogers, J. L., “An Intelligent Advisor for the Design Manager,” NASA TM-101558, 1989.

11. Bloebaum, C. L., "An Intelligent Decomposition Approach for Coupled Engineering Systems,” Fourth AIAA/AF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, 1992

12. Syswerda, G., “Schedule Optimization Using Genetic Algorithms.” Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.

13. Goldberg, D., Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Co., New York, 1989.

14. Altus, S. S., Kroo, I. M., and Gage, P. J. , “A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems,” ASME Paper 95-141, 1995.

15. McCulley, C. M., and Bloebaum, C. L., “Optimal Sequencing for Complex Engineering Systems Using Genetic Algorithms,” Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, 1994.

16. McCulley, C. M., and Bloebaum, C. L., “A Genetic Tool for Optimal Design Sequencing in Complex Engineering Systems,” To appear in Structural Optimization, 1996.

17. Smith, R. P., and Eppinger, S. D., “Identifying Controlling Features of Engineering Design Iteration,” Management Science, 1996.