Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1526041, 14 pages
https://doi.org/10.1155/2017/1526041

Research Article

A Method for Multidisciplinary System Analysis Based on Minimal Feedback Variables

Qian Yin, Yizhong Wu, and Boxing Wang

National CAD Supported Software Engineering Centre, Huazhong University of Science and Technology, Wuhan, China

Correspondence should be addressed to Boxing Wang; boxingwang@hust.edu.cn

Received 7 July 2017; Revised 20 November 2017; Accepted 28 November 2017; Published 21 December 2017

Academic Editor: Carlo Cosentino

Copyright © 2017 Qian Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As modern engineering design usually involves dependence of one discipline on another, multidisciplinary system analysis (MDSA) plays an important role in the multidisciplinary simulation and design optimization on coupled systems. The paper proposes an MDSA method based on minimal feedback variables (MDSA_MF) to enhance the solving efficiency. There are two phases in the method. In phase 1, design structural matrix (DSM) is introduced to represent a coupled system, and each off-diagonal element is denoted by a coupling variable set; then an optimal sequence model is built to obtain a reordered DSM with minimal number of feedback variables. In phase 2, the feedback in the reordered DSM is broken, so that the coupled system is transformed into one directed acyclic graph; then, regarding the inputs depending on the broken feedback as independent variables, a least-squares problem is constructed to minimize the residuals of these independents and corresponding outputs to zero, which means the multidisciplinary consistence is achieved. Besides, the MDSA_MF method is implemented in a multidisciplinary platform called FlowComputer. Several examples of coupled systems are modeled and solved in the platform using several MDSA methods. The results demonstrate that the proposed method could enhance the solving efficiency of coupled systems.

1. Introduction

Engineering design generally involves multidisciplinary dependence relationships of one discipline on another. For coupled system, these dependent relationships among the disciplines make up one or more loops. Thus, multidisciplinary system analysis (MDSA) is required to achieve the output-input consistence of all the dependence relationships by iteratively executing discipline analyses. Accordingly, an optimization on a coupled system using general nonlinear optimization methods could be time-consuming.

To enhance the solving efficiency, various multidisciplinary design optimization (MDO) frameworks [1–3] are proposed to handle the discipline couplings by decomposition and coordination strategies. Some MDO frameworks, for example, individual discipline feasible (IDF) [4] and collaborative optimization (CO) [5], eliminate the discipline couplings and enforce the multidisciplinary consistence at the final solution. These methods, however, could not obtain a multidisciplinary feasible solution when the optimization is interrupted. Other types of MDO frameworks, for example, multidisciplinary feasible (MDF) [4], concurrent subspace optimization (CSSO) [6], and bilevel integrated system synthesis (BLISS) [7], try to reduce the number of MDSA processes using different strategies while ensuring the multidisciplinary feasibility during the whole optimization process. Thus, the solving efficiency of MDSA could be essential to enhance the MDO process on coupled systems. Furthermore, different multidisciplinary analysis and optimization strategies under uncertainty are developed to handle the stochastic and/or epistemic uncertainties in coupled engineering problems [8–10]. A likelihood-based approach is proposed to estimate the probability density function of coupling variables [11] and is further extended to handle the model uncertainty [12] and the uncertainty propagation in high dimensional coupled systems [13]. Gibbs sampling and sequential importance resampling techniques are introduced to reduce the computational cost for decoupled multidisciplinary uncertainty analysis [14, 15]. These MDSA methods under uncertainty generally guarantee statistical multidisciplinary consistence, rather functional consistence. The present paper is focused on enhancing the solving efficiency of deterministic MDSA.

Several methods can be used to perform MDSA for coupled systems. Fixed point iteration (FPI) is a common-used method for MDSA [16]. When certain convergence condition is satisfied, a multidisciplinary feasible solution could be obtained. However, the FPI method converges slowly, which could lead to numerous discipline simulations [17, 18]. Newton-like methods [16] could achieve rapid convergence in that derivative information is used. The methods might fail to converge when solving from a bad starting point. Nonlinear least-squares (NLS) method [19] could be regarded as a generic MDSA method. It constructs a least-squares problem to minimize the sum of residuals of coupling relationships to zero and to obtain a multidisciplinary feasible solution. The constructed least-squares problem for MDSA can be solved flexibly by related NLS methods, or other general optimization algorithms.

Various engineering design platforms are developed to provide integrated multidisciplinary design environments. These platforms could integrate discipline design tools, define coupled system models, and perform multidisciplinary system analysis and design optimization on engineering problems [20, 21]. Commercial software tools, for example, ModelCenter [22], iSIGHT [23], and VisualDOC [24], are mainly focused on the integration of discipline tools and the capability of diverse design exploration methods [25, 26]. Some simple MDO frameworks, for example, IDF [4] and CO [5], can be implemented directly based on optimizer-like components and wrapped analysis components within some commercial tools [23, 27]. Several open source platforms, for example, DAKOTA [28], pyMDO [29], and openMDAO [30], could support automatic implementation of several MDO frameworks and their variants from specific problem descriptions [31, 32]. Some MDSA methods, or generic MDSA solvers, are provided in some of the platforms. ModelCenter provides a *Converger* component based on FPI method to achieve the convergence between the guessed variables and the calculated variables. The Gauss-Seidel iteration is the default algorithm to perform MDSA in pyMDO [29]. Within OpenMDAO, *BroydenSolver* and *FixedPointIterator* are provided to perform the iterative system analysis [31]. Within other MDO platforms, however, implementation of MDSA is generally provided by users. Furthermore, with the number of coupling variables increasing, the MDSA could be too large to be solved efficiently. These generic solvers might have difficulties in performing MDSA on the coupled systems with large number of couplings.

The paper proposed an MDSA method based on minimal feedback variables (MDSA_MF) to enhance the solving efficiency. The method includes two phases. In phase 1, design structural matrix (DSM) is introduced to represent a coupled system, and each off-diagonal element is denoted by a coupling variable set mapping from one discipline into another. Then, an optimal discipline sequence model is constructed to minimize the number of feedback variables by reordering the discipline sequence, and obtain a reordered DSM with minimal feedback variables. In phase 2, the feedback in the lower triangle of the reordered DSM is broken, so that the coupled system is transformed into a directed acyclic graph in terms of graph theory. Then, regarding the input variables depending on the broken feedback as independent variables, a least-squares problem with respect to these new independent inputs is constructed to minimize the sum of residuals of the independents and the corresponding outputs. When the objective of the least-squares problem is minimized to zero, the multidisciplinary consistence of the broken couplings is achieved. Besides, the implementation of the MDSA_MF method in a multidisciplinary design platform, called FlowComputer, is presented. Discipline integration based on Commercial-off-the-shelf (COTS) is provided to integrate discipline components, and a graphical user interface with dragging-and-dropping operations and visual data displayed is presented.

The rest of the paper is organized as follows. The next section lists the general MDSA methods used in the paper. Section 3 describes the DSM representation of coupled systems, proposes an optimal discipline sequence model to minimize the number of feedback variables, and presents the procedure of the MDSA_MF method. Section 4 describes the implementation of the MDSA_MF method in a multidisciplinary design platform. In Section 5, test cases of coupled systems are implemented in FlowComputer, and numerical results are investigated. Conclusions and future work are presented in the final section.

2. Related Methods of Multidisciplinary System Analysis

A large engineering design system usually involves a series of disciplines depending on one another. Such a multidisciplinary system can be generally stated as formulation (1).

$$\begin{aligned} y_1 &= y_1\left(\mathbf{X}, y_2, y_3, \ldots, y_n\right) \\ y_2 &= y_2\left(\mathbf{X}, y_1, y_3, \ldots, y_n\right) \\ &\vdots \\ y_n &= y_n\left(\mathbf{X}, y_1, y_2, \ldots, y_{n-1}\right), \end{aligned} \tag{1}$$

where n is the number of disciplines in the multidisciplinary system, $\mathbf{X}$ is the independent input vector, and y_i is the output variable of the ith discipline. The equation $y_i = y_i(\mathbf{X}, y_j)$ represents the ith discipline in the coupled system, where the notation $\mathbf{X}$ represents the vector of independent input variables $[x_1, x_2, \ldots, x_m]^{\mathrm{T}}$; y_j is the input variable depending on the jth discipline and is usually called coupling variable.

As the disciplines are dependent on one another, one or more execution loops exist. For a nonlinear system, the multidisciplinary consistence of coupling variables could not be satisfied if all of the disciplines are executed only once. Therefore, some iterative system analysis process is required. In this section, several iterative methods for MDSA are described.

2.1. Fixed Point Iteration. Fixed point iteration (FPI) method uses the original equations of the system as the iterative functions from a starting point of coupling variables [16]. Jacobi iteration and Gauss-Seidel iteration are the typical FPI methods. The former uses the values of the coupling variables from previous iteration to evaluate the outputs, and the disciplines could be run in parallel. The latter uses the recent evaluated values of other disciplines from current iteration as much as possible, and the disciplines are executed sequentially [33]. In most cases, the Gauss-Seidel iteration could converge faster than Jacobi iteration, for the newly updated values from current iteration might be more near to the solution. Formulations (2) and (3) state the iterative equations of Jacobi iteration and Gauss-Seidel iteration, respectively.

$$\mathbf{y}^{(k+1)} = y\left(\mathbf{X}, \mathbf{y}^{(k)}\right), \tag{2}$$

where $\mathbf{X}$ is the independent input vector, $\mathbf{y}^{(k)} = [{y_1}^{(k)}, {y_2}^{(k)}, \ldots, {y_n}^{(k)}]^{\mathrm{T}}$ is the output vector of the previous iteration, and $\mathbf{y}^{(k+1)} = [{y_1}^{(k+1)}, {y_2}^{(k+1)}, \ldots, {y_n}^{(k+1)}]^{\mathrm{T}}$ is the output vector of the current iteration.

$$\begin{aligned} {y_1}^{(k+1)} &= y_1\left(\mathbf{X}, {y_2}^{(k)}, {y_3}^{(k)}, \ldots, {y_n}^{(k)}\right) \\ {y_2}^{(k+1)} &= y_2\left(\mathbf{X}, {y_1}^{(k+1)}, {y_3}^{(k)}, \ldots, {y_n}^{(k)}\right) \\ &\vdots \\ {y_n}^{(k+1)} &= y_n\left(\mathbf{X}, {y_1}^{(k+1)}, {y_2}^{(k+1)}, \ldots, {y_{n-1}}^{(k+1)}\right), \end{aligned} \tag{3}$$

where $\mathbf{X}$ is the independent input vector, ${y_i}^{(k)}$ is the output of the ith discipline from the previous iteration, and ${y_i}^{(k+1)}$ is the output of the ith discipline during the current iteration.

2.2. Newton-Like Method. Newton-like methods convert the original coupled system into its residual form as formulation (4) and determine the next iterative point using the residual values and the corresponding derivative from the current point [16].

$$\begin{aligned} &R_1\left(y_1, y_1\left(\mathbf{X}, y_2, y_3, \ldots, y_n\right)\right) = 0, \\ &R_2\left(y_2, y_2\left(\mathbf{X}, y_1, y_3, \ldots, y_n\right)\right) = 0, \\ &\vdots \\ &R_n\left(y_n, y_n\left(\mathbf{X}, y_1, y_2, \ldots, y_{n-1}\right)\right) = 0, \end{aligned} \tag{4}$$

where R_i represents the residual form of the ith discipline, n is the number of disciplines, $\mathbf{X}$ is the independent input vector, and y_i represents the coupling variable.

The Newton-Raphson iterative equations [16] are presented as formulation (5).

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \left[J\left(\mathbf{X}, \mathbf{y}^{(k)}\right)\right]^{-1} R\left(\mathbf{X}, \mathbf{y}^{(k)}\right), \tag{5}$$

where $\mathbf{y}^{(k)}$ is the vector of coupling variables from the previous iteration, $\mathbf{y}^{(k+1)}$ is the output vector of coupling variables of the current iteration, and $J(\mathbf{X}, \mathbf{y}^{(k)})$ is the Jacobi matrix of the discipline residuals to the coupling variables during the previous iteration.

2.3. Nonlinear Least-Squares Methods. Nonlinear least-squares (NLS) methods [19] break the coupling relationships and construct a least-squares objective, which minimizes the sum of squares of the residuals of the broken couplings, to find a multidisciplinary feasible solution. The least-squares problem is as formulation (6):

$$\begin{aligned} &\text{Find} \quad {y_i}^t \\ &\min \quad \sum_{i=1}^{n} \left\| {y_i}^t - y_i \right\|_2^2, \end{aligned} \tag{6}$$

where n represents the number of coupling variables, y_i is the output variable of the ith discipline, and ${y_i}^t$ is the unknown design variable, corresponding to the input variable of a broken coupling relation depending on y_i. Here, y_i is determined by formulation (7).

$$y_i = y_i\left(\mathbf{X}, {y_j}^t\right), \quad j \neq i. \tag{7}$$

The NLS algorithms, or other optimization algorithms, could be used to solve the least-squares problem, which makes the multidisciplinary problem more flexible to be solved. As each least-squares term is constructed with respect to a coupling variable, the number of unknown design variables is equal to the number of the least-squares terms.

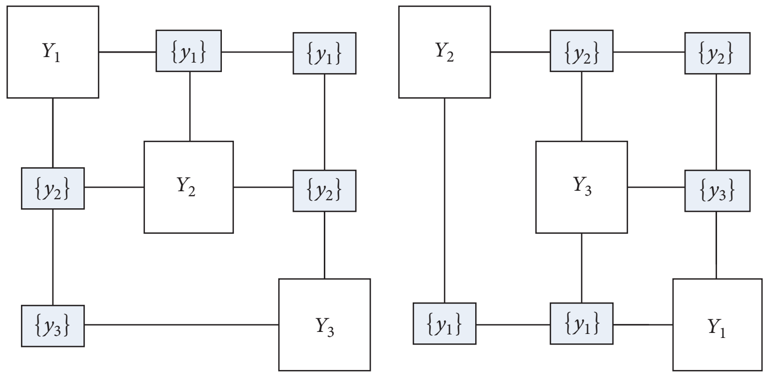
3. The Framework of MDSA_MF

3.1. Discipline Dependence Representation of Coupled Systems. Design Structure Matrix (DSM) [34] is usually used to represent the dependence of one discipline on another in a coupled system. In the matrix, diagonal elements represent the disciplines, which might be analytical functions, specific disciplines, subsystems, components, black-boxes, or other objects. Each element in upper triangle represents a feed-forward coupling relationship between associated disciplines, and each one in lower triangle represents a feedback. Because the DSM representation is the adjacency matrix of the discipline dependence graph, a coupled system is generally a directed cyclic graph. Formulation (8) shows the general matrix representation.

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix}, \tag{8}$$

where n represents the number of disciplines, Y_{ii} represents the ith discipline, and Y_{ij} represents the dependence relationship of the jth discipline on the ith discipline.

The coupling relationships represented by the off-diagonal elements can be different expressions. A Boolean value, that is, “1” or “0,” can represent whether one discipline depends on another [35, 36]. The Boolean DSM could also be employed to model and solve Boolean Dynamical



(a) The initial DSM (b) The reordered DSM

Figure 1: The DSM representations of an example system.

Systems [37, 38]. Derivative information could quantitatively indicate the influence of one discipline on another at a given point [39, 40]. And a natural number can represent the number of variables mapping from one discipline into another [41].

In the present paper, each off-diagonal element is represented by a collection of variables mapping from an output of one discipline into an input of another discipline. To simplify the collection, a set of output variables is often used. This representation can be converted into a Boolean value, or the number of feedback variables. Also, the representation can be extended to include other information about the corresponding coupling.

3.2. The Optimal Discipline Sequence Model. In engineering designs, the MDSA problem could be too large to be solved efficiently when the number of coupling variables is large. In this case, a part of the couplings, for example, the feedback couplings, could be selected to construct a least-squares problem as Section 2.3 to implement the MDSA. Accordingly, the selected couplings are broken and the coupled system is transformed into one without feedback couplings. In terms of graph theory, one directed acyclic graph of the system is obtained.

The size of the least-squares problem depends on the number of selected feedback variables. With different sequence of diagonal elements, the feedback couplings in the lower triangle could be different. Several DSM-based optimization methods are proposed to reorder the discipline sequence [39, 42–44]. These methods try to minimize the number of feedback coupling loops [45, 46], or minimize an integrated objective taking other factors, for example, time, cost, and modularity, into account [43, 47]. Partitioning a coupled system into several small subsystems is another objective to reduce the complexity of the problems [41, 44].

The paper is focused on minimizing the number of feedback variables to reduce the MDSA problem size. The objective is to reduce the number of all the feedback variables in the lower triangle of DSM. Each off-diagonal element of the DSM is represented by a variable set consisting of feed-forward or feedback variables between two disciplines. Figure 1 shows the DSM representation of an example system with three disciplines. With the initial DSM as Figure 1(a), there are two feedback variables, y_2 and y_3. With the reordered DSM as Figure 1(b), there is only one feedback variable, y_1.

Figure 2 shows the Boolean DSM representation of the aforementioned example. Both the initial Boolean DSM and the reordered Boolean DSM have two feedback marks. However, less feedback variables could be selected with the reordered DSM.

To minimize the number of feedback variables in the lower triangle, an optimal sequence problem is stated as formulation (9).

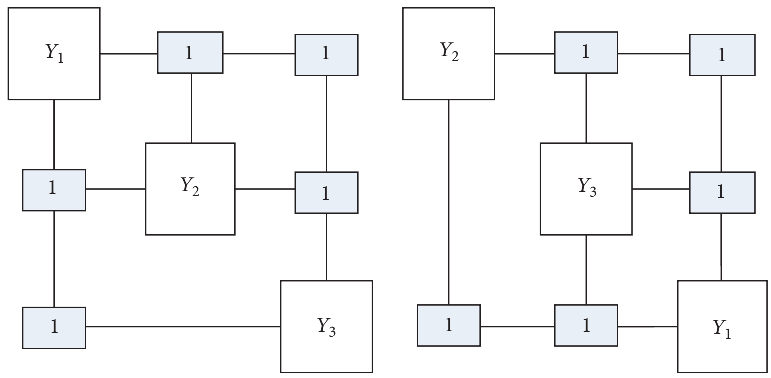
$$\min \quad \left| \bigcup_{i=2}^{n} \bigcup_{j=1}^{i-1} Y_{ij} \right|, \tag{9}$$

where n is the number of disciplines in a complex system and Y_{ij} is a feedback variable set mapping from discipline i into discipline j. $\bigcup_{i=2}^{n} \bigcup_{j=1}^{i-1} Y_{ij}$ represents the union set of all the variable sets in lower triangle, and $|Y|$ denotes the number of elements in set Y.

3.3. MDSA_MF Procedure. The optimal discipline sequence model described in previous subsection is used to obtain a reordered DSM with the minimal number of feedback variables. The feedback variables in the lower triangle are selected to construct a least-squares problem to minimize the sum of residuals of the feedback couplings to zero and achieve the multidisciplinary consistence. The problem is stated as follows.

$$\begin{aligned} &\text{Find} \quad y_i^{s} \\ &\min \quad \sum_{i=1}^{n} \left\| y_i^{s} - y_i \right\|_2^2, \end{aligned} \tag{10}$$

where n represents the number of the selected coupling variables, y_i is the output variable of the ith discipline, and y_i^{s}



(a) The initial Boolean DSM (b) The reordered Boolean DSM

Figure 2: The Boolean DSM representations of an example system.

is the unknown design variable, passed on to corresponding discipline input with respect to the broken coupling.

As the feedback is removed, the disciplines are executed sequentially. The outputs with respect to the selected feedback are determined by formulation (11).

$$y_i = y_i\left(X, {y_j}^s, y_k\right), \quad j > i,\ k < i, \tag{11}$$

where ${y_j}^s$ is the selected design variable, y_i is the output of the ith discipline, and y_k is the output of the kth discipline ahead of the ith discipline in the execution sequence of disciplines.

In addition, some coupled systems could be divided into several strongly connected components. Each strongly connected component is a directed cyclic subgraph of the system. However, there is no interdependence relationship between any two strongly connected components. Thus, the system denoting each strongly connected component as a block is a directed acyclic graph, and the strongly connected components can be solved sequentially. The method of searching strongly connected component could aid in the discipline ordering [48].

Figure 3(a) shows the initial DSM representation of an example coupled system. The system can be divided into two strongly connected components and four individual nodes. Figure 3(b) shows the reordered DSM. A discipline can be regarded as a strongly connected component if the discipline is not coupled to other disciplines.

The procedure of MDSA_MF is presented as follows.

Step 1. Search strongly connected components, and divide the system into a series of strongly connected components. Tarjan depth first search algorithm [49] is used to search strongly connected components.

Step 2. Topologically order the strongly connected components to generate the solving sequence.

Step 3. For each strongly connected component, reorder the discipline sequence to obtain a sub-DSM with minimal number of feedback variables by formulation (9).

Step 4. Solve the strongly connected components sequentially, and if some components do not depend on each other, they could be executed in parallel. For each strongly connected component, it is solved as follows.

Step 4.1. Identify the feedback variables, break the feedback, and generate evaluation functions using the subsystem without feedback.

Step 4.2. Construct a least-squares objective with respect to the residuals of broken couplings.

Step 4.3. Initialize the inputs of broken couplings with given starting values, or the current values of the corresponding outputs.

Step 4.4. Solve the least-squares problem as formulation (10).

Step 4.5. Pass on the output values to the corresponding inputs of the next subsystem, and turn to *Step 4.1.*

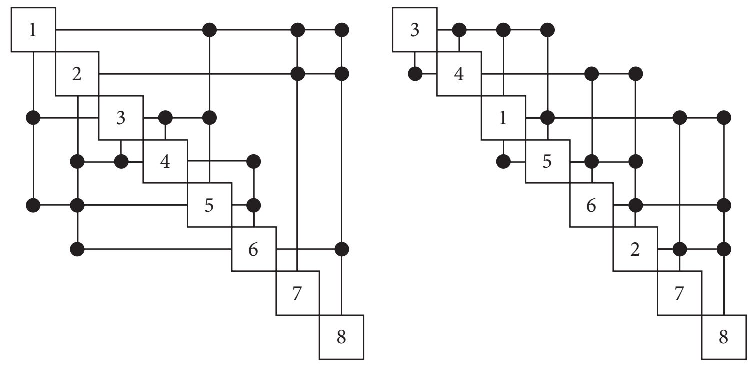
Step 5. Complete solving, and present outputs.

4. System Design and Implementation

In the section, the implementation of MDSA_MF in a visual and intuitive multidisciplinary platform, called FlowComputer, is presented to integrate different discipline models; define the dependence relationships and solve coupled systems.

Component objects and link objects are developed to define data and process models of coupled systems. A component object represents a discipline, or a data processing node, and executes some discipline analysis or computes a series of outputs from given inputs. A link object describes the variable mappings from one component to another. And the proposed MDSA_MF method is implemented to solve the coupled system modeled in FlowComputer.

4.1. Discipline Integration Based on COTS Wrapping. Commercial-off-the-shelf (COTS) wrapping techniques are developed to integrate commercial software tools, or legacy codes. One of the major COTS wrapping approaches to discipline integration is the In-Process-Out (IPO) method, in which input files (I), process program (P), and output files (O) are



(a) The initial DSM (b) The reordered DSM

Figure 3: An example DSM of a coupled system.

used to integrate discipline tools. Input variables and output variables are stored in the input files and the output files, respectively. Process program is typically a discipline analysis tool, or a BAT file including a serial of discipline tool command lines, which reads values of input variables from input files, executes the corresponding analysis, and writes values of outputs into output files. Then, FlowComputer is able to integrate discipline simulation codes by wrapping input variables and output variables from input files and output files and specifying process programs and other supporting files. Currently, most discipline analysis tools, like *Ansys*, *Nastran*, *Adams*, *Abqus*, *Fluent*, *Ansoft*, and so on, can be integrated by this approach. And a *Generic Wrapper* component is provided to integrate various discipline analysis tools, especially the legacy simulation codes.

For some cases that input and output variables are embedded in discipline models; another approach, plug-in method, is used to extract input and output variables from the model file of a third-party software tool by its API interface. Similarly, FlowComputer sets inputs to the model file, updates the model, and extracts outputs by the API interface. Now, discipline tools, including *MATLAB*, *Pro/E*, *CATIA*, and *Excel*, are integrated using this approach.

In addition, *Expression* component is also provided to compute a set of explicit expressions from given inputs.

4.2. Feedback Based Representation of Couplings. Link objects are designed to represent the dependence of one discipline on another. A link object consists of a source component, a destination component, and a set of variable mappings from the former to the latter. Thus, the couplings in a multidisciplinary system could be presented as a collection of link objects.

To represent the coupling relationships, a type of feedback links is introduced to facilitate the modeling and solving of coupled systems. During the process of creating variable links, if a dependence loop is detected, the last constructed link is marked as a feedback link. DSM representation could intuitively exhibit the coupling relationships in a complex system. In the present study, the collection of link objects is used to represent the couplings, and DSM representation is used to intuitively exhibit the dependence relationships. Each link object represents a dependence relationship corresponding to an off-diagonal element of the DSM representation.

4.3. MDSA Implementation. Searching algorithm of strongly connected components and discipline reordering algorithm are used to analyze coupled systems, and all the iterative methods for MDSA described in Section 2 are available to solve coupled systems, or strongly connected components. By default, the MDSA_MF method is used to construct the least-squares objective, and the *hybrd* solver from Minpack package [50] is selected to solve the least-squares problem.

4.4. Introduction to the User Interface. The main user interface of FlowComputer, shown as Figure 4, is composed of flow view, components tree view, components class view, and components list view. According to a selected component class in the component class view, available components are displayed in the component list view. The components can be dragged and dropped into the flow view to integrate discipline models, and the relationship between two components can be defined by dragging-and-dropping operations. All components are displayed as a tree in the components tree view. Once a coupled system model is well defined, it can be executed automatically and monitored visually, and results can be shown graphically.

Dependence relationships between any two components could be constructed visually. Mapping viewer shown as Figure 5 is provided to define, edit, and display variable mappings between any two disciplines. Two component trees are listed on the left and the right of the mapping viewer, and link lines indicating mappings are drawn from the variables on the left tree to input variables on the right tree. Users can drag-and-drop any variable on the left tree onto an input variable on the right tree to define a map from the former variable to the latter. The default map relation is that the right variable is equal to the left one.

The DSM representations of a coupled system in FlowComputer are shown as Figure 6. There are nine disciplines, represented by nine blocks on the diagonal. The solid connection lines in the upper triangle represent the feed-forward couplings, and the dashed lines in the lower triangle represent the feedback.

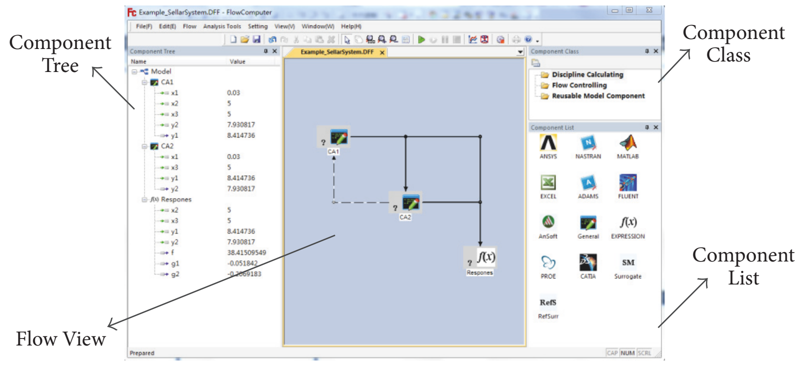


FIGURE 4: The main user interface of FlowComputer.

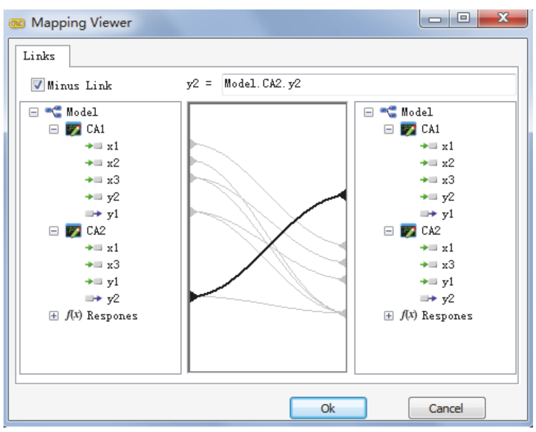


FIGURE 5: The mapping viewer for dependence relationships.

5. Case Studies

In this section, two coupled systems are modeled and solved in FlowComputer. The MDSA method based on all couplings (MDSA_AC), the MDSA method based on initial feedback variables (MDSA_IF), and the MDSA_MF method are investigated. Different iterative algorithms, including FPI, the Newton method, and the *hybrd* solver, are employed within each MDSA method. The number of function calls to disciplines is recorded to evaluate the solving efficiency, and the 2-norm of discipline residuals to the final solution, as formulation (12), is used to evaluate the accuracy of different methods.

$$r = \left\| \mathbf{y}^* - y\left(\mathbf{X}, \mathbf{y}^*\right) \right\|_2, \tag{12}$$

where r is the 2-norm of discipline residuals, $\mathbf{X}$ represents the independent input variable vector with given values, and $\mathbf{y}^*$ represents the final solution of the coupling variables.

5.1. Test Case 1. The test case is modified from the scalable problem [51] and is stated as follows:

$$y_1 = \frac{-\left(9.7160z + 1.3627x_1 - 11.008\left(y_3\right)^{1/3} - 8.2846y_{16}\right)}{6.4976},$$

$$y_2 = \frac{-\left(1.5717z + 7.8294x_2 - 72.238\left(y_8\right)^{1/3} - 5.4647y_4\right)}{17.0438},$$

$$y_3 = \frac{-\left(2.0489z + 4.5625x_3 - 67.350\left(y_{18}\right)^{1/3} - 7.8362y_{20}\right)}{2.2016},$$

$$y_4 = \frac{-\left(1.9935z + 9.3701x_4 - 43.968\left(y_9\right)^{1/3} - 6.9368y_6\right)}{10.9294},$$

$$y_5 = \frac{-\left(8.2844z + 3.2383x_5 - 11.369\left(y_{10}\right)^{1/3} - 9.2195y_{15}\right)}{9.7504},$$

$$y_6 = \frac{-\left(1.3453z + 1.8450x_6 - 35.783\left(y_{19}\right)^{1/3} - 6.1040y_{14}\right)}{13.8736},$$

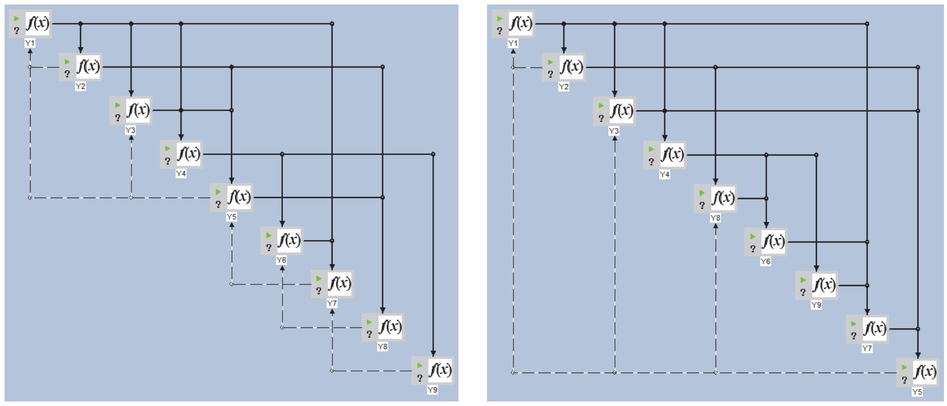
$$y_7 = \frac{-\left(2.4311z + 4.6650x_7 - 11.369\left(y_{10}\right)^{1/3} - 3.2488y_1\right)}{3.5614},$$

$$y_8 = \frac{-\left(2.5485z + 6.2020x_8 - 61.040\left(y_{14}\right)^{1/3} - 8.5219y_2\right)}{14.4476},$$

$$y_9 = \frac{-\left(1.3537z + 7.0370x_9 - 85.219\left(y_2\right)^{1/3} - 1.7807y_7\right)}{8.7936},$$

$$y_{10} = \frac{-\left(9.6129z + 1.8295x_{10} - 92.195\left(y_{15}\right)^{1/3} - 4.8752y_5\right)}{2.2738},$$

$$y_{11} = \frac{-\left(8.1890z + 5.1827x_{11} - 11.008\left(y_3\right)^{1/3} - 6.1040y_{14}\right)}{16.7888},$$



(a) The initial DSM (b) The reordered DSM

Figure 6: The screenshot of the DSM representation of a coupled system.

$$
\begin{aligned}
y_{12} &= \frac{-\left(1.5986z + 2.3169x_{12} - 28.841\left(y_{17}\right)^{1/3} - 9.2195y_{15}\right)}{4.7212}, \\
y_{13} &= \frac{-\left(6.1798z + 1.9304x_{13} - 69.368\left(y_{6}\right)^{1/3} - 6.7350y_{18}\right)}{8.9082}, \\
y_{14} &= \frac{-\left(3.5043z + 3.3825x_{14} - 23.606\left(y_{12}\right)^{1/3} - 4.4541y_{13}\right)}{12.2080}, \\
y_{15} &= \frac{-\left(7.9848z + 9.9501x_{15} - 48.752\left(y_{5}\right)^{1/3} - 2.3606y_{12}\right)}{18.4390}, \\
y_{16} &= \frac{-\left(1.3627z + 1.3782x_{16} - 78.362\left(y_{20}\right)^{1/3} - 8.3944y_{11}\right)}{16.5692}, \\
y_{17} &= \frac{-\left(3.7686z + 3.4848x_{17} - 85.219\left(y_{2}\right)^{1/3} - 6.7350y_{18}\right)}{5.7682}, \\
y_{18} &= \frac{-\left(4.9599z + 9.0051x_{18} - 54.647\left(y_{4}\right)^{1/3} - 4.3968y_{9}\right)}{13.4700}, \\
y_{19} &= \frac{-\left(8.6044z + 7.2126x_{19} - 44.541\left(y_{13}\right)^{1/3} - 7.2238y_{8}\right)}{7.1566}, \\
y_{20} &= \frac{-\left(7.7096z + 2.8943x_{20} - 69.368\left(y_{6}\right)^{1/3} - 2.3606y_{12}\right)}{15.6724},
\end{aligned}
\tag{13}
$$

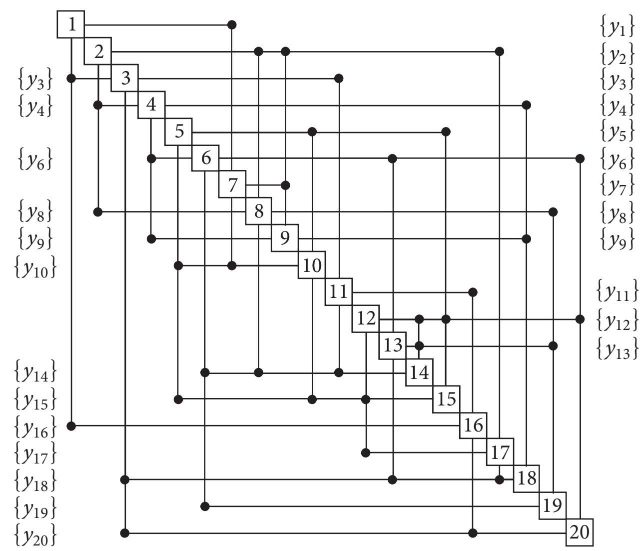


Figure 7: The initial DSM of the coupled system in test Case 1.

where x_i and z are the independent variables and y_i is the output of the ith discipline.

This coupled system with 20 disciplines has 21 independent inputs and 20 coupling variables. The DSM of the system is shown as Figure 7. The disciplines are coded with ascend digital numbers from 1 to 20. Each diagonal box with a number represents one discipline, and each black dot in the off-diagonal represents a coupling variable, marked on the left or on the right. The system is a strongly connected graph, and the reordered DSM with minimal feedback variables is shown as Figure 8.

Table 1 shows the solving results starting from the independent variables $z = 1$ and $x_i = 1$ and the coupling variables $y_i = 10$. Here, each discipline is treated as a black-box simulation. The gradients required by the Newton method and the *hybrd* solver are computed via forward difference method with a relative step size of 10^{-6}. The Newton method and the *hybrd* solver would terminate if the 2-norm of discipline

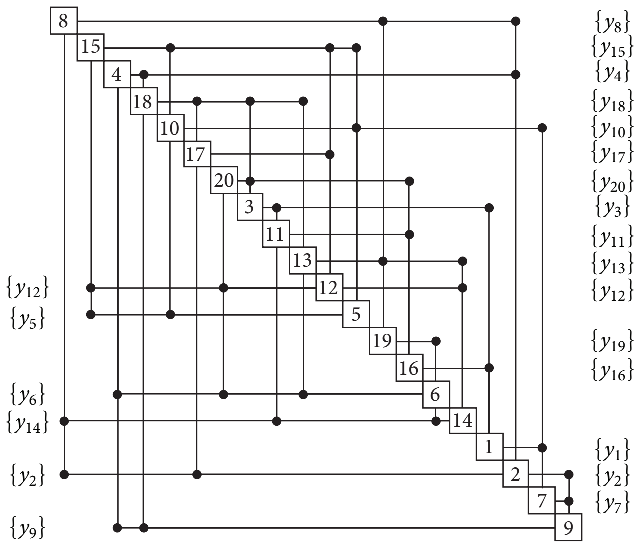


FIGURE 8: The reordered DSM with minimal feedback variables in test Case 1.

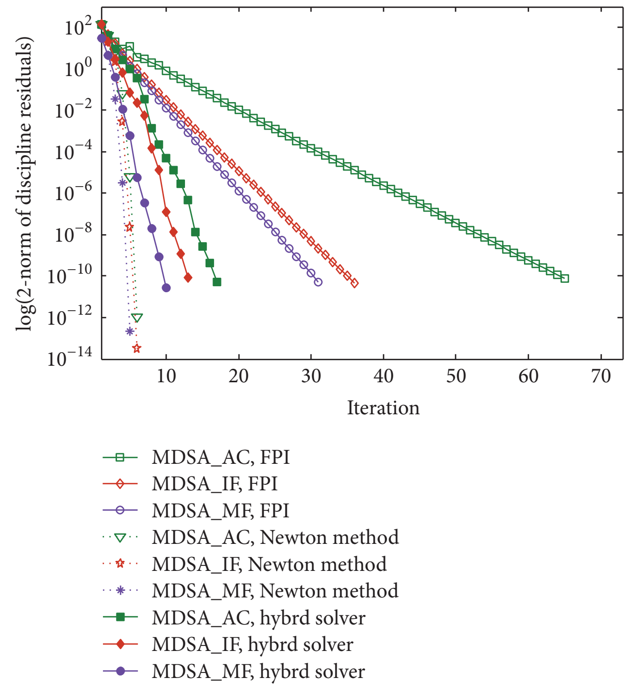


FIGURE 9: Convergence histories of the 2-norm of discipline residuals on test case 1.

residuals is less than 10^{-10}. For the FPI method, the termination criteria are satisfied when the 2-norm of the distance between consecutive solutions is less than 10^{-10}. Starting from other values of the independent variables and other initial values of the coupling variables, similar results are obtained.

The data in Table 1 indicate that the coupled system is successfully solved using all the MDSA methods. For a given iterative method, for example, the *hybrd* solver, the MDSA_MF uses the fewest function calls to disciplines, followed by the MDSA_IF. And for the MDSA method based on the same selected couplings, for example, MDSA_MF, the *hybrd* solver uses the fewest discipline evaluations. The coupled system has 20 coupling variables. As shown in Figures 7 and 8, 13 feedback variables, y_3, y_4, y_6, y_8, y_9, y_{10}, y_3, y_4, y_6, y_8, y_9, y_{10}, and y_9, are selected for MDSA_IF, and six coupling variables, y_2, y_5, y_6, y_9, y_{12}, and y_{14}, are selected for MDSA_MF. Sequentially, MDSA_MF uses less function calls to compute the derivative information. As the Jacobi matrix of the discipline residuals to coupling variables at each iterate is computed by the finite difference method, the Newton method uses more function calls to the disciplines.

Figure 9 shows the convergence histories of the 2-norm of discipline residuals. Similarly, for a given iterative method, for example, *hybrd* solver, MDSA_MF uses the fewest iterates, followed by MDSA_IF. The Newton methods converge to the final solution by the fastest speed, for derivative information is used at each iterate. However, the discipline evaluations increase when the finite difference method is used to compute derivative information.

5.2. Test Case 2. The second test case, from [39], is stated as formulation (14).

$$
\begin{aligned}
t_1 &= 23.521 \left(x_{11}\right)^2 + 45.122 \left(x_{12}\right)^{1/2} + 0.9352 \left(y_1\right)^{1/2} \\
&\quad - 0.2835 v_2, \\
t_2 &= 23.175 x_{12} + \frac{30.645}{x_{13}} - 0.5392 v_1 + 0.0024 \left(z_1\right)^2, \\
t_3 &= 12.944 x_{11} + \frac{43.846}{x_{12}} + 18.345 \left(x_{13}\right)^2 \\
&\quad - 0.0113 \left(v_2\right)^2 - 0.0021 \left(z_2\right)^2 - 0.1635 u_2, \\
t_4 &= 12.347 \left(x_{11}\right)^{2/3} + 21.478 x_{12} + 7.745 \left(x_{13}\right)^2 \\
&\quad + 0.563 \left(w_2\right)^{1/2} - 0.7256 v_4 + \frac{0.9476}{y_3}, \\
u_1 &= 8.435 \left(x_{21}\right)^2 + 12.563 x_{22} + 0.1577 z_1 \\
&\quad - 0.9388 \left(w_3\right)^{1/2}, \\
u_2 &= 24.547 x_{21} + \frac{24.356}{x_{22}} - \frac{1.457}{\left(t_1\right)^2} + 0.3846 t_3 \\
&\quad + \frac{0.9862}{\left(w_2\right)^{1/2}} - 0.7377 \left(z_3\right)^{1/2}, \\
u_3 &= 9.367 \left(x_{22}\right)^2 + 0.0285 \left(t_1\right)^{1/4} - \frac{0.6758}{\left(y_1\right)^{1/2}}, \\
v_1 &= 16.846 x_{31} + 10.360 \left(x_{32}\right)^2 - 0.3367 t_2 \\
&\quad + 0.157 \left(u_2\right)^{1/2} + 0.0054 z_1 - \frac{0.4783}{\left(w_2\right)^{1/2}},
\end{aligned}
$$

Table 1: The result data for test Case 1.

	Iterative method/solver	r	Evaluation number $D_1 \sim D_{20}$
MDSA_AC	FPI	4.95×10^{-11}	65
	Newton method	1.05×10^{-12}	106
	hybrd solver	4.94×10^{-11}	37
MDSA_IF	FPI	1.79×10^{-11}	36
	Newton method	3.08×10^{-14}	71
	hybrd solver	8.44×10^{-11}	26
MDSA_MF	FPI	1.07×10^{-11}	31
	Newton method	2.22×10^{-13}	29
	hybrd solver	2.80×10^{-11}	16

r represents the 2-norm of discipline residuals and D_i represents the ith discipline. The evaluation counts of different disciplines are the same.

$$\begin{aligned}
v_2 &= 23.644 (x_{32})^{1/2} + 14.466x_{33} - 0.0064 (t_1)^2 \\
&\quad + 0.5347 y_2, \\
v_3 &= 8.846 (x_{31})^2 + 44.467 (x_{33})^{1/2} + 0.3748 (u_3)^{1/2} \\
&\quad - 0.1094 w_1, \\
v_4 &= 14.536 (x_{32})^2 - 0.0949 y_1 + \frac{0.4503}{t_4} \\
&\quad + 0.1003 (w_3)^{1/2} + 0.0083 u_1 - \frac{0.0776}{(y_3)^{1/2}}, \\
w_1 &= 14.896 (x_{41})^3 + 13.746x_{42} + 0.4567 y_1 \\
&\quad - 0.0144 (y_2)^2 + \frac{0.0944}{z_2}, \\
w_2 &= 16.436 (x_{42})^2 - 0.0113 (z_1)^2 - 0.0673 z_2 \\
&\quad + \frac{0.0356}{z_3} + 0.0275 u_2, \\
w_3 &= 8.638x_{41} + \frac{26.693}{x_{42}} + 6.536 (x_{43})^2 - 0.6685 t_4 \\
&\quad - 0.8467 v_4, \\
y_1 &= 27.783x_{51} + 35.552x_{52} + 10.377 (x_{53})^2 \\
&\quad - 0.0059 (t_1)^2 + 0.0983 u_3 + 0.0633 (v_3)^{1/2}, \\
y_2 &= 15.367 (x_{51})^2 + 18.653x_{53} + 0.1079 u_3 + \frac{0.2763}{w_1} \\
&\quad - 0.831 v_2, \\
y_3 &= 22.561x_{52} + \frac{35.649}{x_{53}} + 0.6463 z_3 - \frac{0.7366}{t_4} \\
&\quad - 0.0667 u_1, \\
y_4 &= 9.882 (x_{51})^2 + 0.0011 (t_3)^2 - 0.0359 u_2 \\
&\quad + \frac{0.2033}{z_3}, \\
z_1 &= 8.629x_{61} + 7.932 (x_{62})^2 - 0.0274 t_2 - 0.0235 t_3 \\
&\quad + 0.0021 v_1, \\
z_2 &= 14.367 (x_{61})^2 + \frac{48.837}{x_{62}} + 0.0009 (v_2)^2 \\
&\quad - 0.0328 (w_1)^{1/2} - \frac{0.0825}{v_3}, \\
z_3 &= 29.663x_{62} - 0.0846 w_1 + 0.0024 (t_2)^{1/2} \\
&\quad + 0.0436 u_2 + \frac{0.0875}{w_2} - 0.0289 (y_4)^{1/2},
\end{aligned} \tag{14}$$

where x_{ij} represents the independent input and t_i, u_i, v_i, w_i, y_i, and z_i are the coupling variables.

The system has 21 disciplines and 21 coupling variables. The DSM of the system is shown as Figure 10. The disciplines are coded with ascend digital numbers from 1 to 21. The boxes on the diagonal and the black dots on the off-diagonal denote the same as Figure 7.

The coupled system could be divided into three strongly connected components, and the reordered DSM is shown as Figure 11. The strongly connected components are denoted by *Group A*, *Group B*, and *Group C*, respectively. Within each strongly connected component, the discipline sequence with minimal number of feedback variables is presented.

Within FlowComputer, the coupled system is successfully solved using several methods starting from some given values of the independent variables and different initial values of the coupling variables. This coupled system could be solved by constructing an MDSA problem with 21 disciplines and could also be solved by constructing three sequential subproblems

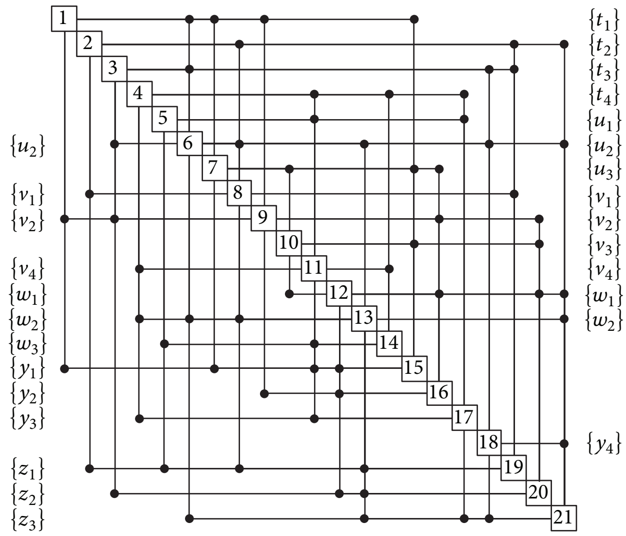


Figure 10: The initial DSM of the coupled system in case 2.

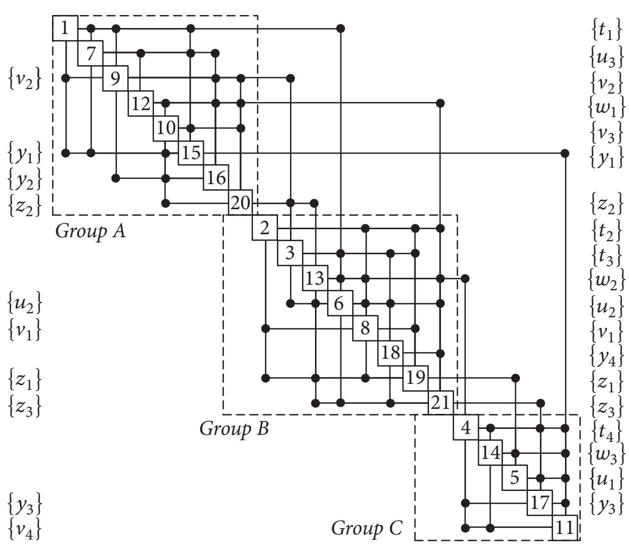


Figure 11: The reordered DSM of the system in case 2 with minimal feedback variables.

corresponding to the three strongly connected components in Figure 11. Both of the two solving methods are investigated. The settings for algorithms, the step size of finite differences, and the termination criteria are the same as the first test case.

Tables 2 and 3 show the results starting from the independent variables $x_{ij} = 1$ and the coupling variables $(t_i, u_j, v_k, w_l, y_m, z_n) = (10, 10, 10, 10, 10, 10)$. The evaluation number of the three groups of disciplines and the 2-norm of discipline residuals at the final solution are presented. Solving from other initial values of coupling variables presents similar data. However, solving from some given values of the independent variables far from the point $x_{ij} = 1$ might fail in finding a multidisciplinary solution.

The data listed in Tables 2 and 3 show that the coupled system is successfully solved with good accuracy using all of the methods. For each iterative method, MDSA_MF uses

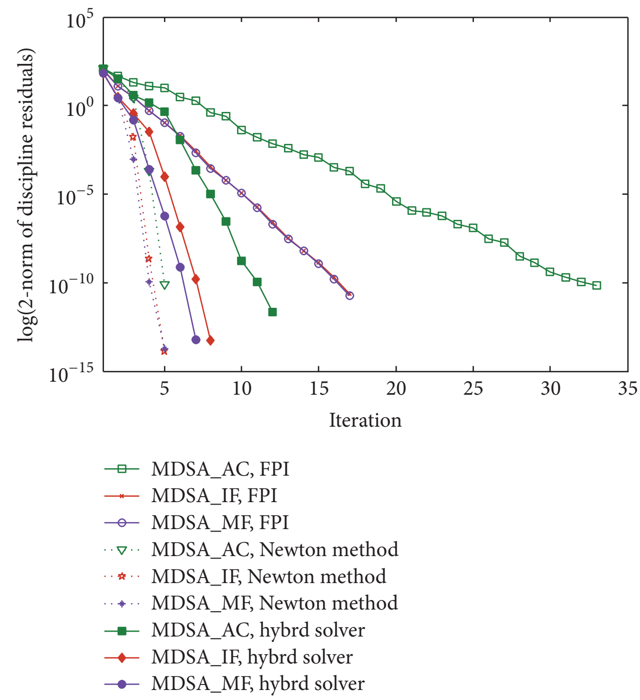


Figure 12: Convergence histories of the Group A in test case 2.

the fewest discipline evaluations, followed by the MDSA_IF method. If the system is solved as a whole coupled unit as in Table 2, the FPI method uses fewer function calls when the initial feedback, or the minimal feedback, is selected to be broken. However, the *hybrd* solver uses the fewest function calls when the three strongly connected components are solved sequentially. The possible reason is that the strategy solving the strongly connected components sequentially reduces the MDSA size. The strategy, however, does not influence the efficiency of Gauss-Seidel iteration significantly. Hence, MDSA_MF could use the fewest function calls, and solving the strongly connected components sequentially could further enhance the MDSA efficiency.

Figure 12 shows the convergence histories of the 2-norm of discipline residuals for solving the subsystem, *Group A*. The histories of solving the other two subsystems present the similar behavior. The convergence data indicate that MDSA_MF could use the fewest iterates for a given iterative method. The FPI iteration shows slow convergence. As derivative information is updated at each iterate, the Newton iterations converge to the final solution by fastest speed. However, computing the derivative information using the finite difference method increases the function calls. When the derivative information is easy to be obtained, the Newton iteration could generally converge to the final solution as the fastest speed.

6. Conclusions

The paper proposes a two-phase MDSA method based on minimal number of feedback variables, called MDSA_MF, to enhance the solving efficiency. In phase 1, DSM is introduced to represent a coupled system, and each off-diagonal element

Table 2: The result data for test case 2 solving as a whole coupled system.

	Iterative method/solver	r	Evaluation number $D_1 \sim D_{21}$
MDSA_AC	FPI	3.00×10^{-11}	33
	Newton method	2.70×10^{-14}	111
	hybrd solver	1.03×10^{-11}	33
MDSA_IF	FPI	5.61×10^{-12}	17
	Newton method	2.02×10^{-14}	57
	hybrd solver	1.57×10^{-12}	21
MDSA_MF	FPI	4.01×10^{-12}	17
	Newton method	2.32×10^{-14}	45
	hybrd solver	1.56×10^{-12}	18

r represents the 2-norm of discipline residuals and D_i represents the ith discipline. The evaluation counts of different disciplines are the same.

Table 3: The result data for test case 2 solving the three strong components sequentially.

	Iterative method/solver	r	Evaluation number		
			Group A	Group B	Group C
MDSA_AC	FPI	5.01×10^{-11}	33	32	20
	Newton method	7.24×10^{-11}	37	37	19
	hybrd solver	3.94×10^{-11}	20	15	11
MDSA_IF	FPI	6.35×10^{-12}	17	16	12
	Newton method	3.65×10^{-14}	25	19	13
	hybrd solver	1.04×10^{-11}	13	11	8
MDSA_MF	FPI	6.51×10^{-12}	17	16	7
	Newton method	8.33×10^{-15}	21	16	10
	hybrd solver	8.31×10^{-12}	11	9	6

r represents the 2-norm of discipline residuals.

of the DSM is denoted by a coupling variable set mapping from one discipline into another. An optimal discipline sequence problem is constructed to obtain a reordered DSM with minimal number of feedback variables in the lower triangle. In phase 2, the feedback in the lower triangle is broken, and the coupled system is transformed into a directed acyclic graph. Then, regarding the inputs depending on the broken feedback as independent variables, a least-squares problem with respect to these new independent variables is constructed to minimize the sum of residuals of the broken feedback to zero, and to further achieve a multidisciplinary feasible solution. Searching strongly connected components is also used to aid in the discipline reordering. Besides, the MDSA_MF method is implemented in a multidisciplinary design platform, called FlowComputer. The platform also provides the capacity of discipline integration based on COTS wrapping and modeling and solving GUI for coupled systems.

Two test cases of coupled systems are modeled in FlowComputer, and several MDSA methods using different iterative method are investigated. The results demonstrate that MDSA_MF could use the fewest function calls. And the strategy dividing the system into several strongly connected components could further enhance the efficiency.

The MDSA_MF selects the minimal number of feedback variables as unknown variables to solve coupled systems. Thus, the disciplines are executed sequentially, and parallel computing is not considered in the present work. Besides, the paper is focused on deterministic multidisciplinary analysis and does not include the uncertainty factors in engineering problems. The ongoing and future work includes (a) employing the available parallel computer resources to improve the MDSA efficiency and (b) implementing the MDSA_MF method on engineering problems under uncertainty.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is supported by the National Science Foundation of China [Grant no. 51575205] and the Manufacturing Industry Product Innovation Knowledge Service Platform's Development and Application [Grant no. 2013AA040603].

References

[1] J. Sobieszczanski-Sobieski and R. T. Haftka, "Multidisciplinary aerospace design optimization: Survey of recent developments," *Journal of Structural Optimization*, vol. 14, no. 1, pp. 1–23, 1997.

[2] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[3] L. Brevault, M. Balesdent, N. Bérend, and R. L. Riche, "Decoupled multidisciplinary design optimization formulation for interdisciplinary coupling satisfaction under uncertainty," *AIAA Journal*, vol. 54, no. 1, pp. 186–205, 2016.

[4] E. J. Cramer, J. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, "Problem formulation for multidisciplinary optimization," *SIAM Journal on Optimization*, vol. 4, no. 4, pp. 754–776, 1994.

[5] R. D. Braun, *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*, Ann Arbor, Stanford University, 1996.

[6] R. S. Sellar, S. M. Batill, and J. E. Renaud, "Response surface based, concurrent subspace optimization for multidisciplinary system design," in *Proceedings of the 34th Aerospace Sciences Meeting and Exhibit, 1996*, American Institute of Aeronautics and Astronautics, January 1996.

[7] J. Sobieszczanski-Sobieski, J. S. Agte, and R. R. Sandusky Jr., "Bilevel integrated system synthesis," *AIAA Journal*, vol. 38, no. 1, pp. 164–172, 2000.

[8] W. Yao, X. Q. Chen, W. C. Luo, M. van Tooren, and J. Guo, "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, vol. 47, no. 6, pp. 450–479, 2011.

[9] Z. Jiang, W. Chen, and B. J. German, "Statistical sensitivity analysis considering both Aleatory and epistemic uncertainties in multidisciplinary design," in *Proceedings of the AIAA AVIATION 2014 -15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 2014*, American Institute of Aeronautics and Astronautics, June 2014.

[10] Z. Hu and S. Mahadevan, "A surrogate modeling approach for reliability analysis of a multidisciplinary system with spatio-temporal output," *Structural and Multidisciplinary Optimization*, vol. 56, no. 3, pp. 553–569, 2017.

[11] S. Sankararaman and S. Mahadevan, "Likelihood-based approach to multidisciplinary analysis under uncertainty," *Journal of Mechanical Design*, vol. 134, no. 3, Article ID 031008, 2012.

[12] C. Liang, S. Mahadevan, and S. Sankararaman, "Stochastic multidisciplinary analysis under epistemic uncertainty," *Journal of Mechanical Design*, vol. 137, no. 2, Article ID 21404, 2015.

[13] C. Liang and S. Mahadevan, "Stochastic multidisciplinary analysis with high-dimensional coupling," *AIAA Journal*, vol. 54, no. 4, pp. 1209–1219, 2016.

[14] S. F. Ghoreishi and D. L. Allaire, "Compositional uncertainty analysis via importance weighted Gibbs sampling for coupled multidisciplinary systems," in *Proceedings of the 18th AIAA Non-Deterministic Approaches Conference, 2016*, usa, January 2016.

[15] S. F. Ghoreishi and D. L. Allaire, "Adaptive Uncertainty Propagation for Coupled Multidisciplinary Systems," *AIAA Journal*, vol. 55, no. 11, pp. 3940–3950, 2017.

[16] J. H. Mathews and K. D. Fink, *Numerical methods using MATLAB*, vol. 31, Prentice hall, Upper Saddle River, NJ, USA, 1999.

[17] E. Arian, "Convergence Estimates for Multidisciplinary Analysis and Optimization," in *Institute for Computer Applications in Science and Engineering (ICASE*, Institute for Computer Applications in Science and Engineering (ICASE), 1997.

[18] J. T. Allison, M. Kokkolaras, and P. Y. Papalambros, "On selecting single-level formulations for complex system design optimization," *Journal of Mechanical Design*, vol. 129, no. 9, pp. 898–906, 2007.

[19] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems*, 2nd edition, 2004, Methods for Non-Linear Least Squares Problems.

[20] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, "Metamodeling in multidisciplinary design optimization: how far have we really come?" *AIAA Journal*, vol. 52, no. 4, pp. 670–690, 2014.

[21] C. W. Liu, X. X. Jin, and L. S. Li, "A web services-based multidisciplinary design optimization framework for complex engineering systems with uncertainties," *Computers in Industry*, vol. 65, no. 4, pp. 585–597, 2014.

[22] H. Kim, B. Malone, and J. Sobieszczanski-Sobieski, "A Distributed, Parallel, and Collaborative Environment for Design of Complex Systems," in *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, California.

[23] P. N. Koch, J. P. Evans, and D. Powell, "Interdigitation for effective design space exploration using iSIGHT," *Structural and Multidisciplinary Optimization*, vol. 23, no. 2, pp. 111–126, 2002.

[24] S. Tiwari, S. Lankalapalli, and J. P. Leiva, "Design process integration and optimization with VisualDOC," in *Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September 2012.

[25] O. de Weck, J. Agte, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "State-of-the-Art and Future Trends in Multidisciplinary Design Optimization," in *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, USA, 2007.

[26] J. Agte, O. De Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "MDO: Assessment and direction for advancement-an opinion of one international group," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, pp. 17–33, 2010.

[27] S. Tiwari, H. Dong, and S. Lankalapalli, "Multidisciplinary Optimization with VisualDOC, in 10th World Congress on Structural and Multidisciplinary Optimization," in *Proceedings of the Multidisciplinary Optimization with VisualDOC, in 10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, 2013.

[28] M. S. Eldred, K. R. Dalbey, W. J. Bohnhoff et al., "DAKOTA : a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. Version 5.0, user's reference manual.," Tech. Rep. SAND2010-2184, 2010.

[29] J. R. R. A. Martins, C. Marriage, and N. Tedford, "PyMDO: An Object-Oriented framework for multidisciplinary design optimization," *ACM Transactions on Mathematical Software*, vol. 36, no. 4, article no. 20, 2009.

[30] J. Gray, K. T. Moorey, and B. A. Naylorz, "OpenMDAO: An open source framework for multidisciplinary analysis and optimization," in *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, MAO 2010*, USA, September 2010.

[31] J. Gray, K. T. Moore, T. A. Hearn, and B. A. Naylor, "Standard platform for benchmarking multidisciplinary design analysis and optimization architectures," *AIAA Journal*, vol. 51, no. 10, pp. 2380–2394, 2013.

[32] N. P. Tedford and J. R. R. A. Martins, "Benchmarking multidisciplinary design optimization algorithms," *Optimization and Engineering*, vol. 11, no. 1, pp. 159–183, 2010.

[33] A. B. Lambe and J. R. R. A. Martins, "Extensions to the design structure matrix for the description of multidisciplinary

design, analysis, and optimization processes," *Structural and Multidisciplinary Optimization*, vol. 46, no. 2, pp. 273–284, 2012.

[34] D. V. Steward, "The design structure system: a method for managing the design of complex system ," *IEEE Transactions on Engineering Management*, vol. 28, no. 3, pp. 71–74, 1981.

[35] N. F. Michelena and P. Y. Papalambros, "A hypergraph framework for optimal model-based decomposition of design problems," *Computational Optimization and Applications*, vol. 8, no. 2, pp. 173–196, 1997.

[36] X. Yao, J. Zhou, Y. Li, and E. Liu, "An Enhanced Collaborative Optimization Approach with Design Structure Matrix Algorithms to Group and Decouple Multidisciplines," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4340916, 2016.

[37] M. Imani and U. M. Braga-Neto, "Maximum-likelihood adaptive filter for partially observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.

[38] M. Imani and U. Braga-Neto, "Multiple Model Adaptive Controller for Partially-Observed Boolean Dynamical Systems," in *Proceedings of the Americal Control Conference*, 2017.

[39] J. Rogers and C. Bloebaum, "Ordering design tasks based on coupling strengths," in *Proceedings of the 5th Symposium on Multidisciplinary Analysis and Optimization*, Panama City Beach,FL,U.S.A..

[40] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: a review and new directions," *IEEE Transactions on Engineering Management*, vol. 48, no. 3, pp. 292–306, 2001.

[41] H. W. Park, M. S. Kim, D. H. Choi, and D. Mavris, "Optimizing the Parallel Process Flow for the Individual Discipline Feasible Method," in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia.

[42] Z. Lu and J. R. R. A. Martins, "Graph partitioning-based coordination methods for large-scale multidisciplinary design optimization problems," in *Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September 2012.

[43] H. M. E. Abdelsalam and H. P. Bao, "Re-sequencing of design processes with activity stochastic time and cost: An optimization-simulation approach," *Journal of Mechanical Design*, vol. 129, no. 2, pp. 150–157, 2007.

[44] J. T. Allison, M. Kokkolaras, and P. Y. Papalambros, "Optimal partitioning and coordination decisions in decomposition-based design optimization," *Journal of Mechanical Design*, vol. 131, no. 8, pp. 0810081–0810088, 2009.

[45] A. Kusiak and J. Wang, "Decomposition of the design process," *Journal of Mechanical Design*, vol. 115, no. 4, pp. 687–695, 1993.

[46] J. L. Rogers, *DeMAID: A Design Manager's Aide for Intelligent Decomposition User's Guide*, DeMAID, 1989.

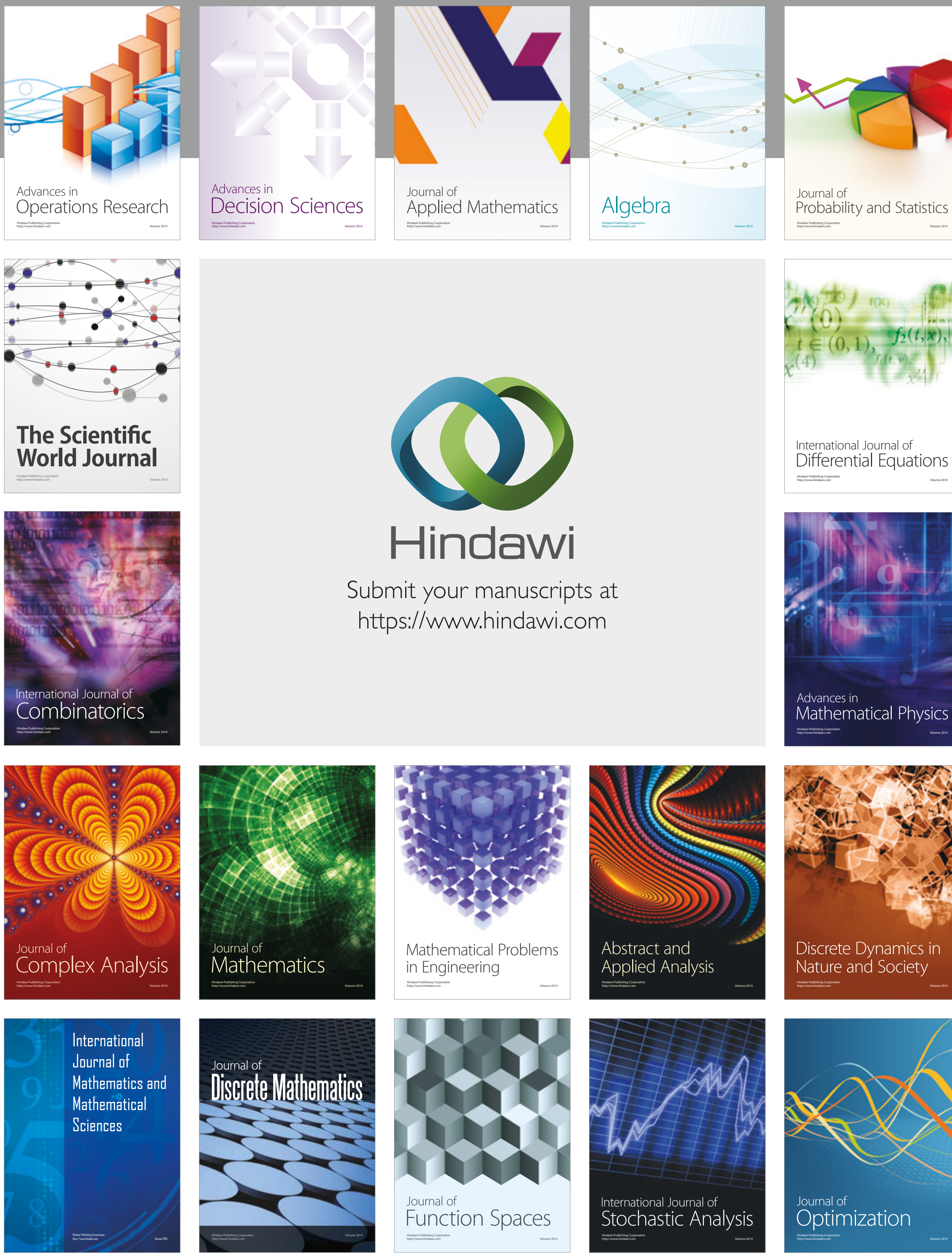
[47] J. Rogers, "DeMAID/GA - An enhanced design manager's aid for intelligent decomposition," in *Proceedings of the 6th Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA, USA.

[48] A. Pothen and C.-J. Fan, "Computing the block triangular form of a sparse matrix," *ACM Transactions on Mathematical Software*, vol. 16, no. 4, pp. 303–324, 1990.

[49] R. E. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[50] J. Moré, B. Garbow, and K. Hillstrom, "User guide for MINPACK-1," Tech. Rep. ANL–80-74, 1980.

[51] N. Tedford and J. Martins, "On the Common Structure of MDO Problems: A Comparison of Architectures," in *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Va, USA.


Advances in
Operations Research
Advances in
Decision Sciences
Journal of
Applied Mathematics
Algebra
Journal of
Probability and Statistics
The Scientific
World Journal
International Journal of
Differential Equations
Hindawi
Submit your manuscripts at
https://www.hindawi.com
International Journal of
Combinatorics
Advances in
Mathematical Physics
Journal of
Complex Analysis
Journal of
Mathematics
Mathematical Problems
in Engineering
Abstract and
Applied Analysis
Discrete Dynamics in
Nature and Society
International
Journal of
Mathematics and
Mathematical
Sciences
Journal of
Discrete Mathematics
Journal of
Function Spaces
International Journal of
Stochastic Analysis
Journal of
Optimization