10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

Multidisciplinary Optimization with VisualDOC

Santosh Tiwari, Hong Dong, Srinivas Lankalapalli, and Juan Pablo Leiva

Vanderplaats Research & Development, Inc., Novi, MI, USA. Email: {stiwari, hdong, slankal, jp}@vrand.com

1. Abstract
In this paper, we describe how to formally and systematically perform multi-disciplinary optimization (MDO) using VisualDOC. Typically, when performing MDO studies, the user is required to identify the linking/coupling variables, dependent and independent inputs/outputs, formulate the system-level and sub-system level optimization problems, integrate different disciplines and sub-systems together, and model the data flow and link the modeled problems with optimization to perform the design study. It is specifically demonstrated how VisualDOC can automate this entire process without requiring the user to write a computer program or manually perform the integration. With VisualDOC, the user is only required to identify the linking/coupling variables, formulate the system-level and sub-system level optimization problems, and define the flow of execution. This paper primarily focuses on the specific challenges in process integration, data transfer between different systems, and coordinated execution of optimization to perform the design study. The available computational and algorithmic tools and their solution as provided by VisualDOC are presented. For the purpose of demonstration, a heat exchanger design problem is considered with designable structural and thermal components. The computational model for heat exchanger is treated as black-box (i.e. no information is available to the optimizer about the analysis program and it is not possible to decompose/split the analysis for any system/sub-system) and is provided as a computer simulation. Three different MDO techniques: i) multiple disciplines feasible (MDF), ii) individual disciplines feasible (IDF), and iii) collaborative optimization (CO) are modeled in VisualDOC. A complete flowchart using standard VisualDOC components is presented for each MDO technique. The performance, efficiency, and suitability of each of these techniques are compared and their advantages and disadvantages are discussed. It is shown that the proposed approach for model creation and design process integration enables one to perform such design studies easily and reliably.
2. Keywords: VisualDOC, Multidisciplinary Optimization, Process Integration, MDF, IDF, CO

3. Introduction

Multi-disciplinary optimization (MDO) as the name suggests involves optimization with analysis components spanning multiple different (more than one) disciplines. Multi-disciplinary optimization as an optimization methodology is well established and is widely used in academia and industry [1, 2, 3, 4, 5, 6, 4, 7, 8, 10, 11]. A comprehensive overview and description of some of the most common MDO methods can be found in [3] and [11]. In this paper, a systematic solution procedure to solve a given MDO problem using a given formal MDO method is presented. Typically, when solving a MDO problem, the user is required to identify the system and sub-systems, linking/coupling variables, inputs/outputs, formulate the optimization problem, etc. to obtain the solution. The user relies on a programming-like environment and hand-crafts (implements) the code to execute the entire design process. With the advancements in MDO software such as VisualDOC, it is not necessary any more to write a computer program to solve a MDO problem.

Before introducing the formal MDO methods and their proposed solution procedure using VisualDOC, a distinction is made based on the coupling (linking) of multiple disciplines (or a set of equations). The type of dependence of a set of sub-systems (or disciplines) can be classified into two categories.

1. *Acyclic Dependency*: When more than one set of sub-systems (disciplines) are linked such that a multi-disciplinary analysis (MDA) can be performed without resorting to an iterative process to achieve consistency, then such a dependency is referred to as acyclic dependency in this paper. In this case, sub-systems are linked such that there is no reverse dependency (sub-system A can depend on sub-system B, but not vice versa). A dependency graph of such a set of sub-systems will not have a link (edge) emanating from a node that will point to any of its ancestors. To perform MDA, the sub-system(s) that is (are) not dependent on any other sub-system is (are) evaluated first, then the sub-systems that only depend upon the already evaluated sub-system(s) are evaluated, and so

on. The complete MDA is accomplished in a single evaluation of each sub-system and all the inputs and outputs are consistent.

2. *Cyclic Dependency*: When more than one set of sub-systems (disciplines) are linked such that a multi-disciplinary analysis (MDA) cannot be performed without resorting to some sort of iterative process to achieve consistency, then such a dependency is referred to as cyclic dependency in this paper. In this case, sub-systems are linked such that there does exist at least one reverse dependency (sub-system A can depend on sub-system B, and sub-system B can depend on sub-system A or its predecessors). A dependency graph of such a set of sub-systems will have at least one link (edge) emanating from a node that will point to one (or more) of its ancestors. In this case, performing a single evaluation of each sub-system will not accomplish MDA and the set of inputs and outputs need not be consistent.

Cyclic dependency is pictorially depicted in Figures 1 and 2. It should be noted that if the dependency is acyclic, then none of the MDO formulations proposed in literature are necessary to perform MDA and solve the optimization problem. The entire set of sub-systems behaves as a single system so far as optimization is concerned. At each iteration of the optimization process, all the sub-systems can be evaluated correctly in a single step, and therefore the formal MDO methods are not needed to solve such problems. Such problems typically do not fall under the purview of MDO. Only when the dependency is cyclic, formal MDO methods are necessary. Since, in such a case, MDA cannot be performed in a single step (iteration) to obtain consistent inputs and outputs, researchers have proposed various methods to perform optimization and achieve consistency in as few function evaluations (iterations) as possible. The focus of the MDO methods is not only to facilitate consistency, but also decouple them (e.g. to run them in parallel or achieve interdisciplinary independence) or reduce the associated computational cost. It is also important to note that all MDO formulations aim at obtaining convergence to a set of consistent inputs and outputs. In this paper, VisualDOC is used to model and solve different MDO formulations.

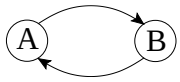

Figure 1: Cyclic dependency (2 systems)

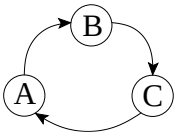

Figure 2: Cyclic dependency (3 systems)

VisualDOC [12, 13] is a general-purpose multidisciplinary design and optimization software. It is a tool for design process definition, integration, execution, and automation. It includes design modules such as Optimization, Design of Experiments, Responses Surface Models, and Probabilistic (Robust and Reliability-based) Models. VisualDOC can be used to add these modules to almost any design task. It allows the designer/user to graphically create a connected workflow of components and define each component in the flowchart appropriately. More information about VisualDOC can be found on its webpage *http://www.vrand.com/visualdoc.html*. VisualDOC includes tools for: i) specifying a process as a flowchart, defining the flow of execution (coordinate execution of different components), defining flow of information (communication of data between different components), interfacing with different kinds of analysis programs, and a large number of design algorithms. Thus, in essence, it includes all the pieces necessary to model a solution to a MDO problem. The user is still required to identify (specify) all the inputs and outputs for each discipline, and formulate the system-level and sub-system-level optimization problems.

An important aspect when formulating the MDO problem is the issue of system and sub-system boundaries. In an analytical problem [3, 11], the user often has the flexibility (freedom) to choose the system and sub-system boundaries. If the system boundaries are chosen appropriately then it may help the optimization process by minimizing the number of iterations (or function evaluations). There is no specific rule that governs the choice of the system boundaries. However, a general guideline is to choose the boundaries such that the number of linking variables (number of dependent inputs and outputs) is minimized. Lesser number of linking variables results in fewer optimization variables as well as fewer iterations to achieve consistency. Therefore, the associated computational cost is minimized. In this paper, the sub-systems (analysis programs) are treated as a black-box simulation. Hence, the system boundaries are a given, and the user does not have the freedom to define system boundaries. This is

almost always the case when working with domain-specific simulation software as the analysis program. The inputs and outputs to a given analysis program are fixed and the designer (user) cannot choose a different set of inputs or outputs to run a simulation. In this paper, the sub-systems are assumed to be black-boxes (closed simulation software) which have clearly defined inputs and outputs that cannot be altered.
The remainder of this paper is organized as follows. The following section contains an overview of the three different MDO methods presented in this study. In the next section, the example problem (Electronic Packaging Design) used for the purpose of demonstration is described. Section 6 contains the description of the formulated optimization problems and obtained simulation results. Finally in section 7, conclusions from this study are presented and the ongoing future work is described.

4. MDO Formulations

Numerical optimization algorithms have seen extensive development for over fifty years and, today, we can solve nonlinear constrained optimization problems involving many thousands of variables and constraints. In engineering, formal numerical optimization has been successfully applied to structures, fluid mechanics, heat exchangers, gas dynamic lasers etc. to name a few. Furthermore, optimization is not confined to a single discipline and multi-disciplinary optimization, or MDO, is now commonplace. It is therefore necessary to include facilities in optimization software to make it easy and efficient to solve MDO problems. MDO involves using Optimization to solve design problems that span multiple disciplines. Generally, the disciplines involved are coupled (linked) to each other such that the output of one discipline depends upon the output of another. Often, this dependency is cyclic in that the output of first depends upon the output of second and vice versa. In such a case, the optimization process must not only find the optimum but also ensure that the disciplines (their inputs and outputs) are consistent with each other. To solve such linked systems, several MDO strategies have been proposed. In this section, a brief description of three MDO techniques (MDF, IDF and CO) is presented. For a more comprehensive overview of these MDO formulations, the reader is referred to [3] and [11].
The following general description of a coupled optimization problem (figure and notation taken from [11]) is used in this paper. There are two disciplines in the coupled example problem shown in Figure 3. The description of each symbol in Figure 3 is as follows.

- x_1 is the independent input to sub-system 1 only
- x_2 is the independent input to sub-system 2 only
- x is the common independent input to sub-systems 1 and 2
- y_1 is the output from sub-system 1 that is not input to any other sub-system
- y_2 is the output from sub-system 2 that is not input to any other sub-system
- y_{12} is the output from sub-system 1 that is input to sub-system 2
- y_{21} is the output from sub-system 2 that is input to sub-system 1

Hence, to evaluate (simulate) sub-system 1, the inputs x, x_1, and y_{21} are required. The outputs from sub-system 1 are y_1 and y_{12}. Similarly to evaluate (simulate) sub-system 2, the inputs x, x_2, and y_{12} are required. The outputs from sub-system 2 are y_2 and y_{21}. The design variables are x, x_1, and x_2. The optimization problem statement is as follows.

$$\begin{array}{ll} \text{Minimize} & f(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) \\ \text{Subject to} & g(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) \leq 0 \\ & h(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) = 0 \end{array} \tag{1}$$

4.1 Multi-disciplinary Feasible (MDF)

This is a single-level method. In the MDF formulation, as the name suggests, all the disciplines are feasible (consistent) at each iteration. In this strategy, only those inputs that are not output from any sub-system (independent inputs) are treated as design variables. All the linking variables are treated as part of the system. A local iteration between the sub-systems is performed until the linking variables (both the inputs and outputs) become consistent. This is why this formulation is also called All-In-One (AIO) formulation. In this case, MDA is achieved for each discipline at each iteration. After the consistency is achieved, the objectives and constraints are then evaluated and sent to the optimizer. This

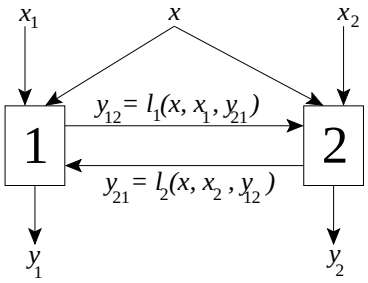

Figure 3: MDO Problem with Two Disciplines

process repeats till the optimization has converged. The problem statement for the MDF formulation is as follows (it is identical to original optimization problem statement). The MDF formulation is pictorially depicted in Figure 4. The optimizer generates the value of variables (independent inputs) x, x_1, and x_2 at each iteration. Then the linking variables y_{12} and y_{21} are determined using a local iteration till consistency is achieved. After y_{12} and y_{21} are determined, the objective function f and constraints g can be determined. The responses are then propagated to the optimizer.

$$\begin{array}{ll} \text{Design variables:} & x, x_1, x_2 \\ \text{Minimize} & f(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) \\ \text{Subject to} & g(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) \leq 0 \\ & h(x, x_1, x_2, y_{12}, y_{21}, y_1, y_2) = 0 \end{array} \tag{2}$$

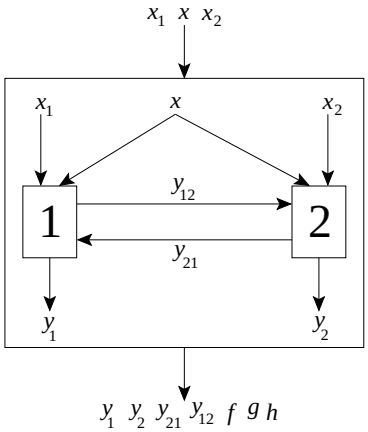

Figure 4: MDF Formulation

4.2 Individual Discipline Feasible (IDF)

IDF also is a single-level method. The IDF formulation decouples different sub-systems by using auxiliary optimization variables. With this strategy, it is possible to analyze (simulate) each sub-system at once in parallel. Therefore, this formulation is also called distributed analysis optimization. With this formulation, a complete multi-disciplinary analysis (MDA) is not required, and the optimization drives the complete system to feasibility and optimality by controlling the interdisciplinary coupling variables. Corresponding to each dependent (linking) variable, surrogate variables are introduced. All the sub-systems are evaluated using the surrogate variables. The objectives and the constraints are also evaluated using the surrogate variables. To maintain consistency, additional equality constraints are introduced that attempt to match the surrogate variables with the corresponding dependent outputs. The IDF formulation is pictorially depicted in Figure 5. With IDF, it can be noticed that there is an increase in the number of variables as well as addition of nonlinear equality constraints. Both these modifications generally increase the computational cost (number of iterations) associated with optimization. This increase in computational cost is generally offset by the fact that MDA is not required in the case of IDF and therefore local

iteration of sub-systems is not performed.

$$
\begin{array}{ll}
\text{Design variables:} & x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}} \\
& y_{12} = l_1\left(x, x_1, \overline{y_{21}}\right) \\
& y_{21} = l_1\left(x, x_2, \overline{y_{12}}\right) \\
\text{Minimize} & f(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) \\
\text{Subject to} & g(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) \leq 0 \\
& h(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) = 0 \\
& \|y_{12} - \overline{y_{12}}\| = 0 \\
& \|y_{21} - \overline{y_{21}}\| = 0
\end{array}
\tag{3}
$$

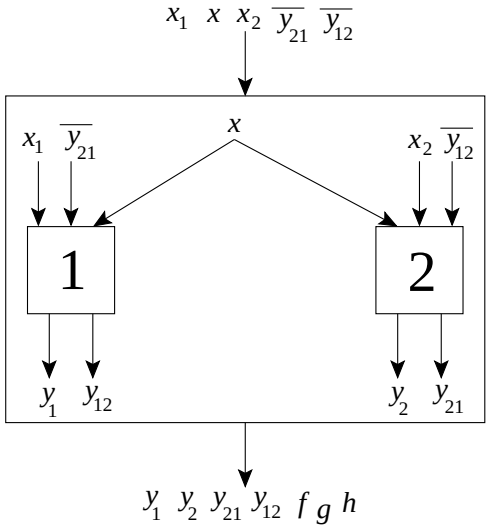

Figure 5: IDF Formulation

A significant issue with the IDF formulation (and with any formulation that uses surrogate variables) is that the optimizer needs the lower bound and upper bound for the surrogate variables. This information may not always be available. The initial value is also needed for optimization but this limitation is present with every MDO method including MDF.

4.3 Collaborative Optimization (CO)

CO is a bi-level approach which has system level optimization and the subsystem level optimization. The system level optimization drives specific system-level targets towards the optimum. The sub-system level optimization control the discipline design variables and the linking variables to match the system level targets. At any iteration of system-level optimization, the sub-system level optimization is run to match the system-level targets. Since CO is a multi-level formulation, it is generally very expensive compared to MDF, IDF, and other single-level approaches. The advantage of CO however is that it promotes disciplinary autonomy whilst achieving interdisciplinary compatibility. With CO, the subsystems are maintained at a feasible state which prevents a breakdown of disciplinary analysis. The CO formulation is pictorially depicted in Figure 6. Since there are system-level and sub-system level optimization problems, it is necessary to define a convention. The convention is shown in Table 1 and is taken form [11]. In Table 1, x^1 and x^2 are surrogate for the independent common design variable x. With this convention for the design variables, the CO formulation (for two subsystems) is as follows.

$$
\begin{array}{ll}
\text{System-level optimization} & \\
\text{Design variables:} & x, \overline{y_{12}}, \overline{y_{21}} \\
\text{Minimize} & f(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) \\
\text{Subject to} & g(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) \leq 0 \\
& h(x, x_1, x_2, \overline{y_{12}}, \overline{y_{21}}, y_1, y_2) = 0 \\
& J_1 = \|x^1 - x\| + \|y_{12} - \overline{y_{12}}\| = 0 \\
& J_2 = \|x^2 - x\| + \|y_{21} - \overline{y_{21}}\| = 0
\end{array}
\tag{4}
$$

Table 1: CO Variable Naming Convention

Original	System	Subsystem
x	x	x^1, x^2
x_1	-	x_1
x_2	-	x_2
y_{12}	$\overline{y_{12}}$	y_{12}
y_{21}	$\overline{y_{21}}$	y_{21}

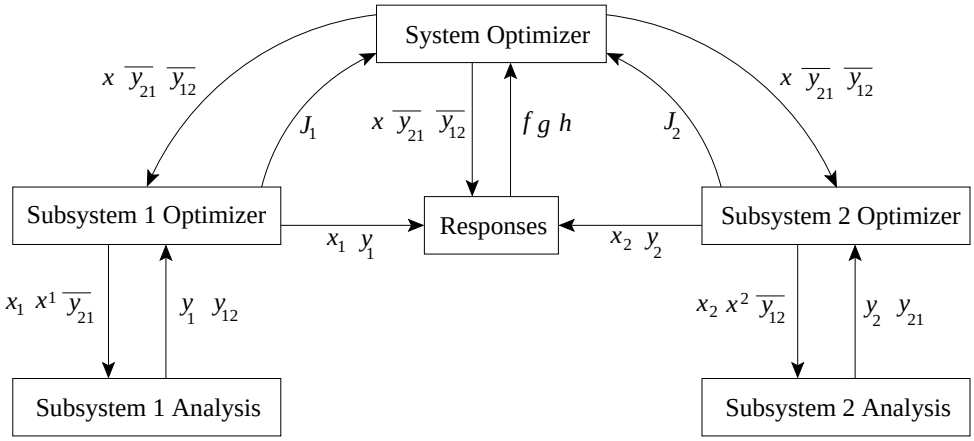

Figure 6: CO Formulation

$$
\begin{array}{ll}
\text{Subsystem 1 optimization} & \\
\text{Design variables:} & x^1, x_1, y_{12} \\
\text{Input parameters:} & x, \overline{y_{12}} \text{ (from system optimizer)} \\
\text{Minimize} & J_1 = \|x^1 - x\| + \|y_{12} - \overline{y_{12}}\| \\
\text{Subject to} & g_1(x^1, x_1, y_{12}, y_1) \leq 0 \\
 & h_1(x^1, x_1, y_{12}, y_1) = 0
\end{array} \tag{5}
$$

$$
\begin{array}{ll}
\text{Subsystem 2 optimization} & \\
\text{Design variables:} & x^2, x_2, y_{21} \\
\text{Input parameters:} & x, \overline{y_{21}} \text{ (from system optimizer)} \\
\text{Minimize} & J_2 = \|x^2 - x\| + \|y_{21} - \overline{y_{21}}\| \\
\text{Subject to} & g_2(x^2, x_2, y_{21}, y_2) \leq 0 \\
 & h_2(x^2, x_2, y_{21}, y_2) = 0
\end{array} \tag{6}
$$

5. Description of the Electronic Packaging Problem

The electronic packaging design [1, 14, 15, 3] is a famous multidisciplinary optimization (MDO) problem. It has two subsystems which are electrical and thermal subsystems. The task in this problem is to design a heat sink which is used to dissipate the heat generated by the resistors. The objective in this problem is to maximize the watt density of the system by choosing i) suitable dimensions for the heat sink, ii) deciding on the material to be used for the resistor, and iii) their normal resistances. An important feature of this problem is that the two systems are coupled and therefore they affect each other and hence one system cannot be solved independently of the other. Watt density of the system is defined as the total power dissipated by the circuit divided by the volume of the heat sink. There are various constraints that limit the current division, maximum temperature reached, reliability, and weight. Most of the details of this problem are omitted in this paper and the reader is referred to [14] and [15] for a complete description of the this problem. The problem description and the source code can be downloaded from http://www.eng.buffalo.edu/Research/MODEL/mdo.test.orig/class2prob3.html. There are eight independent design variables (x_1 to x_8), 13 state variables (y_1 to y_{13}), two inequality constraints (g_1 and g_2) and one equality constraint h. The lower bound and the upper bound for the design variables

is taken from [3]. The electrical subsystem is available in analytical form as equations, and the thermal subsystem is available as an executable program with fixed number of inputs and outputs. The inputs and outputs for the two subsystems are as follows.

Thermal subsystem
Inputs: x_1, x_2, x_3, x_4 and y_2, y_3
Outputs: y_{11}, y_{12}, y_{13}

Electrical subsystem
Inputs: x_5, x_6, x_7, x_8 and y_{11}, y_{12}, y_{13}
Outputs: y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}

The objective function f, inequality constraints g, and the equality constraint h are computed using the state variables y. The inputs and outputs for each sub-system are shown in Figure 7.

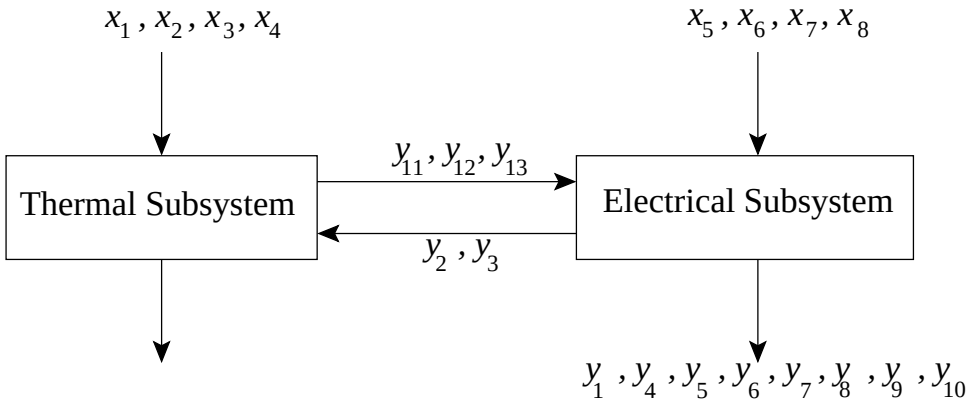

Figure 7: Inputs and Outputs for the Thermal and Electrical Subsystems

6. Solution using VisualDOC

In this section, the solution to the Electronic Packaging Design problem using VisualDOC is presented. For an optimization software to be able to solve MDO problems involving MDF, IDF, and CO (and various other) formulations, it should provide certain capabilities apart from the usual optimization and process integration capability. These capabilities enable a software to easily, efficiently, and reliably solve a MDO problem. The list of needed capabilities are as follows.

- It should support multi-level optimization (needed for bi-level MDO formulations). A system level optimizer should be able to drive sub-system level optimizers. This also implies that an analysis component (or sub-flow) must be able to contain optimization and other sub-flows.
- The sub-system level optimizers are seeded with information generated in the system level optimizer. The software should thus be able to not only communicate the value of design variables, objectives, and constraints; but transfer lower bound, upper bound, initial value, optimum value etc. from one component to another across different levels.
- The sub-system level optimization is run multiple times. Each time, the optimizers start from some initial value. It is desired (for faster convergence) that subsequent runs of an optimization component use the best found solution in the previous run as the starting point.

VisualDOC includes all the above capabilities to facilitate solution of MDO problems. The solution using the three MDO (MDF, IDF, and CO) strategies is presented next.

6.1 VisualDOC Solution for MDF Formulation

The MDF formulation is akin to regular optimization in which the linking variables are part of the analysis sub-flow. A fixed point iteration like scheme is used inside the analysis sub-flow to achieve consistency between the linking variables. The variables, objectives, and constraints for the optimizer are as follows.

Variables: x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8
State variables: y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}
Responses: f, g_1, g_2, and h

The VisualDOC flowchart is shown in Figure 8. The VisualDOC convergence plot for the MDF formulation is shown in Figure 9. The optimization algorithm took 186 iterations to converge (find the optimum). Since the analysis-subflow executes a fixed point iteration to achieve consistency, for each optimization iteration, there are multiple simulations of the analysis programs. The total number of function evaluations (one function evaluation implies an evaluation of each sub-system) in this case is 492. The obtained optimum value of the objective function is -6.39e-5.

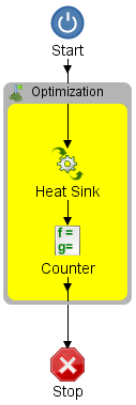

Figure 8: VisualDOC Flowchart for MDF Formulation

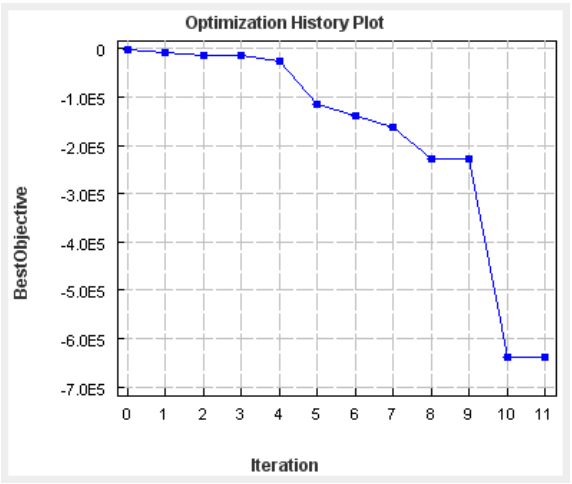

Figure 9: Convergence Plot for MDF Formulation

6.2 VisualDOC Solution for IDF Formulation

In the IDF formulation, surrogate variables are introduced for all the linking variables. The symbol z is used for the surrogate variables (i.e. z_i is a surrogate for variable x_i and z_{ij} is a surrogate for linking variable y_{ij}). The analysis are decoupled and therefore can be run in parallel if desired. In this example, there are no common independent variables. The variables, objectives, and constraints for the optimizer are as follows.

Design variables:	$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, z_2, z_3, z_{11}, z_{12}, z_{13}$
Input to Thermal subsystem:	$x_1, x_2, x_3, x_4, z_2, z_3$
Output from Thermal subsystem:	y_{11}, y_{12}, y_{13}
Input to Electrical subsystem:	$x_5, x_6, x_7, x_8, z_{11}, z_{12}, z_{13}$
Output from Thermal subsystem:	$y_1, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$
Original Objectives and Constraints:	f, g_1, g_2, h (evaluated using surrogates instead of output linking variables)
Additional Equality Constraint 1:	$h_1 = \|y_2 - z_2\| + \|y_3 - z_3\| = 0$
Additional Equality Constraint 2:	$h_2 = \|y_{11} - z_{11}\| + \|y_{12} - z_{12}\| + \|y_{13} - z_{13}\| = 0$

(7)

The VisualDOC flowchart is shown in Figure 10. The VisualDOC convergence plot for the IDF formulation is shown in Figure 11. The optimization algorithm took 220 iterations to converge (find the optimum). Since the analysis-subflow executes exactly once for each optimization iteration, the total number of function evaluations (one function evaluation implies an evaluation of each sub-system) in this case is 220. We notice that IDF is more than twice as fast (as compared to MDF) on this example problem. The obtained optimum value of the objective function is -5.86e5 which is slightly worse than the solution obtained using MDF. The convergence plot in Figure 11 shows red markers since the constraints are very slightly violated (1e-4 instead of the specified tolerance of 1e-6).

6.3 VisualDOC Solution for CO Formulation

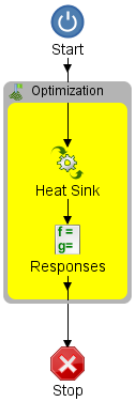

Figure 10: VisualDOC Flowchart for IDF Formulation

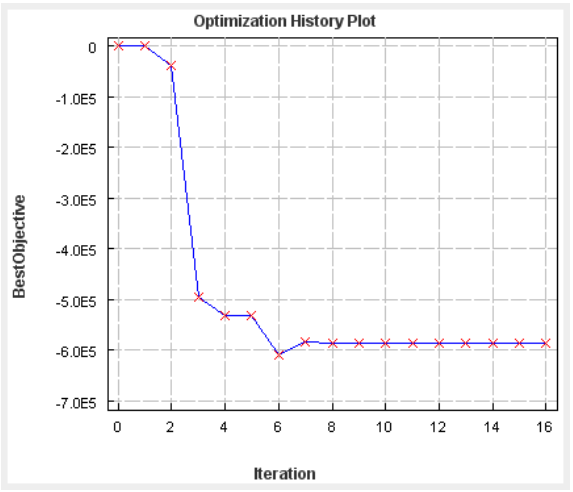

Figure 11: Convergence Plot for IDF Formulation

In the CO formulation, system-level and sub-system level optimization are performed. As in IDF, surrogate variables are introduced for all the linking variables. Since there are no common independent variables in this problem, no surrogates are introduced for them. The symbol z is used for the surrogate variables (i.e. z_i is a surrogate for variable x_i and z_{ij} is a surrogate for linking variable y_{ij}). The analysis are decoupled and are optimized separately. The variables, objectives, and constraints for the optimizer are as follows.

System-level Optimization	
Design variables:	$z_2, z_3, z_{11}, z_{12}, z_{13}$
Original Objectives and Constraints:	f (evaluated using surrogates instead of output linking variables)
Additional Equality Constraint 1:	$J_1 = \|y_{11} - z_{11}\| + \|y_{12} - z_{12}\| + \|y_{13} - z_{13}\| = 0$
Additional Equality Constraint 2:	$J_2 = \|y_2 - z_2\| + \|y_3 - z_3\| = 0$

(8)

Thermal Subsystem Optimization	
Design variables:	$x_1, x_2, x_3, x_4, y_2, y_3$
Input parameters:	$z_2, z_3, z_{11}, z_{12}, z_{13}$ (from system optimizer)
Input to Thermal subsystem:	$x_1, x_2, x_3, x_4, y_2, y_3$
Output from Thermal subsystem:	y_{11}, y_{12}, y_{13}
Minimize	$J_1 = \|y_{11} - z_{11}\| + \|y_{12} - z_{12}\| + \|y_{13} - z_{13}\| = 0$
Subject to	Thermal constraints evaluated using output linking variables

(9)

Electrical Subsystem Optimization	
Design variables:	$x_5, x_6, x_7, x_8, y_{11}, y_{12}, y_{13}$
Input parameters:	$z_2, z_3, z_{11}, z_{12}, z_{13}$ (from system optimizer)
Input to Electrical subsystem:	$x_5, x_6, x_7, x_8, y_{11}, y_{12}, y_{13}$
Output from Thermal subsystem:	$y_1, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$
Minimize	$J_2 = \|y_2 - z_2\| + \|y_3 - z_3\| = 0$
Subject to	Electrical constraints evaluated using output linking variables

(10)

The VisualDOC flowchart is shown in Figure 12. The VisualDOC convergence plot for the CO formulation is shown in Figure 13. The system level optimization took 141 iterations to converge. This implies that the sub-system level optimizers were run 141 times each. The total number of function evaluations used by the Thermal subsystem optimization (for all the runs) is 4532, whereas for the Electrical subsystem optimization is 2272. Hence, the Thermal subsystem optimization took an average of 32 function

evaluations and Electrical subsystem optimization took an average of 16 function evaluations to converge. It can easily be noticed that the CO (bi-level) formulation is significantly more expensive than MDF and IDF (single-level). The obtained optimum value of the objective function for this example problem is -6.47e5 which is slightly better than the solution obtained by MDF.

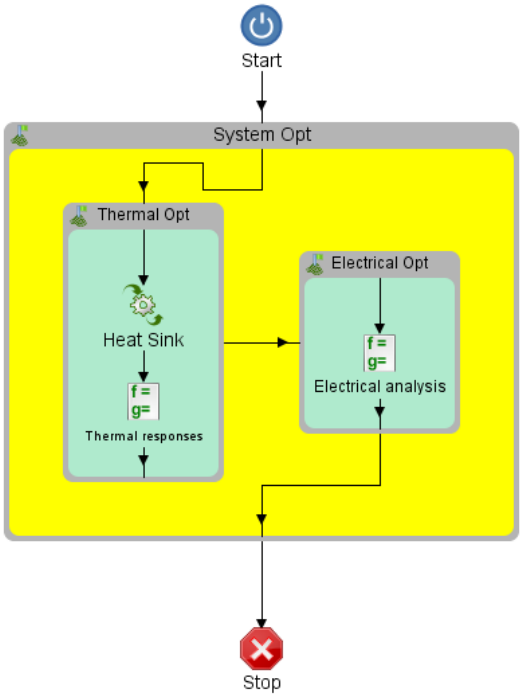

Figure 12: VisualDOC Flowchart for CO Formulation

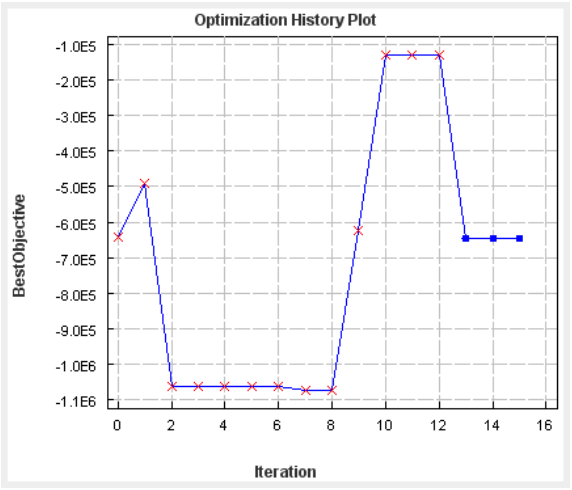

Figure 13: Convergence Plot for CO Formulation

7. Conclusion and Future Work

In this paper, the Electronic Packaging problem is solved in an MDO framework using VisualDOC. Three different MDO techniques (multi-disciplinary feasible - MDF, individual discipline feasible - IDF, and collaborative optimization - CO) are demonstrated. The focus of this paper is on using VisualDOC instead of writing a computer program to solve these problems. The specific capabilities that an optimization software should support for the solution using MDO formulations are discussed and it is demonstrated how the process integration and parameterization capability of VisualDOC facilitates such solution approach. Acceptable solutions were obtained with all the MDO formulations. It was observed that IDF was faster (220 function evaluations) than MDF (492 function evaluations) and CO (4532 Thermal simulations and 2272 Electrical simulations). CO however was able to obtain a slightly better solution than MDF and IDF.

It is also shown in this paper, that a software system such as VisualDOC can automate the entire process of solving a multi-disciplinary optimization problem using any of the MDO formulations, but the user is still responsible for making all the decisions. The decisions include defining system and sub-system boundaries, determining inputs and outputs to each analysis program, and formulating system and sub-system level optimization problems. After the user has formulated the optimization problems, VisualDOC can then be used to obtain the optimum solution. It was also observed that all the three MDO formulations presented in this paper require the user to make an educated guess for the bounds and initial value of the surrogate variables. The performance and the total execution time vary significantly on the choice of surrogate variables and the starting point (initial value) for the optimization. It was also observed that adding non-linear equality constraints almost always makes the optimization problem harder (the number of optimization iterations in MDF is lesser than in IDF). The other MDO techniques namely Simultaneous Analysis and Design (SAND), Bi-level Integrated System Synthesis (BLISS), and Concurrent Sub Space Optimization (CSSO) are not presented in this paper which are part of the currently ongoing future work.

8. References

[**1**] S. Kodiyalam and J. Sobieszczanski-Sobieski. Multidisciplinary design optimization - some formal methods, framework requirements, and application to vehicle design. *International Journal of Vehicle Design*, 25(1–2):3–22, 2001.

[**2**] K. F. Hulme and C. L. Bloebaum. A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE. *Structural Multi-disciplinary Optimization*, 19:17–35, 2000.

[**3**] K. F. Hulme. *The Design of a Simulation-based Framework for the Development of Solution Approaches in Multidisciplinary Design Optimization*. PhD thesis, University of NewYork at Buffalo, 2000.

[**4**] K. F. Hulme, C. L. Bloebaum, and Y. Nozaki. A performance-based investigation of parallel and serial approaches to multidisciplinary analysis convergence. In *Proceedings of the 8^{th} AIAA/USAF/NASA /ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000.

[**5**] N. M. Alexandrow and R. M. Lewis. Comparative properties of collaborative optimization and other approaches to MDO. In *Proceedings of the 1^{st} ASMO UK/ISSMO Conference on Engineering Design Optimization*, 1999.

[**6**] N. M. Alexandrow and S. Kodiyalam. Initial results of an MDO method solution study. In *Proceedings of 7^{th} AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 1315–1327, 1998.

[**7**] J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, 14(1):1–23, 1997.

[**8**] N. M. Alexandrow and M. Y. Hussaini. *Multidisciplinary Design Optimization: State of the Art (Proceedings in Applied Mathematics Series: No. 80)*. Soc for Industrial & Applied Math, 1997.

[**10**] J. Sobieszczanski-Sobieski. Multidisciplinary design optimization: An emerging new engineering discipline. *Advances in Structural Optimization*, pages 783–496, 1995.

[**11**] C. C. Johnson. An introduction to multidisciplinary design optimization methods. Master's thesis, Clemson University Clemson SC, 2012.

[**12**] S. Tiwari, H. Dong, B. Watson, and J. P. Leiva. Visualdoc: New capabilities for concurrent and integrated simulation and design. In *13^{th} AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2010-9177.

[**13**] S. Tiwari, S. Lankalapalli, and J. P. Leiva. Design process integration and optimization with visualdoc. In *14^{th} AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2012-1358798.

[**14**] J. Korngold, G. Gabriele, J. Renaud, and G. Kott. Application of multidisciplinary design optimization to electronic package design. In *Proceedings of the 4^{th} AIAA/NASA/USAF/OAI Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-92-4704-CP.

[**15**] D. Xiaoping and W. Chen. Efficient uncertainty analysis methods for multidisciplinary robust design. *AIAA Journal*, 40(3):545–552, 2002.