Zhang, Shuai and Minisci, Edmondo and Liu, Mingyong and Huang, Yunke and Riccardi, A (2018) Improved collaborative optimization in multidisciplinary design optimization problems. In: 8th International Systems & Concurrent Engineering for Space Applications Conference, 2018-09-26 - 2018-09-28, University of Strathclyde. ,

IMPROVED COLLABORATIVE OPTIMIZATION IN MULTIDISCIPLINARY DESIGN OPTIMIZATION PROBLEMS

Shuai Zhang [(1)], Edmondo Minisci [(2)], Mingyong Liu[(1)], Yunke Huang[(1)], A Riccardi[(2)]

[(1)]School of Marine Science and Technology, Northwestern Polytechnical University, China, (shuaizhang@mai.nwpu.edu.cn)
[(2)] Aerospace Centre of Excellence, Department of Mechanical & Aerospace Engineering, University of Strathclyde, UK, (edmondo.minisci@strath.ac.uk)

ABSTRACT

This paper is about a new approach for concurrent design based on collaborative optimization, a distributed optimization method for multidisciplinary designs. The key idea of the proposed method is to consider the global objective in each subspace optimization problem with an additional interaction channel for coupling variables, while maintaining an easy coordination of design variables for system level problem. The improved collaborative optimisation is applied to two academic test cases to demonstrate its feasibility and validity.

1. INTRODUCTION

In their review about Multidisciplinary Design Optimization (MDO), Joaquim and Andrew [1] propose to define it as a field of engineering that focuses on the use of numerical optimization for the design of systems that involve a number of disciplines or subsystems. Designers have to simultaneously consider different disciplines, because the performance of a multidisciplinary system is influenced not only by the performance of the individual discipline but also by their interactions. In this context, a simple single-level method (centralized controller) can struggle or fail to handle complexity of the system/problem. Currently, MDO problems are better handled via more sophisticated distributed strategies that allow designers to deal with the involved optimization problems independently via their own prefer optimization tools, maintaining coherency among shared variables. Since with the concurrent increase of complexity of different disciplines, each discipline is going to be a black-box for other disciplines and even for the system, a bottom-up approach can be much more appropriate to ensure the concurrent cooperation among the different disciplines.

Collaborative Optimization (CO) is quite a popular distributed method for the design of multidisciplinary systems that was first formulated and proposed in 1994 [2]. The standard collaborative optimization decomposes the MDO problem into the single disciplines, providing a significant degree of independence for each discipline, leading to the disciplinary subspaces with an unusually high level of autonomy, and the MDO problem is reformulated as a two-level optimization. The system level is responsible for the coordination by determining targets for each subspace responses with compatibility constraints. The objective of subspace level is to match the targets from system level as closely as possible while satisfying the local constraints. CO has been successfully used in many academic examples and practical engineering design problems, such as the design of launch vehicles [3], rocket engines [4], aircraft family design [5]. However, CO has major limitations that in practice lead to poor computational performance and convergence characteristics. It is understood that the reason of the issues is the use of equality form for the compatibility constraints in system level, which makes it hard to satisfy the Karush-Kuhn-Tucker (KKT) conditions [6]. To alleviate the issues, the response surface technology was used to approximately estimate the system level compatibility constraint, which improved the convergence problem to some extent [7]. Another approach is using relaxation method to improve the convergence problem and poor efficiency, where the system level compatibility constraint is used by inequality form instead of equality form through a relaxation factor [6]. These methods improve the convergence problem to some extent, but the objective function is still presented in the system level only, meaning that the subspace has weak design authority.
Brain and Kroo [8] proposed an enhanced collaborative optimization (ECO), which allows each subspace to have the prior knowledge of all other subspaces' constraints and global objective function. This enables the subspace to be the main decision-maker, while maintaining the low dimensionality of the system level problem. Although the efficiency and robustness were enhanced by modelling all the preference of other subspaces' constraints in each subspace, the trade-off is in the additional time required to build and update the model for each subspace. In addition, although the information interaction among subspaces is also enhanced, it looks like ECO is contrary to the original intention of CO, because it is not expected for each subspace to have all other subspaces' preferences beforehand. In contrast to the simplified solution of the decomposed optimization problems, each subspace in ECO has the same number of constraints like the All-in-One method description, which is not practical. Considering the information interaction among subspaces, in the collaborative optimization, each subspace has very limited knowledge of the preference and constraints of other subspaces. Information is only

shared indirectly via the system level targets. This provides a significant freedom for each subspace to handle the problem, enabling disciplinary designers to make the optimization independently, but on the other hand, leading to a low efficiency of convergence.
This paper focuses on the enhancement of subspace design authority and information interaction among subspaces. We propose an improved collaborative optimization (ICO) based on the original CO and ECO. The key idea in ICO is to consider the global objective in each subspace optimization problem with an additional interaction channel for coupling variables, while maintaining an easy coordination of design variables for system level problem. The improved collaborative optimization has two main contributions. Firstly, ICO enhances the subspace design authority. It can be assumed that each subspace knows the relevant portions of the global objective rather than tries to best match some set of targets. Each subspace should solve the optimization problem while considering the local objective and the local constraints. That is, the subspaces are responsible for most of the design decisions and the system is limited to providing dynamic 'moving limits'. The moving limits ensure that the design variables converge in the right direction and will not take large steps in the wrong direction. Transferring the global objective from the system level into subspace level is an important improvement, because it pays more attention to the individual rather than the system. Secondly, In most works, there is no direct interaction among subspaces [8]. Information of each subspace (discipline or constraints) is only shared indirectly via the system optimization results. This indirect information may slow down the convergence. In the proposed approach, a public memory space for coupling variables is proposed, making it possible for each subspace to call the relevant coupling variables directly. The exchange of state (coupling) variables has the added benefit that it exchanges the connotative constraints of each discipline, leading to a more efficient compatibility. A simple framework with added exchange channel in a two-discipline example is shown in Fig. 1.

This paper is organized as follows. Section 2 provides the compact mathematical description of MDO and CO. Section 3 provides a detailed description of the proposed method ICO. Section 4 illustrates the application of ICO to some examples, followed by conclusions in Section 5.

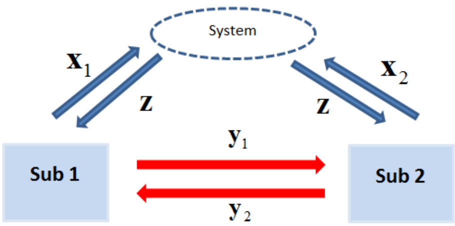

Figure 1. ICO with two disciplines

2. MDO and CO

An overview of MDO problem and the typical decomposition method CO are presented in this section. The improved collaborative optimization is shown in the next section with the comparison of CO.

In MDO problems, each discipline has a certain degree of independence but also communicate with other disciplines through coupling variables. The outputs of one discipline may depend on its design variables and the other subspaces' state variables. The formulation of MDO problem is defined as

$$\begin{array}{lll}
\min & f = f(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) & \\
\text{s.t} & \mathbf{g}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) > 0 & \\
\text{Analysis} & \mathbf{y}_i = \mathbf{D}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_j) & \\
\text{Boundary} & \mathbf{x}_i^{\min} \le \mathbf{x}_i \le \mathbf{x}_i^{\max} & \mathbf{x}_s^{\min} \le \mathbf{x}_s \le \mathbf{x}_s^{\max} \\
 & \mathbf{y}_i^{\min} \le \mathbf{y}_i \le \mathbf{y}_i^{\max} & \mathbf{y}_j^{\min} \le \mathbf{y}_j \le \mathbf{y}_j^{\max} \\
 & i, j = 1, 2, \ldots, n \quad i \ne j &
\end{array}$$

where f is objective function. $\mathbf{g}_i$ is local constraint. $\mathbf{D}_i$ is the discipline. $\mathbf{x}_s$ and $\mathbf{x}_i$ are shared design variable and local design variable. $\mathbf{y}_i$ is state/coupling variable.

CO is one of decomposition-based methods that divide a design problem into system level problem and subspace level problem. The objective function is only presented in the system level. The optimization results of each subspace are considered by constraints in system level. The system level is illustrated as:

$$\begin{array}{ll}
\min & f = f(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) \\
\text{s.t} & J_i = \left\| \mathbf{z} - \mathbf{x}_s^* \right\|_2^2 = 0 \qquad i = 1, \ldots, n
\end{array}$$

where $\mathbf{x}_s^*$ are subspace target responses that provide each subspace's best attempt to meet the system level targets $\mathbf{z}$.

The constraint of system level J_i is the compatibility term, which is presented as entirely objective function of subspace level. The subspace tries its best to match the targets for shared variables (here the shared variables includes shared design variables and state/coupling variables) that have been sent by the system level, while satisfying the local constraints. The subspace level is illustrated as:

$$\begin{array}{ll}
\min & J_i = \left\| \mathbf{x}_s - \mathbf{z} \right\|_2^2 \\
\text{s.t.} & \mathbf{g}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) \ge 0 \\
 & \mathbf{y}_i = \mathbf{D}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_j) \qquad i, j = 1, \ldots, n, \quad i \ne j
\end{array}$$

Note that, the x_s^* are treated as dependent variables, which demands that the subspace must be re-optimized each time when the system level evaluates its constraints. This increases the burden on computation. In addition, the number of shared variables (shared design variables and coupling variables) has a negative impact on the computational efficiency, which means decreasing the number variables can improve the convergence efficiency.

3. IMPROVED COLLABORATIVE OPTIMIZATION METHOD

3.1 System level problem

Similar to the basic CO, the improved collaborative optimization is also composed of two levels. The system level is an unconstrained minimization problem with a memory of coupling variables. The system is defined as

$$\begin{aligned} &\min \quad J_{sys} = \sum (\mathbf{z} - \mathbf{x}_s^*)^2, \quad \mathrm{sto}(\mathbf{y}) \\ &\text{s.t.} \quad \text{No constraints} \end{aligned} \tag{1}$$

where $\mathbf{z}$ is the system level target for shared design variable. $\mathbf{x}_s^*$ is each subspace's best attempt to match the system level target while satisfying local constraints, and $\mathrm{sto}(\mathbf{y})$ is a storage of coupling variables.

The objective of unconstrained minimization is to ensure that all subspaces converge to the same values of shared design variables $\mathbf{x}_s^*$ while satisfying their local constraints. The memory of coupling variables $\mathrm{sto}(\mathbf{y})$ collected from each subspace produces a route for each subspace to call corresponding coupling variables directly if needed. The inputs of system level contain $\mathbf{x}_s^*$ and $\mathbf{y}$. $\mathbf{x}_s^*$ are subspace target responses providing each subspace's best attempt to achieve compatibility. $\mathbf{y}$ are coupling variables providing each subspace's best result while considering its own boundary condition and the coupling variables provided by other subspaces. The outputs are new targets (i.e., shared design variables $\mathbf{z}$ and all coupling variables $\mathbf{y}$.

3.2 Subspace level problem

The subspace level is an independent optimization problem that is responsible for most of design decisions. The objective function includes two components: a portion of global objective and a quadratic measure of compatibility. The subspace level is defined as follows.

$$\begin{aligned} &\min \quad J_i = F(\mathbf{x}_s) + \lambda_c \sum (\mathbf{x}_s - \mathbf{z})^2 \\ &\text{s.t.} \quad \mathbf{g}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) \geq 0 \\ &\qquad \mathbf{y}_i = \mathbf{D}_i(\mathbf{x}_s, \mathbf{x}_i, \mathbf{y}_j) \quad i, j = 1, \ldots, n \quad i \neq j \end{aligned} \tag{2}$$

where $F(\mathbf{x})$ is the model of global objective. g_i is the local inequality constraint. $\mathbf{D}_i$ is the analysis or equality constraint. $\mathbf{y}_i$ and $\mathbf{y}_j$ are the coupling variables (responses of the discipline analysis). λ_c is the penalty parameter. n is the number of subspaces.

The global objective function is presented in the subspace level rather than the system level. This allows the subspace to have more dependable information to design the MDO variables. In addition, the compatibility of design variables (i.e., $\mathbf{x}_s$) are used to match the system level targets. The coupling variables (i.e., $\mathbf{y}$) are not considered in the compatibility. They are obtained only by each subspace's discipline (the input coupling variables could be taken out from the system level memory directly) while satisfying corresponding boundary conditions. The inputs of subspace level contain system level targets $\mathbf{z}$ and necessary coupling variables $\mathbf{y}_j$. The outputs are target responses, design variables $\mathbf{x}_s$ and coupling variables $\mathbf{y}_i$.

Note that each subspace does not require models of the constraints from all other subspaces. Besides, the information is not only shared indirectly through the system level targets, but also shared directly through the public storage of coupling variables. This direct interaction is easy to perform but has great valuable to improve the convergence efficiency, because the discipline i obtains its state variables by considering other disciplines' coupling variables, which carry the local constraints information of other disciplines (It shows the best attempt other discipline can provide). This additional information exchange performs better than CO with regards to cooperation. In CO, each subspace regards the coupling variables as unconstrained and leaves all the design variables and state variables to the system level to compromise. Fig. 2 shows the framework of ICO.

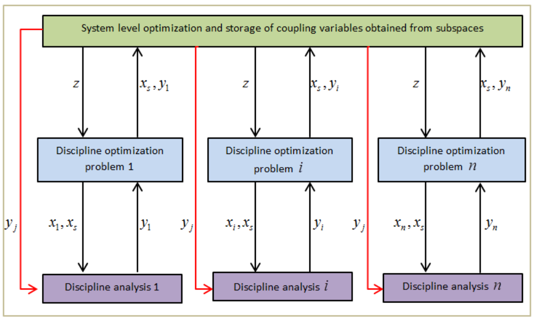

Figure 2. ICO framework

3.3 Solution process

This section presents the solution process of ICO (Improved Collaborative Optimization). ICO requires a two-step process to solve the MDO problem. In the first step, the initial system level targets $\mathbf{z}$ for shared variables and a set of initial coupling variables $\mathbf{y}$ are sent to each subspace. The subspace i treats the targets $\mathbf{z}$ and necessary coupling variable $\mathbf{y}_j \ (i \neq j)$ as parameters, allowing it to solve its optimization problem without requiring other subspaces' constraints or analysis information. The subspace i returns target responses $\mathbf{x}_s^*$ and the output of discipline analysis (coupling variable/state variable) $\mathbf{y}_i \ (i \neq j)$ to the system level. In the second step, the system level obtains the average of the target responses returned from the subspaces. Besides, it stores the coupling variables provided by the subspaces directly. The targets $\mathbf{z}$ and coupling variables $\mathbf{y}$ are then updated. The process is repeated until compatibility is realized.

4. APPLICATIONS

In this section, two examples are implemented to show the application of the proposed method. The efficiency and accuracy of the proposed method are compared with MDF and CO. All optimization problems are implemented by the standard Particle Swarm Optimization (PSO) method [9].

4.1 Example 1

The first example is taken from an academic case in literature by Alexandrov et al. [6]. This simple academic example shows some basic features of MDO problems, such as multiple interdependent disciplines and constraints. This example was used to present some properties of original collaborative optimization. The proposed method is evaluated and compared with collaborative optimization via this example. The formulation of problem is:

$$\begin{aligned}
&\min && f = \frac{1}{2}(y_1^2 + 10y_2^2 + 5(x_3 - 3)^2) \\
&\text{s.t.} && g_1 = x_1 + x_3 - 1 \le 0 \\
& && g_2 = x_2 - x_3 + 2 \le 0 \\
&\text{Analysis} && y_1 = \frac{1}{2}(x_1 - y_2) \\
& && y_2 = \frac{1}{2}(x_2 - y_1)
\end{aligned} \tag{3}$$

The test case has three design variables (two local design variables x_1, x_2 and one shared design variable x_3) and two coupling variables (y_1, y_2). The problem can be reformulated by ICO as follows:

System level optimization:

$$\begin{aligned}
&\min && f = [(z_3 - x_3^{(1)})^2] + [(z_3 - x_3^{(2)})^2] \\
&\text{s.t} && \text{No Constraints}
\end{aligned} \tag{4}$$

Subspace 1:

$$\begin{aligned}
&\min && f_1 = [\frac{1}{2}(y_1^2 + 10y_2^2 + 5(x_3 - 3)^2)] \\
& && + \lambda_c (x_3 - z_3)^2 \\
&\text{s.t} && g_1 = x_1 + x_3 - 1 \le 0 \\
&\text{Analysis} && y_1 = \frac{1}{2}(x_1 - y_2)
\end{aligned} \tag{5}$$

Subspace 2:

$$\begin{aligned}
&\min && f_2 = [\frac{1}{2}(y_1^2 + 10y_2^2 + 5(x_3 - 3)^2)] \\
& && + \lambda_c (x_3 - z_3)^2 \\
&\text{s.t} && g_2 = x_2 - x_3 + 2 \le 0 \\
&\text{Analysis} && y_2 = \frac{1}{2}(x_2 - y_1)
\end{aligned} \tag{6}$$

The optimization results are shown in Tab. 1. The optimization results by three methods are similar. However, the system iterations highlight great computational savings of 98% over the CO method. The ICO also converges more effectively than that of MDF, 80% computational savings. On the other hand, the number of design variables for the two decomposition methods are shown in Tab. 2. Compared with the design variables defined in CO, the ICO reduces the design variables by 8 (4 for system level and 2 for each subspace level). In summary, ICO method efficiently solves this test case while designing fewer variables for system and subspaces.

Table 1 Optimization results for example 1.

$\varepsilon = 0.001$ $\lambda_c = 0.1$	Initial value	MDF	CO	ICO
x_1	1	-1.9013	-1.9016	-1.9115
x_2	6	-1.009	-1.0133	-0.9550
x_3	-2	2.9301	2.9156	2.9549
y_1	3	-0.9017	-0.8977	-0.9557
y_2	-10	-0.9557	-0.0428	0.0000
f	567	0.4337	0.4299	0.4562
System iterations		31	483	6

Table 2 The number of design variables for example 1.

	CO	ICO
System	5	2
Subspace 1	4	3
Subspace 2	3	2

4.2 Example 2

The second example is another typical test case which is widely used for evaluation of different decomposition MDO algorithms. It was first introduced by Sellar .et al. [10]. The test case can be described as follows

$$
\begin{aligned}
&\min && f = x_2^2 + x_3 + y_1 + e^{-y_2} \\
&\text{s.t} && g_1 = (y_1 / y_{1a}) - 1 \geq 0 \\
& && g_2 = 1 - (y_2 / y_{2a}) \geq 0 \\
&\text{Bounds:} && -10 \leq x_1 \leq 10 \\
& && 0 \leq x_2 \leq 10 \\
& && 0 \leq x_3 \leq 10 \\
&\text{Analysis} && y_1 = x_1^2 + x_2 + x_3 - 0.2 y_2 \\
& && y_2 = y_1^{1/2} + x_1 + x_3 \\
&\text{parameters} && y_{1a} = 8 \\
& && y_{2a} = 24
\end{aligned} \tag{7}
$$

The test case has been successfully solved by using ICO. In this MDO problem, there are three shared design variables and two coupling variables, while it has no local variables in the subspaces. The problem can be reformulated by ICO as follows:

System level optimization:

$$
\begin{aligned}
&\min && f = [(z_1 - x_1^{(1)})^2 + (z_2 - x_2^{(1)})^2 + (z_3 - x_3^{(1)})^2] \\
& && \quad + [(z_1 - x_1^{(2)})^2 + (z_3 - x_3^{(2)})^2] \\
&\text{s.t} && \text{No Constraints}
\end{aligned} \tag{8}
$$

Subspace 1:

$$
\begin{aligned}
&\min && f_1 = [x_2^2 + x_3 + y_1 + e^{-y_2}] + \lambda_c \cdot [(x_1 - z_1)^2 \\
& && \quad + (x_2 - z_2)^2 + (x_3 - z_3)^2] \\
&\text{s.t} && g_1 = (y_1 / y_{1a}) - 1 \geq 0 \\
&\text{Bounds} && -10 \leq x_1 \leq 10 \\
& && 0 \leq x_2 \leq 10 \\
& && 0 \leq x_3 \leq 10 \\
&\text{Analysis} && y_1 = x_1^2 + x_2 + x_3 - 0.2 y_2
\end{aligned} \tag{9}
$$

Subspace 2:

$$
\begin{aligned}
&\min && f_2 = [x_2^2 + x_3 + y_1 + e^{-y_2}] \\
& && \quad + \lambda_c \cdot [(x_1 - z_1)^2 + (x_3 - z_3)^2] \\
&\text{s.t} && g_2 = 1 - (y_2 / y_{2a}) \geq 0 \\
&\text{Bounds} && -10 \leq x_1 \leq 10 \\
& && 0 \leq x_2 \leq 10 \\
& && 0 \leq x_3 \leq 10 \\
&\text{Analysis} && y_2 = y_1^{1/2} + x_1 + x_3
\end{aligned} \tag{10}
$$

The optimization results by three methods are compared in Tab. 3. The solutions of CO and ICO have the same accuracy. However, the number of system iterations using ICO is less than that using CO and MDF (98% over the CO and 67% over MDF). The reason is that two subspaces in ICO communicate with each other by exchanging coupling variables, which contains the local constraints information. Besides, each subspace has the global objective function in its optimization problem, leading a great efficiency to solve the problem synchronously. The number of design variables for CO and ICO methods are shown in Tab. 4. Compared with the design variables defined in CO, the ICO reduces the design variables by 4 (2 for system level and 1 for each subspace level).

Table 3 Optimization results for example 2.

$\varepsilon = 0.001$ $\lambda_c = 0.1$	Initial value	MDF	CO	ICO
x_1	1	3.0269	3.0235	3.0375
x_2	0	0.0087	0.0058	0.0146
x_3	5	0.0000	0.0696	0.0000
y_1	5	8.0000	8.0000	8.0000
y_2	0	5.8554	5.7411	5.8788
f	11	8.0029	8.0728	8.0029
System iterations		30	500	10

Table 4 The number of design variables for example 2.

Column 1	CO	ICO
System	5	3
Subspace 1	4	3
Subspace 2	3	2

5. CONCLUSIONS

This paper introduces the improved collaborative optimization (ICO). ICO builds on existing decomposition methods such as collaborative optimization and enhanced collaborative optimization. Compared with the standard CO and more advanced ECO methods, there are two improvements in ICO. One is that ICO enhances the subspace design authority. The relevant portions of the global objective are transmitted into each subspace. The subspaces are then responsible for most of the design decisions and the system is limited to providing dynamic 'moving limits' for shared variables. This improvement enhances the design authority of individual, while maintaining an easy coordination of design variables for system level problem. The other is that, ICO provides an additional direct interaction channel for information interaction among subspaces. Compared with original formulation of CO, information is only shared indirectly through the system level targets. Each subspace has no direct information exchange among other subspaces. This mere system-subsystem information flow causes low efficiency for complex MDO problems, especially when coupling variables increase. Compared with formulation of ECO, each subspace requires the models of all other subspaces' constraints, which seems impractical for information interaction among subspaces. Building the models of other subspaces' constraints before each iteration is also a waste of time and computational source. However, ICO provides an additional information channel for each subspace to exchange the coupling variables. This new information flow allows each subspace has some knowledge of other subspaces' constraints and disciplines, helping the subspace discipline to make more reasonable attempts for the targets. This leads to great computational savings for MDO problems. Results from examples suggest that ICO yields significant computational savings and simplicity, relative to collaborative optimization and enhanced collaborative optimization.

ACKNOWLEDGMENTS

This research is partially supported by the Natural Science Foundation of China (NSFC, 51379176, 51679201) and International Cooperation Training Program for the Innovative Talents of China Scholarship Council.

REFERENCES

1. Martins, J. R., & Lambe, A. B. (2013). Multidisciplinary design optimization: a survey of architectures. AIAA journal, 51(9), 2049-2075.

2. Kroo, I., Altus, S., Braun, R., Gage, P., & Sobieski, I. (1994, September). Multidisciplinary optimization methods for aircraft preliminary design. In 5th symposium on multidisciplinary analysis and optimization (p. 4325).

3. Braun, R. D., Moore, A. A., & Kroo, I. M. (1997). Collaborative approach to launch vehicle design. Journal of spacecraft and rockets, 34(4), 478-486.

4. Cai, G., Fang, J., Zheng, Y., Tong, X., Chen, J., & Wang, J. (2010). Optimization of system parameters for liquid rocket engines with gas-generator cycles. Journal of Propulsion and Power, 26(1), 113-119.

5. Allison, J., Roth, B., Kokkolaras, M., Kroo, I., & Papalambros, P. (2006, September). Aircraft family design using decomposition-based methods. In 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (p. 6950).

6. Alexandrov, N. M., & Lewis, R. M. (2002). Analytical and computational aspects of collaborative optimization for multidisciplinary design. AIAA journal, 40(2), 301-309.

7. Sobieski, I., Manning, V., & Kroo, I. (1998). Response surface estimation and refinement in collaborative optimization. In 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization (p. 4753).

8. Roth, B., & Kroo, I. (2008). Enhanced collaborative optimization: application to an analytic test problem and aircraft design. In 12th AIAA/ISSMO multidisciplinary analysis and optimization conference (p. 5841).

9. Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760-766). Springer, Boston, MA.

10. Sellar, R., Batill, S., & Renaud, J. (1996). Response surface based, concurrent subspace optimization for multidisciplinary system design. In 34th Aerospace Sciences Meeting and Exhibit (p. 714).