

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Concurrent Subspace Optimization for Aircraft System Design

Ke-shi Zhang

*Northwestern Polytechnical University
China*

1. Introduction

Concurrent Subspace Optimization (CSSO) is one of the main decomposition approaches in Multidisciplinary Design Optimization (MDO). It supports a collaborative and distributed multidisciplinary design optimization environment among different disciplinary groups. Sobieski first proposed the subspace optimization method (Sobieszczanski-Sobieski, 1988), and Sobieski's blueprint was further developed by Bloebaum and subsequently named the concurrent subspace optimization method (Bolebaum, 1991). Renaud developed a second-order variant of the Global Sensitivity Equation (GSE) method and an alternative potential coordination procedure for the CSSO method (Renaud & Gabriele, 1993a, 1993b, 1994). Sellar proposed to replace GSE with the neutral-network based response surface method (Sellar et al., 1996).

The CSSO method allows a complex couple system to be decomposed into smaller, temporarily decoupled subsystems, each corresponding to different disciplines (subspaces). Each subspace optimization minimizes the system objective function subject to its own constraints as well as constraints contributed from the other subspaces. Each subspace optimization use its own high-fidelity analysis tools as well as given surrogate models or low-fidelity analysis tool provided by the other subspaces for analysis. Subsequently, the subspace optimizations can be performed concurrently. The system-level coordination optimization will be implemented completely based on approximation analysis tools. The subspace optimizations and the coordination optimization will be alternatively performed until results are finally decided by the coordination optimization. Therefore, the CSSO method is particularly suited to applications in a design organization where tasks are distributed among different design groups.

The CSSO method was developed initially for a single objective MDO problem. However, most MDO problems are essentially multi-objective. In recent years more work (Aute & Azarm, 2006; Huang & Bloebaum, 2004; McAllister et al., 2000; McAllister et al., 2004; Orr & Hajela, 2005; Parashar & Bloebaum, 2006; Tappeta & Renaud, 1997; Zhang et al., 2008) has focused on extending existing MDO method to handle such multi-objective MDO problems, by means of integrating a multi-objective optimization method within the MDO framework. This kind of method can be called a multi-objective MDO method.

It is an effective way to integrate multi-objective optimization method within the CSSO framework to develop the multi-objective MDO method. CSSO was extended to solve multi-objective MDO problems, including the Multi-objective Pareto CSSO (MOPCSSO) method,

the Multi-objective Range CSSO (MORCSSO) method, the Multi-objective Target CSSO (MOTCSSO) method, the Multi-objective Genetic Algorithm CSSO (MOGACSSO) method and Adaptive Weighted Sum based CSSO (AWSCSSO). In MOPCSSO the Constraint method is integrated within CSSO framework (Huang & Bloebaum, 2007). In MORCSSO and MOTCSSO the concept of designer preference is introduced (Huang & Bloebaum, 2004). In MOGACSSO the Genetic Algorithm is combined with CSSO and in the hope of improving the computational efficiency (Parashar & Bloebaum, 2006). In AWSCSSO the Adaptive Weighted Sum method is introduced into CSSO (Zhang et al., 2008).

2. General description of MDO problem for aircraft system design

Aircraft is a complex engineering system in which multiple disciplines (such as aerodynamics, structure, thrust, noise, electronics, cost, et al.) are included. Actually different disciplines are not independent of each other. For example, deformation of a wing structure affects the aerodynamic lift distribution on the wing and in turn a new deformation is caused, which is the well-known aeroelastic problem. The designers of each discipline cannot work without consideration of other disciplines, which makes the aircraft system design become complicated. Especially in the aircraft preliminary design, the specialists of different disciplines have to often work together and discuss with each other to decide many design variables so that higher performance can be achieved. Aircraft system design is a typical multidisciplinary design problem.

In recent years, industry has paid more attention to improving efficiency in the design of complex systems, such as aircraft. MDO has emerged as an engineering discipline that focuses on the development of new design and optimization strategies for the complex systems. MDO researchers strive to reduce the time and cost associated with the coupling interaction among several disciplines. "Decomposition approaches provide many advantages for the solution of complex MDO problems, as they enable a partitioning of a large coupled problem into smaller, more manageable sub-problems. The resulting computational benefits, besides the obvious one associated with the solution of smaller problems, include creating a potential distributed processing environment. The primary benefit, however, pertains to the savings in personal hours, because groups are no longer required to wait around for other groups in the process to complete their design tasks." (Huang & Bloebaum, 2007)

Mathematically for minimization problems the general form for MDO can be represented as follows:

$$\begin{aligned}
 &\text{Min } F(X, Y_1, \dots, Y_N) \\
 &\text{s.t. } G_i(X, Y_1, \dots, Y_N) \leq 0 \\
 &\quad Y_i = f_i(X, Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_N) \\
 &\quad i = 1, 2, \dots, N \\
 &\quad X = \{x_1, x_2, \dots, x_{NV}\}
 \end{aligned} \tag{1}$$

Where F is the objective function vector that is composed of one or more objectives, G_i is the constraint vector provided by subsystem i , Y_i is the coupling vector of subsystem i , and X is the design vector. The objective function and the constraints can be expressed by the

function of X and Y_i . In this chapter the MDO problem expressed in Eq. (1) will be taken as the example to discuss the CSSO methods.

3. Different frameworks for concurrent subspace optimization

Global Sensitivity Equation based CSSO and Response Surface based CSSO will be discussed in this chapter as the typical CSSO methods.

3.1 Global Sensitivity Equation based CSSO (GSECSSO)

GSECSSO is a bi-level optimization method. As an example, the GSECSSO method for a problem with subsystem 1 and 2 (Eq. (1): one objective and two coupled subsystems) is stated in following paragraphs.

The mathematical models of subspace optimizations of GSECSSO can be written as

| Sub-optimization 1 | Sub-optimization 2 |
|---|---|
| $\text{Min } F(X_1, Y_1, \hat{Y}_2)$ | $\text{Min } \hat{F}(X_2)$ |
| $\text{s.t. } C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(s_1(1-r_1^1) + (1-s_1)t_1^1)$ | $\text{s.t. } \hat{C}_1(X_2) \leq C_1^0(s_1(1-r_2^1) + (1-s_1)t_2^1)$ |
| $\hat{C}_2(X_1) \leq C_2^0(s_2(1-r_1^2) + (1-s_2)t_1^2)$ | $C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(s_2(1-r_2^2) + (1-s_2)t_2^2)$ |
| $Y_1 = f_1(X_1, \hat{Y}_2)$ | $Y_2 = f_2(X_2, \hat{Y}_1)$ |

Where X_i , Y_i and C_i are design variable vector, state variable vector and cumulative constraint of subspace i , respectively. C_i^0 is the value of C_i at the starting point X^0 . The value with symbol '^' above is a linearly approximated one. r_k^p , t_k^p and s_p are responsibility, trade-off and switch coefficients, respectively. In the first iteration, the responsibility coefficients are initialized using the sensitivities of cumulative constraints with respect to design variables (Bolebaum, 1991) and the switch coefficients are set to one. In the following iterations the coordination coefficients are decided by the coordination optimizations.

By means of Kresselmeier-Steinhauser (KS) function (Kreisselmeier & Steinhauser, 1979), the cumulative constraint, C_i , which represents the constraints of subsystem i , is expressed as

$$C_i = g_{i\max} + \frac{1}{\rho} \ln \left(\sum_{k=1}^m \exp(\rho(g_{ik}(X) - g_{i\max})) \right) \quad (3)$$

Where $[g_{i1}, g_{i2}, \dots, g_{im}] = G_i$ and ρ is a positive user-prescribed value.

Global Sensitivity Equation (GSE) (Sobieszczanski-Sobieski, 1990) is a method for computing sensitivity derivatives of state (output) variables with respect to independent (input) variables for complex, internally coupled systems, while avoiding the cost and inaccuracy of finite differencing performed on the entire system analysis. By using GSE expressed below, the global sensitivities are calculated, by

$$\begin{bmatrix} I & -\frac{\partial Y_1}{\partial Y_2} \\ -\frac{\partial Y_2}{\partial Y_1} & I \end{bmatrix} \begin{bmatrix} \frac{dY_1}{dX} \\ \frac{dY_2}{dX} \end{bmatrix} = \begin{bmatrix} \frac{\partial Y_1}{\partial X} \\ \frac{\partial Y_2}{\partial X} \end{bmatrix} \quad (4)$$

The responsibility coefficients divide the responsibility of satisfying constraints among all participating subspaces. r_k^p represents the responsibility allocated to the k -th subspace for satisfying the p -th cumulative constraint. In the first iteration, they are initialized with the sensitivities of cumulative constraints with respect to design variables (Bolebaum, 1991). The trade-off coefficients allow for the violation of a constraint in one subspace optimization in order to gain a large reduction of the objective function. The switch coefficients enable or disable the responsibility coefficients and the trade-off coefficients decided by whether the cumulative constraints are satisfied.

The mathematical model of coordination optimization of GSECSSO can be written as

$$\begin{aligned} \text{Min } F(\mathbf{r}, \mathbf{t}) &= F^* + \sum_{p=1}^2 \sum_{k=1}^2 \frac{dF}{dr_k^p} \bigg|_{\mathbf{X}=\mathbf{X}^*} \Delta r_k^p + \sum_{p=1}^2 \sum_{k=1}^2 \frac{dF}{dt_k^p} \bigg|_{\mathbf{X}=\mathbf{X}^*} \Delta t_k^p \\ \text{s.t. } \sum_{k=1}^2 r_k^p &= 1, \quad \sum_{k=1}^2 t_k^p = 0 \\ 0 \leq r_k^p &\leq 1, \quad -1 \leq t_k^p \leq 1, \quad p, k = 1, 2 \end{aligned} \quad (5)$$

Where F^* is the objective value at \mathbf{X}^* (\mathbf{X}^* is the value of design variable vector after subspace optimizations). $\frac{dF}{dr_k^p} \bigg|_{\mathbf{X}=\mathbf{X}^*}$ and $\frac{dF}{dt_k^p} \bigg|_{\mathbf{X}=\mathbf{X}^*}$ are the optimum sensitivity of F with respect to responsibility and trade-off coefficients.

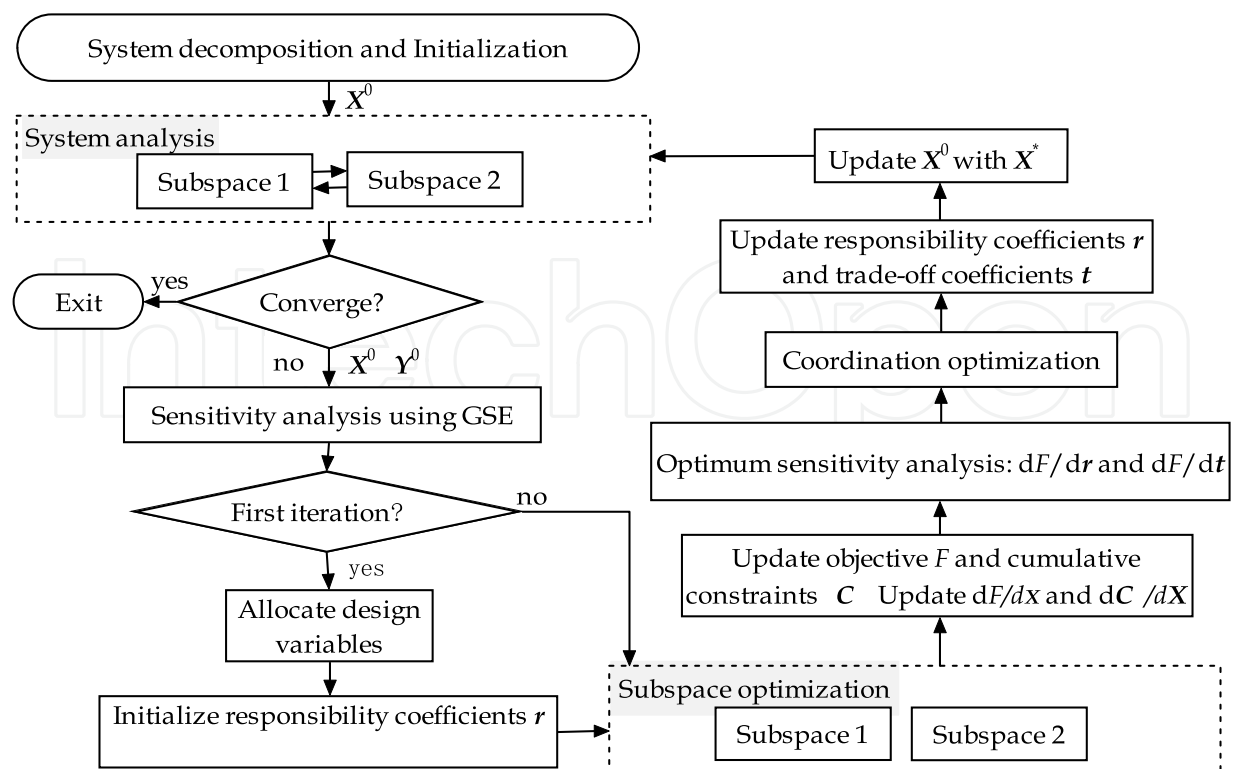


Fig. 1. Flowchart of GSECSSO method

The flowchart of GSECSSO is shown in Fig. 1. The original problem is decomposed and the mathematical models of subspace optimizations and coordination optimization are established as Eq. (2) and Eq. (5). Based on the initial design X^0 , the system analysis is performed and the GSE is used to find the system derivatives. At the first iteration, the design variables are allocated to the appropriate subspaces, as well as the responsibility, trade-off and switch coefficients, r , t , s , are initialized. Subsequently, each subspace optimization minimizes the system objective function subject to its own constraints as well as constraints contributed from the other subspaces. The subspace optimizations are performed concurrently with respect to a disjoint subset of design variables. The updated design vector is the simple combination of local optimal design sub-vectors. After that the global sensitivities and optimum sensitivities are updated. Then the system-level coordination optimization is implemented to optimize coordination parameters.

In GSECSSO, the subspace optimizations are implemented concurrently with respect to a disjoint subset of design variables, which substantially reduces the complexity of the optimization problem within each disciplinary group. The updated design vector is the simple combination of local optimal design sub-vectors. This provides designers with a significant potential benefit in terms of computational effort. However, sometimes the convergence of GSECSSO is oscillatory and premature. It is due to that the trade-off coefficients and the KS parameter ρ with inappropriate value may lead to bad convergence (Huang & Bloebaum, 2004) as main reasons.

3.2 A variant of GSECSSO

The variant of GSECSSO (Huang & Bloebaum, 2004) is much more efficient than the original GSECSSO. It is adopted in multiple multi-objective CSSO methods (Huang & Bloebaum, 2004; Parashar & Bloebaum, 2006; Zhang et al., 2008).

In the variant of GSECSSO, several modifications are made: (1) the trade-off coefficients are abandoned since all trade-offs will occur directly with minimization of the objective functions within each subspace; (2) The KS parameter, ρ , is set to be increased from a small value; (3) an infeasibility minimization is appended to move the point into the feasible region; (4) the coordination optimization are abandoned and the responsibility coefficients are directly calculated via their initialization strategy.

Taking the MDO problem in Eq. (1) (one objective, two coupled subsystems, i.e. $N=2$) as an example, the mathematical models of subspace optimizations for the modified CSSO can be written as

$$\begin{array}{c|c}
 \text{Sub-optimization 1} & \text{Sub-optimization 2} \\
 \hline
 \text{Min } F(X_1, Y_1, \hat{Y}_2) & \text{Min } \hat{F}(X_2) \\
 \text{s.t. } C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1) & \text{s.t. } \hat{C}_1(X_2) \leq C_1^0(1 - r_2^1) \\
 \hat{C}_2(X_1) \leq C_2^0(1 - r_1^2) & C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2) \\
 Y_1 = f_1(X_1, \hat{Y}_2) & Y_2 = f_1(X_2, \hat{Y}_1)
 \end{array} \quad (6)$$

Where X_i , Y_i and C_i are design variable vector, state variable vector and cumulative constraint of subspace i , respectively. C_i^0 is the value of C_i at the starting point X^0 . The value with symbol '^' above is a linearly approximated one. r_k^p is responsibility coefficients.

the flowchart of modified CSSO method is shown in Fig. 2. In the first stage, based on the initial design variables the system analysis is implemented. If the initial scenario does not satisfy the constraints, a minimization will be required to reduce initial infeasibility of the constraints as much as possible. In the subsequent stage, all the sensitivities are computed by using GSE. Based on the sensitivity information, the impacts of design variables upon each subspace can be analyzed and the design variables will be allocated to the subspace of the greatest impact. In the third stage, the subspace optimizations are performed concurrently. There is no system optimization in this method. The updated design vector is the simple combination of the local optimal design sub-vectors.

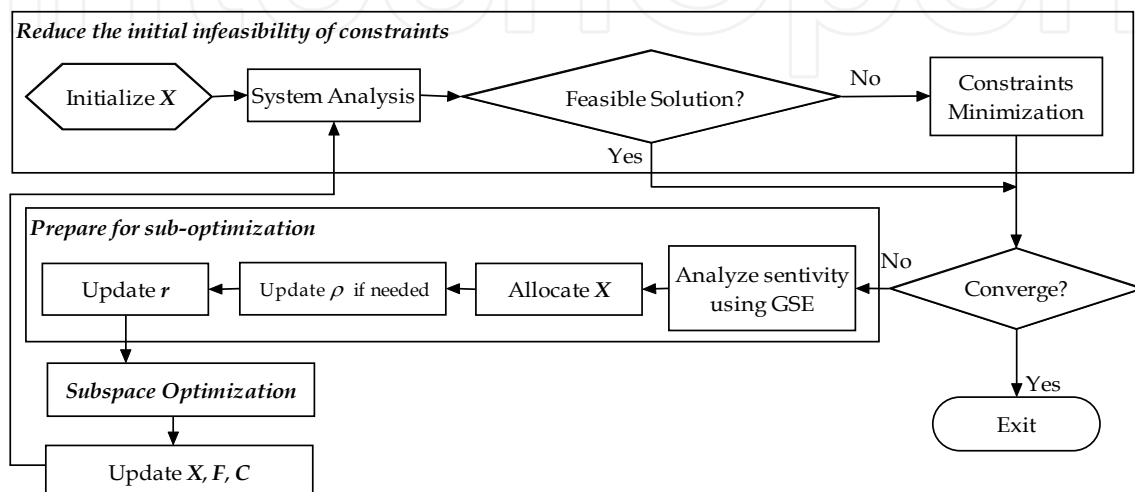


Fig. 2. Flowchart of the modified CSSO method

The variant of GSECSSO behaves better than GSECSSO on convergence performance. Nevertheless, how to set the start value and increase step of the KS parameter is still hard to say. These two values will affect convergence to some extent and need to be further investigated. The constraint minimization will bring extra computation cost. Furthermore, the linear approximation may sometimes cause oscillatory and premature convergence.

3.3 Response Surface based CSSO (RSCSSO)

The RSCSSO method performs optimization in bi-level framework. Taking the MDO problem in Eq. (1) (one objective, two coupled subsystems, i.e. $N=2$) as an example, the mathematical models of subspace optimizations for RSCSSO can be written as

$$\begin{array}{c|c}
 \text{Sub-optimization 1} & \text{Sub-optimization 2} \\
 \hline
 \text{Min } F(X_1, Y_1, \hat{Y}_2) & \text{Min } F(X_2, \hat{Y}_1, Y_2) \\
 \text{s.t. } G_1(X_1, Y_1, \hat{Y}_2) \leq 0 & \text{s.t. } G_1(X_2, \hat{Y}_1, Y_2) \leq 0 \\
 G_2(X_1, Y_1, \hat{Y}_2) \leq 0 & G_2(X_2, \hat{Y}_1, Y_2) \leq 0 \\
 Y_1 = f_1(X_1, \hat{Y}_2) & Y_2 = f_2(X_2, \hat{Y}_1)
 \end{array} \quad (7)$$

Where X_i and Y_i are design variable vector and state variable vector of subspace i , respectively. In subspace i , Y_i is calculated using the high-fidelity analysis tool while the

other state variables with a symbol ‘^’ above are approximated by a quadratic response surface. Actually, other surrogate modeling techniques, such as kriging model or radial basis functions, can be used for approximation as well.

The mathematical model of the coordination optimization of RSCSSO can be written as

$$\begin{aligned} &\text{Min } F\left(\mathbf{X}, \hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2\right) \\ &\text{s.t. } G_1\left(\mathbf{X}, \hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2\right) \leq 0 \\ &\quad G_2\left(\mathbf{X}, \hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2\right) \leq 0 \end{aligned} \tag{8}$$

In the system-level optimization, all the state variables are approximately calculated and the design variables of all subspaces are optimized. The flowchart of RSCSSO is shown in Fig. 3. Firstly, the original problem is decomposed into subspace optimizations in Eq. (7) and coordination optimization in Eq. (8). Then couples of sample points of the design variable vector are generated via Design of Experiment (DOE) method, such as the orthogonal experiment design. The system analysis will be performed based on these data and the results will be stored in the database.

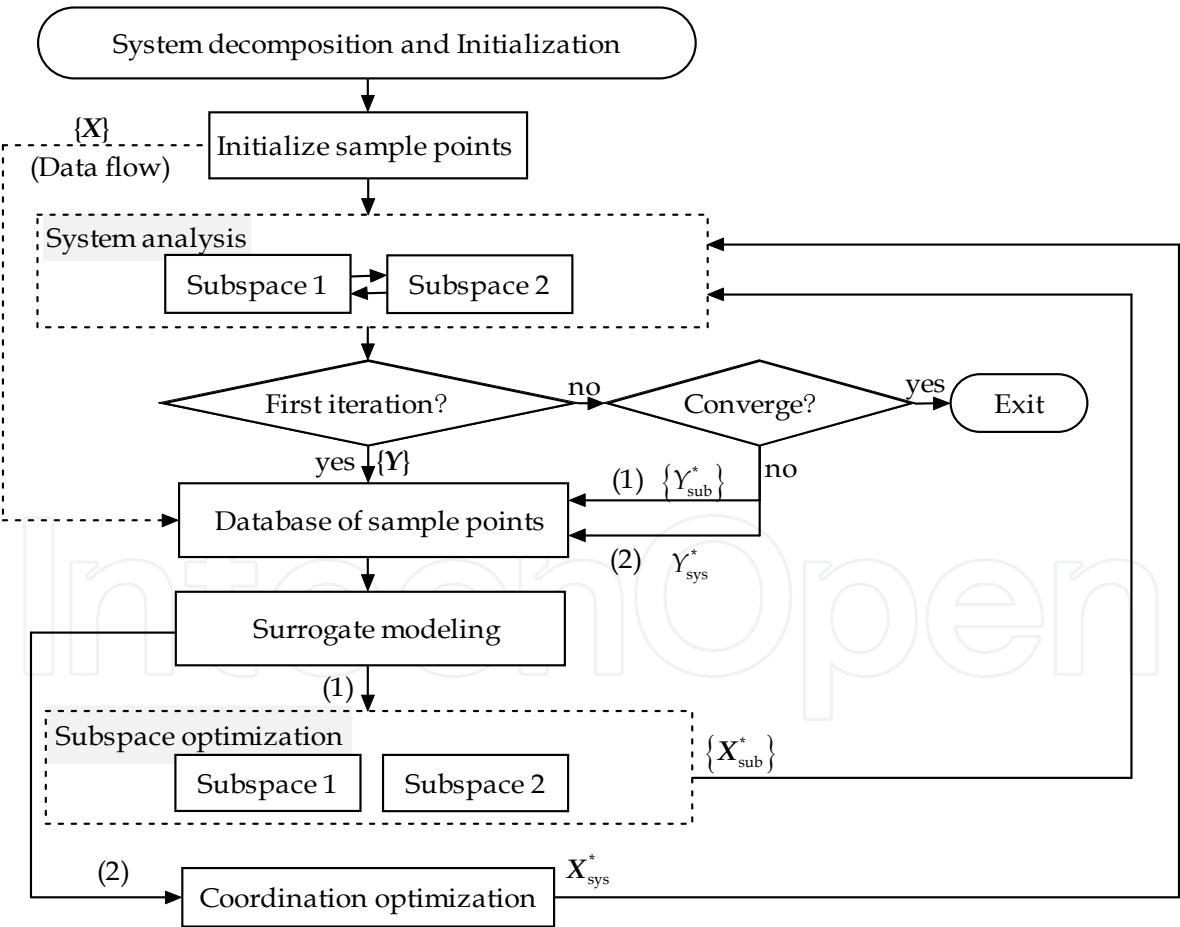


Fig. 3. Flowchart of RSCSSO method

Subsequently these data will be used to generate the response surface models for the state variables. After that the subspace optimizations are performed simultaneously, the results of

which will be analyzed and augmented into the database for updating the response surface models, as flow (1) in Fig. 3. Then all of the design variables will be optimized in the coordination optimization and the result will be used to update response surface models as well, as flow (2) in Fig. 3. The subspace optimizations and the coordination optimization are alternately performed until stable convergence is achieved.

In the RSCSSO method, the subspace optimizations are to generate sufficient design information for approximation. After the concurrent subspace optimizations, a global approximation problem is formulated about the current design vector using information stored in the design database. It is the coordination procedure of the global approximation that drives constraint satisfaction and the overall system optimization. A fast and robust convergence appears in the RSCSSO method. However, with an increase in design variables the sample points needed for creating response surface models will be greatly increased. Furthermore, augmenting optimal points is not so reasonable for improving response surface models. As the final result is completely decided by the coordination optimization, the subspace optimizations have little impact on the results and can be abandoned.

3.4 Examples

3.4.1 Example 1: An analytical example with coupling relationship

The mathematical model for an analytical MDO example considering two coupling disciplines is of the form

$$\begin{aligned}
 \text{Min } f &= x_2^2 + x_3 + y_1 + e^{-y_2} \\
 \text{s.t. } g_1 &= 8 - y_1 \leq 0 \\
 g_2 &= y_2 - 10 \leq 0 \\
 y_1 &= x_1^2 + x_2 + x_3 - 0.2y_2 \\
 y_2 &= \sqrt{y_1} + x_1 + x_3 \\
 x_1 &\in [-10, 10], x_2, x_3 \in [0, 10]
 \end{aligned} \tag{9}$$

Where y_1 and y_2 are state variables belonging to subspace 1 and subspace 2, respectively; g_1 and g_2 are constraints provided by discipline 1 and discipline 2, respectively. The coupling relationship between discipline 1 and discipline 2 is depicted in Fig. 4.

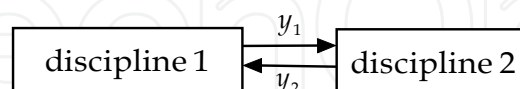


Fig. 4. Coupling relationship between discipline 1 and discipline 2 for a numerical example

The comparison of the results obtained by different methods is listed in Table 1. Sequential Quadratic Programming (SQP) method is taken as a reference for the convenience of this comparative study. From Table 1, several conclusions can be drawn as follows: (1) the system-level analysis can be remarkably reduced when using different CSSO methods, which shows that the CSSO methods are well suited for MDO problems; (2) the variant of GSECSSO method outperforms the RSCSSO method, in view of the optimization results as well as the required number of system- and disciplinary-level analysis.

| variables | SQP | Variant of GSECSSO | RSCSSO |
|---------------------------------|----------|--------------------|----------|
| f | 8.002860 | 8.002943 | 8.031944 |
| x_1 | 3.028427 | 3.028442 | 2.993963 |
| x_2 | 0.000000 | 0.000000 | 0.192359 |
| x_3 | 0.000000 | 0.000000 | 0.000000 |
| Number of system analysis | 122 | 7 | 38 |
| Number of discipline 1 analysis | 0 | 72 | 179 |
| Number of discipline 2 analysis | 0 | 64 | 144 |

Table 1. Comparison of different CSSO methods with direct SQP for an analytical optimization problem

3.4.2 Example 2: Gear reducer optimization

This example is taken from the reference (Azarm & Li, 1989). The object is to minimize the overall weight, subject to the constraints for bending and torque stresses. The mathematical model for the optimization problem is as following:

Min

$$f = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.477\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$$

s.t.

$$g_1 = 27x_1^{-1}x_2^{-2}x_3^{-1} - 1 \leq 0$$
$$g_2 = 397.5x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0$$
$$g_3 = 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} - 1 \leq 0$$
$$g_4 = 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} - 1 \leq 0$$
$$g_5 = 10x_6^{-3}\sqrt{\left(745x_2^{-1}x_3^{-1}x_4\right)^2 + 1.69 \times 10^7} - 1100 \leq 0$$
$$g_6 = 10x_7^{-3}\sqrt{\left(745x_2^{-1}x_3^{-1}x_5\right)^2 + 1.575 \times 10^8} - 850 \leq 0$$
$$g_7 = x_4^{-1}\left(1.5x_6 + 1.9\right) - 1 \leq 0$$
$$g_8 = x_5^{-1}\left(1.1x_7 + 1.9\right) - 1 \leq 0$$
$$g_9 = x_2x_3 - 40 \leq 0$$
$$g_{10} = 5 - x_1x_2^{-1} \leq 0$$
$$g_{11} = x_1x_2^{-1} - 12 \leq 0$$
$$x_1 \in [2.6, 3.6] \quad x_2 \in [0.7, 0.8] \quad x_3 \in [17, 28]$$
$$x_4, x_5 \in [7.3, 8.3] \quad x_6 \in [2.9, 3.9] \quad x_7 \in [5.0, 5.5]$$

(10)

The optimization problem is classified into two disciplines: bearing discipline and shaft discipline, without any coupling relationship.

The optimization problem is solved by SQP and different CSSO methods. The results are depicted in Table 2. The convergence histories are shown in Fig. 5. From this comparative study, some conclusions can be drawn as follows: 1) Both GSECSSO and the variant of GSECSSO enable the reduction of system analysis, which in turn improve the efficiency; 2) No reduction of the system analysis is achieved by using RSCSSO. Nevertheless, the optimization procedure as well as the data-flow interface is much simpler, and the optimization can be potentially improved by implementing parallel system analysis for sample points and parallel subspace optimization; 3) The variant of GSECSSO method outperforms the GSECSSO method.

| Variables | SQP | GSECSSO | Variant of GSECSSO | RSCSSO |
|---------------------------------------|-------------|-------------|--------------------|-------------|
| f | 2994.341316 | 2995.606759 | 2995.607422 | 2994.193811 |
| x_1 | 3.5 | 3.500000 | 3.500000 | 3.500021 |
| x_2 | 0.7 | 0.7 | 0.7 | 0.700000 |
| x_3 | 17 | 17 | 17 | 17.000000 |
| x_4 | 7.3 | 7.300000 | 7.3 | 7.300000 |
| x_5 | 7.715320 | 7.733301 | 7.733330 | 7.715316 |
| x_6 | 3.350215 | 3.352131 | 3.352156 | 3.349631 |
| x_7 | 5.286654 | 5.287256 | 5.287246 | 5.286643 |
| Number of system analysis | 40 | 15 | 11 | 51 |
| Number of bearing-discipline analysis | 0 | 326 | 188 | 280 |
| Number of shaft-discipline analysis | 0 | 2986 | 1818 | 288 |

Table 2. Comparison of different CSSO methods with direct SQP for gear-box optimization

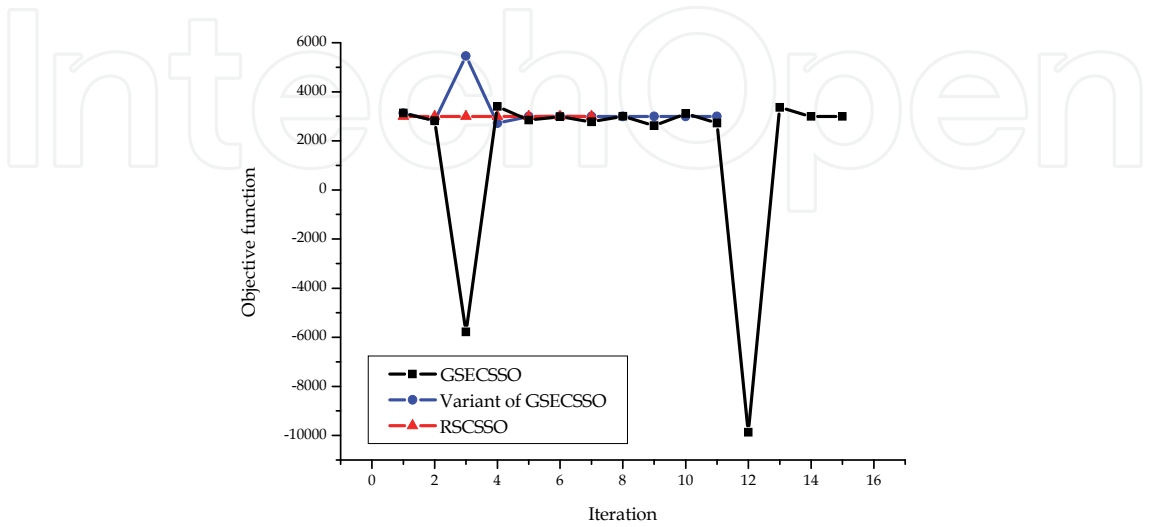


Fig. 5. Convergence history of CSSO optimization for gear reducer

4. Different frameworks for multi-objective concurrent subspace optimization

The MOPCSSO, MORCSSO and AWSCSSO methods will be discussed in this chapter as the typical multi-objective CSSO methods.

4.1 Multi-objective Pareto (MOPCSSO)

The constraint method is an effective multi-objective optimization method which optimizes preferred objective with others treated as constraints. MOPCSSO is developed by introducing the constraint method into the variant of GSECSSO.

Taking the MDO problem in Eq. (1) (two objectives, two coupled subsystems, i.e. $N=2$) as an example, the mathematical models of subspace optimizations of MOPCSSO can be written as

$$\begin{array}{c|c}
 \text{Sub-optimization 1} & \text{Sub-optimization 2} \\
 \hline
 \begin{array}{l}
 \text{Min } F_1(X_1, Y_1, \hat{Y}_2) \\
 \text{s.t. } C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1-r_1^1) \\
 \hat{C}_2(X_1) \leq C_2^0(1-r_1^2) \\
 \hat{F}_2 \leq F_2^0 \\
 Y_1 = f_1(X_1, \hat{Y}_2)
 \end{array} &
 \begin{array}{l}
 \text{Min } F_2(X_2, \hat{Y}_1, Y_2) \\
 \text{s.t. } \hat{C}_1(X_2) \leq C_1^0(1-r_2^1) \\
 C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1-r_2^2) \\
 \hat{F}_1 \leq F_1^0 \\
 Y_2 = f_1(X_2, \hat{Y}_1)
 \end{array}
 \end{array} \quad (11)$$

The variant of GSECSSO method described in Sub-section 3.2 is adopted in the single objective optimization for each objective function. The flowchart of MOPCSSO is same as that in Fig. 2. During the system optimization, all objective functions are improved simultaneously or individual objective function are improved without worsening the others. The optimization continues until no further improvement can be made so that the Pareto optimum can be obtained finally.

4.2 Multi-objective Range CSSO (MORCSSO)

The goal programming method is one of the most popular multi-objective optimization techniques that consider designer's preference. MORCSSO and Multi-objective Target CSSO (MOTCSSO) are both developed by combining idea of the goal programming method with MOPCSSO.

The $[F_2^{\min}, F_2^{\max}]$ is supposed to be the preferred range of objective function F_2 . The MDO problem in Eq. (1) (two objectives, two coupled subsystems, i.e. $N=2$) is taken as an example. The framework of MORCSSO is shown in Fig. 6. The variant of GSECSSO method described in Sub-section 3.2 is adopted for sub-optimizations and system-level coordination. The optimization is performed in two stages to obtain a preferred Pareto point.

1. If the design point does not meet the preference at beginning, the first-stage optimizations are implemented to obtain a design in the preferred range from the starting point.

The mathematical models of subspace optimizations of MORCSSO can be written as

If $F_2^0 < F_2^{\min}$,

| Sub-optimization 1 | Sub-optimization 2 |
|---|--|
| Min $F_1(X_1, Y_1, \hat{Y}_2)$ | Min $ F_2(X_2, \hat{Y}_1, Y_2) - F_2^{\min} $ |
| s.t. $C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1)$ | s.t. $\hat{C}_1(X_2) \leq C_1^0(1 - r_2^1)$ |
| $\hat{C}_2(X_1) \leq C_2^0(1 - r_1^2)$ | $C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2)$ |
| $-\hat{F}_2 \leq -F_2^0$ | $Y_2 = f_1(X_2, \hat{Y}_1)$ |
| $Y_1 = f_1(X_1, \hat{Y}_2)$ | |

(12)

If $F_2^0 > F_2^{\max}$,

| Sub-optimization 1 | Sub-optimization 2 |
|---|--|
| Min $F_1(X_1, Y_1, \hat{Y}_2)$ | Min $ F_2(X_2, \hat{Y}_1, Y_2) - F_2^{\max} $ |
| s.t. $C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1)$ | s.t. $\hat{C}_1(X_2) \leq C_1^0(1 - r_2^1)$ |
| $\hat{C}_2(X_1) \leq C_2^0(1 - r_1^2)$ | $C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2)$ |
| $\hat{F}_2 \leq F_2^0$ | $Y_2 = f_1(X_2, \hat{Y}_1)$ |
| $Y_1 = f_1(X_1, \hat{Y}_2)$ | |

(13)

2. In the second stage the design point is optimized closer to the Pareto frontier gradually within the preferred range.

After the optimization in the first stage, the design is in the designer's preferred objective range. Then the mathematical models of subspace optimizations of MORCSSO is same as those of MOPCSSO in Eq. (11).

In the course of optimization in the second stage, the design point maybe flees out of the preferred objective range again. In such a case, the optimization below should be performed.

If $F_2^0 < F_2^{\min}$,

| Sub-optimization 1 | Sub-optimization 2 |
|---|--|
| Min $F_1(X_1, Y_1, \hat{Y}_2)$ | Min $ F_2(X_2, \hat{Y}_1, Y_2) - F_2^{\min} $ |
| s.t. $C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1)$ | s.t. $\hat{C}_1(X_2) \leq C_1^0(1 - r_2^1)$ |
| $\hat{C}_2(X_1) \leq C_2^0(1 - r_1^2)$ | $C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2)$ |
| $-\hat{F}_2 \leq -F_2^0$ | $\hat{F}_1 \leq F_1^0$ |
| $Y_1 = f_1(X_1, \hat{Y}_2)$ | $Y_2 = f_1(X_2, \hat{Y}_1)$ |

(14)

If $F_2^0 > F_2^{\max}$,

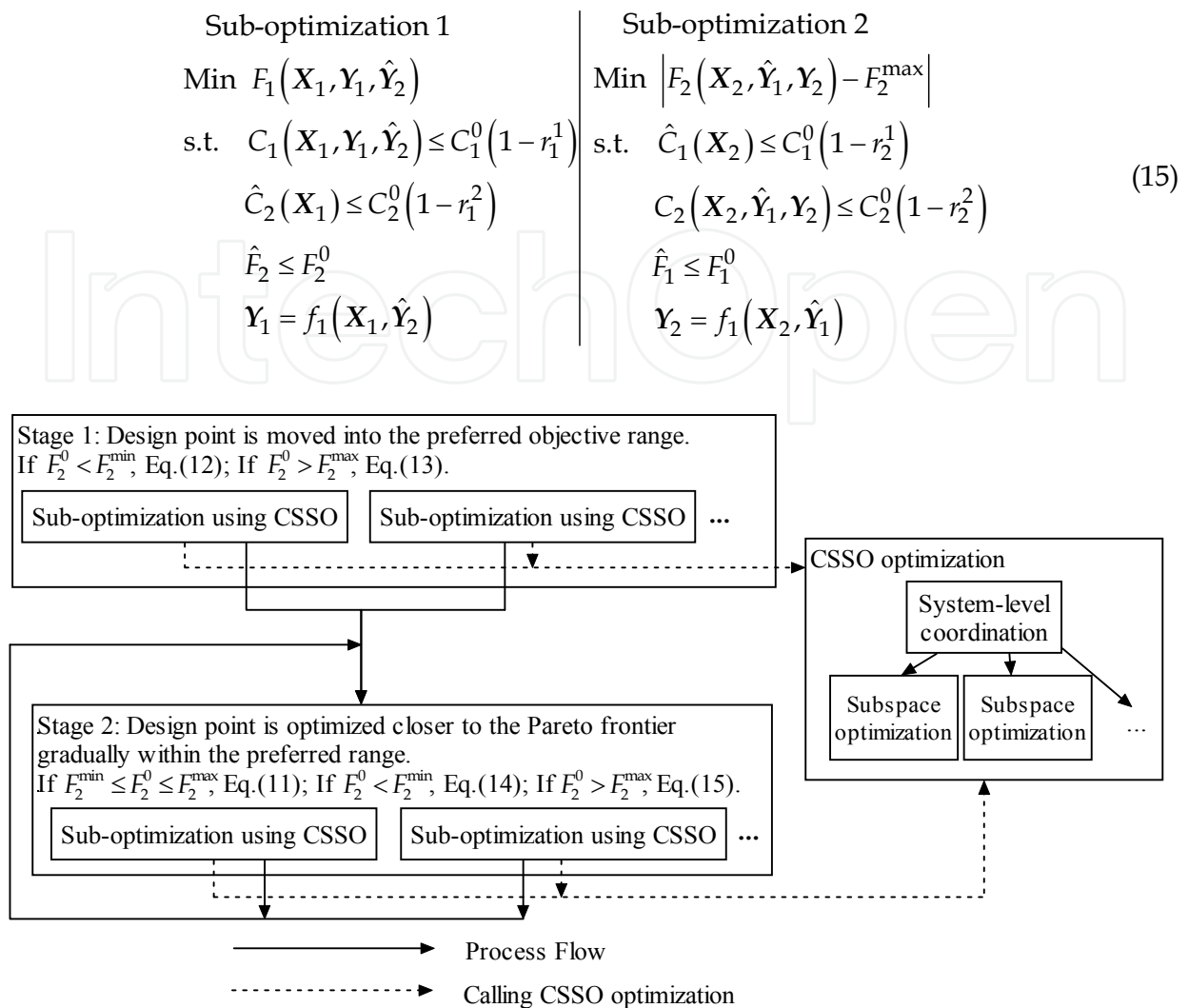


Fig. 6. Framework of MORCSSO method

For the MORCSSO method, in the case of $F_2^0 > F_2^{\max}$ in the second-stage optimizations, the optimizations may converge to the Pareto frontier above the preferred objective range, i.e. the Pareto point that $F_2 > F_2^{\max}$ is obtained.

Why does the optimization fail in this case? It can be analyzed from Eq. (15) in the objective space shown in Fig. 7. For Eq. (15), (1) X_0 should be closer to the line $F_2 = F_2^{\max}$ according to $\text{Min } |F_2 - F_2^{\max}|$, (2) $F_2(X_0)$ should be no more than F_2^0 according to $\hat{F}_2 \leq F_2^0$, (3) X_0 should be optimized to decrease F_1 according to $\hat{F}_1 \leq F_1^0$ and $\text{Min } F_1$, (4) X_0 should be in the feasible region. As shown in Fig. 7, (2) and (3) forces X_0 to move along a direction of \bar{n}_1 in the lower-left shadowed region, in which direction the design point will impossibly move into the preferred objective range. Only along a direction of \bar{n}_2 in the lower-right shadowed region the preferred range can be achieved. In such a case X_0 may move down straightly along the direction of \bar{n} to arrive at the Pareto front. The semi-infinite region between $F_1 \leq F_{\max 1}$ and the Pareto front is named as Blind Region in this chapter, which means the point falling into this region will not converge to the Pareto front in the preferred region any more. This error will happen in the case of three and more objectives as well, as shown in Fig. 8.

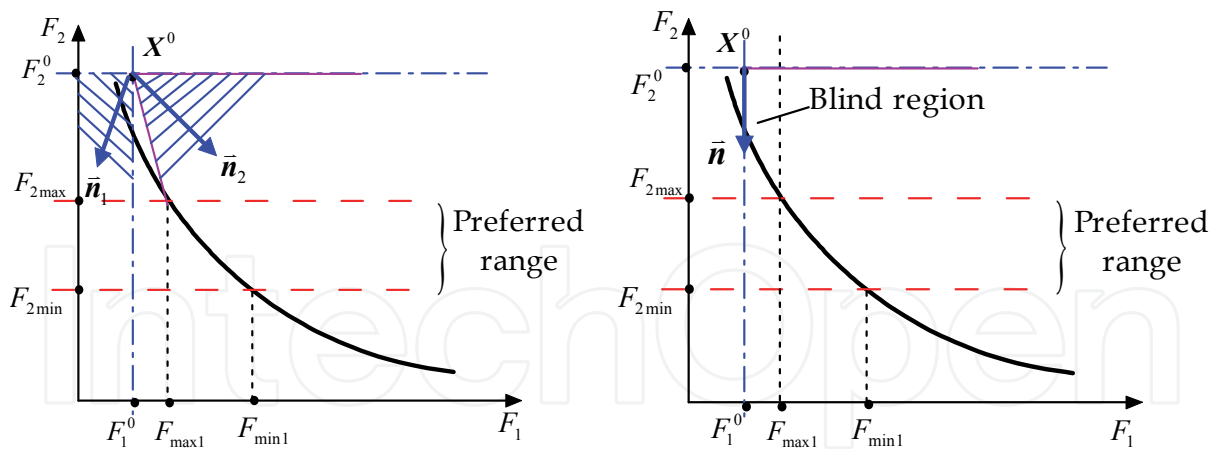


Fig. 7. The analysis of Eq. (15) in bi-objective space

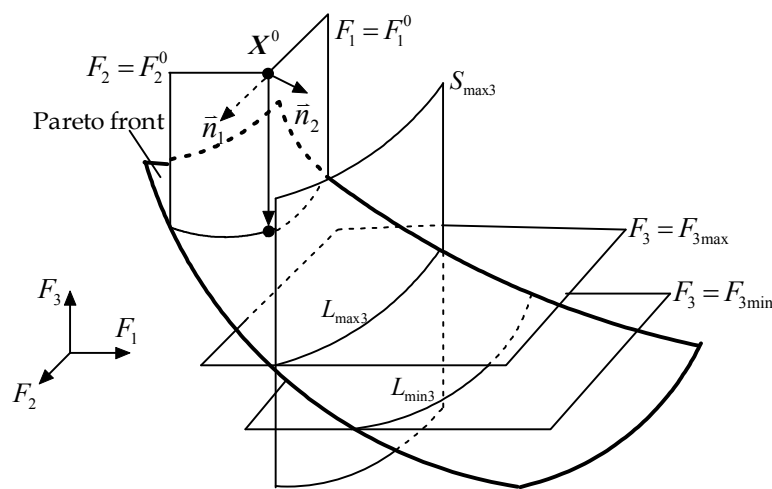


Fig. 8. The analysis of Eq. (15) in three-objective space

How to solve this problem? From Fig. 7, we only need to move the line $F_1 = F_1^0$ right a little bit, then the two shadowed region will be crossed each other. In the mathematical meaning $F_1 \leq F_1^0$ is relaxed to $F_1 \leq F_1^0 + \omega |F_1^0|$ in Eq. (15). This strategy is proven to be effective.

4.3 Adaptive Weighted Sum based CSSO (AWSCSSO)

The procedure of solving the Pareto front by AWSCSSO is similar to Adaptive Weighted Sum (AWS) method (Kim & de Weck, 2004, 2005). As an example, the AWSCSSO method for a generic bi-objective problem, with subsystem 1 and 2, is stated in the following paragraphs (Eq. (1): two objectives and two coupled subsystems). In the same way AWSCSSO can be also applied for the multi-objective problem with three or more subsystems.

1. In the first stage a rough profile of the Pareto front is determined. The variant of CSSO method described in Sub-section 3.2 is adopted in the single objective optimization for each objective function and objective function is normalized as following:

$$\bar{J}_i = \frac{J_i - J^{\text{Nadir}}}{J^{\text{Utopia}} - J^{\text{Nadir}}} \tag{16}$$

When X_i^* is the optimal solution vector for the single objective optimization of the i th objective function J_i , the utopia point and pseudo nadir point are defined as

$$J^{\text{Utopia}} = [J_1(X^{1*}), J_2(X^{2*}), \dots, J_m(X^{m*})] \quad (17)$$

$$J^{\text{Nadir}} = [J_1^{\text{Nadir}}, J_2^{\text{Nadir}}, \dots, J_m^{\text{Nadir}}] \quad (18)$$

Where $J_i^{\text{Nadir}} = \max [J_i(X^{1*}), J_i(X^{2*}), \dots, J_i(X^{m*})]$ and m is the number of objective functions.

Then with a large step size of the weighting factor the usual weighted sum method is used in the variant of GSECSSO to approximate the Pareto front quickly. The subspace optimization for AWSCSSO can be expressed as

| | |
|--|---|
| <p style="text-align: center;">Sub-optimization 1</p> <p>Min $W_1 F_1(X_1, Y_1, \hat{Y}_2) + W_2 \hat{F}_2(X_1)$</p> <p>s.t. $C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1)$</p> <p style="padding-left: 40px;">$\hat{C}_2(X_1) \leq C_2^0(1 - r_1^2)$</p> <p style="padding-left: 40px;">$Y_1 = f_1(X_1, \hat{Y}_2)$</p> | <p style="text-align: center;">Sub-optimization 2</p> <p>Min $W_1 \hat{F}_1(X_2) + W_2 F_2(X_2, \hat{Y}_1, Y_2)$</p> <p>s.t. $\hat{C}_1(X_2) \leq C_1^0(1 - r_2^1)$</p> <p style="padding-left: 40px;">$C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2)$</p> <p style="padding-left: 40px;">$Y_2 = f_2(X_2, \hat{Y}_1, \hat{Y}_3)$</p> |
|--|---|

(19)

Where the value with symbol '^' above is a linearly approximated one, C_1 and C_2 are cumulative constraints of G_1 and G_2 , respectively, and r_k^p represents responsibility assigned to the k -th subsystem for reducing the violation of C_p . The value with superscript '0' is corresponding to the starting point X_0 . W_1 and W_2 are weighting factors for objective function vector F_1 and F_2 , respectively.

By estimating the size of each Pareto patch, the refined regions in the objective space are determined. An example of the mesh refinement in AWSCSSO is shown in Fig. 9. Where hollow points represent the newly refined node P_E (expected solution) while solid points represent initial four nodes that define the patch. As shown in Fig. 9, the quadrilateral patch is taken as an example. If the line segment that connects two neighboring nodes of the patch is too long, it is divided into only two equal line segments. The central point becomes the

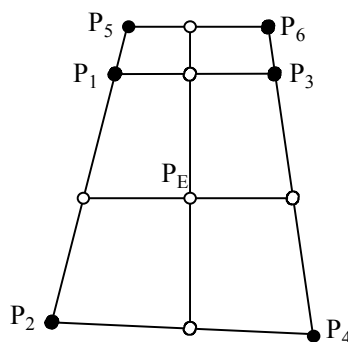


Fig. 9. Refine patches of AWSCSSO method

new refined node. These refined nodes are connected to form a refined mesh. Then the sub-optimizations in Eq. (19) are performed using different additional constraints for different refined nodes and the new Pareto points are obtained. In next step, according to the prescribed density of Pareto points, the Pareto-front patch that is too large will be refined again in the same way. In subsequent steps, the refinement and sub-optimizations are repeated until the number of Pareto points does not increase anymore.

2. In the subsequent stage only these regions are specified as feasible domains for sub-optimization problem with additional constraints. Each Pareto front patch is then refined by imposing additional equality constraints that connect the pseudo nadir point (P_N) and the expected Pareto optimal solutions (P_E) on a piecewise planar surface in the objective space (as shown in Fig. 10).

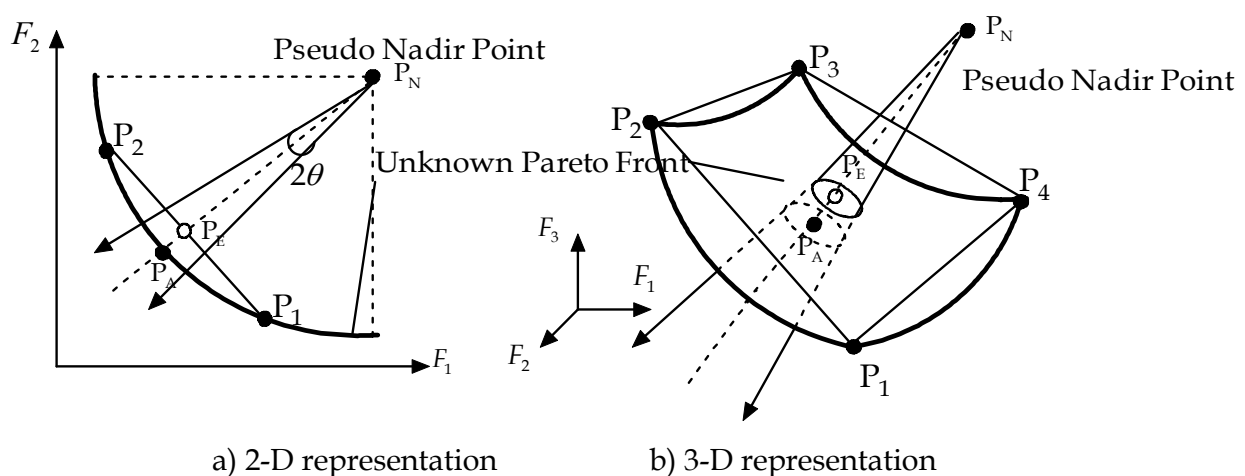


Fig. 10. AWSCSSO method for multidimensional problems

Sub-optimizations are defined by imposing additional constraint H to Eq. (19) as

| Sub-optimization 1 | Sub-optimization 2 |
|---|---|
| Min $W_1 F_1(X_1, Y_1, \hat{Y}_2) + W_2 \hat{F}_2(X_1)$ | Min $W_1 \hat{F}_1(X_2) + W_2 F_2(X_2, \hat{Y}_1, Y_2)$ |
| s.t. $C_1(X_1, Y_1, \hat{Y}_2) \leq C_1^0(1 - r_1^1)$ | s.t. $\hat{C}_1(X_2) \leq C_1^0(1 - r_2^1)$ |
| $\hat{C}_2(X_1) \leq C_2^0(1 - r_1^2)$ | $C_2(X_2, \hat{Y}_1, Y_2) \leq C_2^0(1 - r_2^2)$ |
| $Y_1 = f_1(X_1, \hat{Y}_2)$ | $Y_2 = f_2(X_2, \hat{Y}_1, \hat{Y}_3)$ |
| $H(X_1, Y_1, \hat{Y}_2) \leq 0$ | $H(X_2, \hat{Y}_1, \hat{Y}_3) \leq 0$ |

The additional inequality constraint is

$$H = -\frac{(\bar{F}_E - \bar{F}_N) \cdot (\bar{F}(X) - \bar{F}_N)}{|\bar{F}_E - \bar{F}_N| |\bar{F}(X) - \bar{F}_N|} + L \leq 0 \quad (21)$$

Where L is the adaptive relax factor that is less than 1. \bar{F}_E , \bar{F}_N and $\bar{F}(X)$ are the normalized position vector of node P_E , P_N and the current design point X respectively. In

AWSCSSO, L is set to be increased with the rise of the distribution density of Pareto points.

Fig. 11 shows the framework of AWSCSSO. In Fig. 11, W^{1i} and W^{2i} are weighting factors in stage 1 and stage 2, respectively; H is the additional constraint. The optimization problem is performed in two stages in the AWSCSSO method. In the first stage the Pareto front is approximated quickly with large step size of weight factors. The optimization problems of this stage are defined in Eq. (19). In the subsequent stage, by calculating the distances between neighboring solutions on the front in objective space, the refined regions are identified and the refined mesh is formulated. Only these regions then become the feasible regions for optimization by imposing additional constraints in the objective space. The optimization problems of this stage are defined in Eq. (20). The different locations of new Pareto points are defined by the different additional constraints. Optimization is performed in each of the regions and the new solution set is acquired. Being a MDO problem, the optimization is performed by the variant of GSECSSO method.

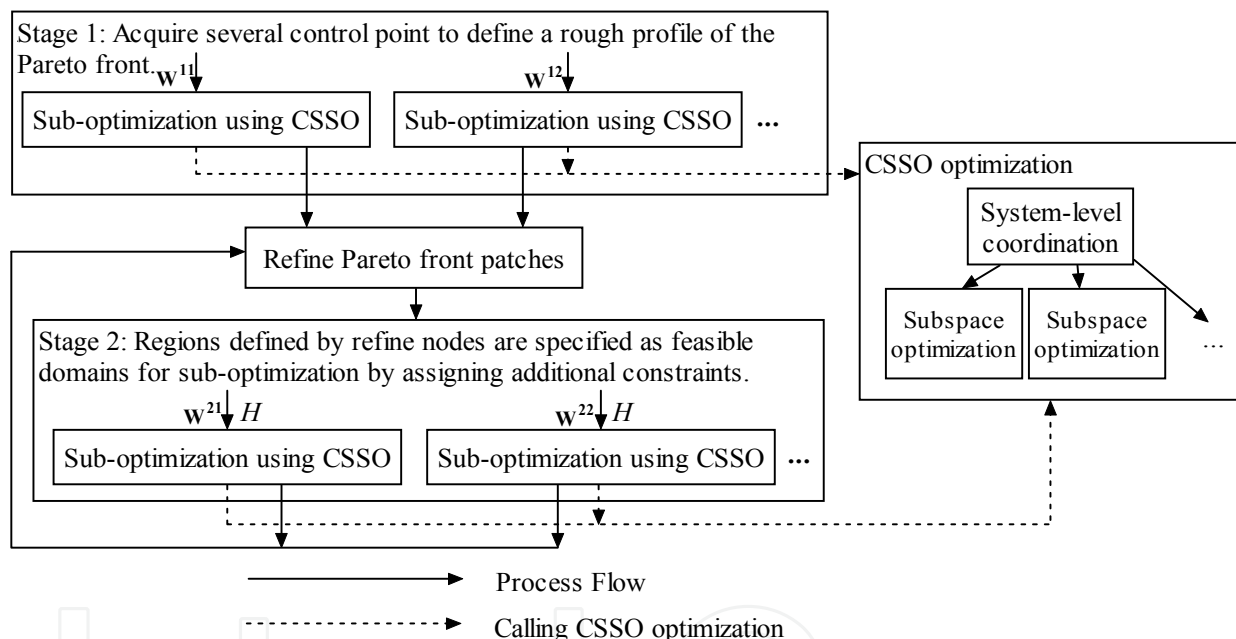


Fig. 11. Framework of AWSCSSO method

4.4 Examples

4.4.1 Example 1: Convex Pareto front

This problem is taken from a test problem (Huang, 2003). This is a test problem available in the NASA Langley Research Center MDO Test Suite. It has two objectives, F_1 and F_2 , to be minimized. It consists of ten inequality constraints, four coupled state variables and ten design variables in two coupled subsystems. The mathematical model is not listed here for concision. We refer the readers to the test problem 1 in the corresponding references.

The comparison of the solution obtained by MOPCSSO and AWSCSSO is shown in Fig. 12. It can be concluded that for the problem with convex Pareto front a uniformly-spaced wide-distributed and smooth Pareto front can be obtained by AWSCSS method. When using MOPCSSO I have not captured the whole range.

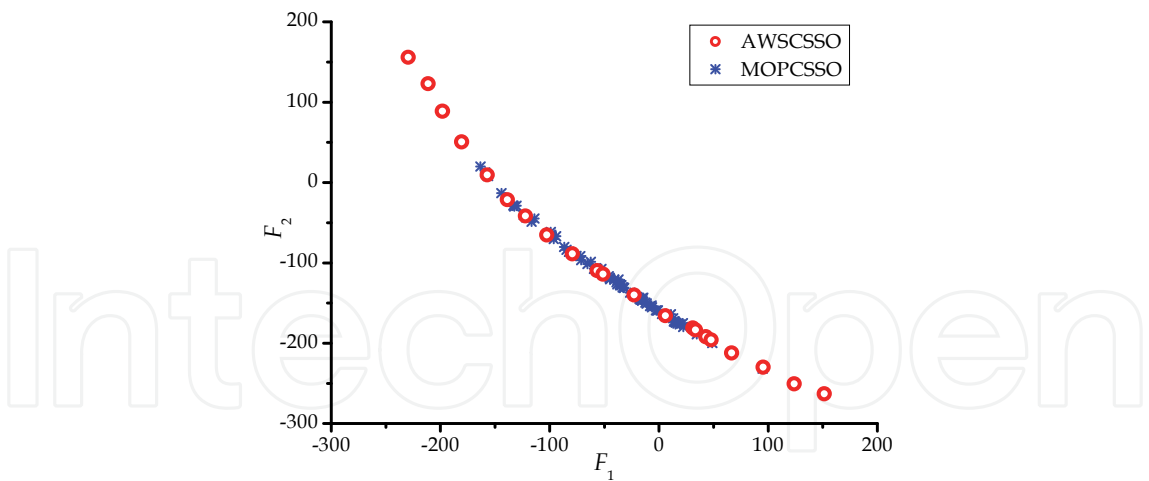


Fig. 12. Comparison of Pareto front obtained by using AWSCSSO and MOPCSSO

4.4.2 Example 2: Non-convex Pareto front

This problem consists of two objective functions, six design variables and six constraints. Two objectives, F_1 and F_2 need to be minimized. The model problem is defined as

Min

$F_1(x) = -\left(25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2\right)$

Min

$F_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$

s.t.

$c_1(x) = x_1 + x_2 - 2 \geq 0$

$c_2(x) = 6 - x_1 - x_2 \geq 0$

$c_3(x) = 2 + x_1 - x_2 \geq 0$

$c_4(x) = 2 - x_1 + 3x_2 \geq 0$

$c_5(x) = 4 - (x_3 - 3)^2 - x_4 \geq 0$

$c_6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0$

$0 \leq x_1, x_2, x_6 \leq 10, \quad 1 \leq x_3, x_5 \leq 5, \quad 0 \leq x_4 \leq 6$

(22)

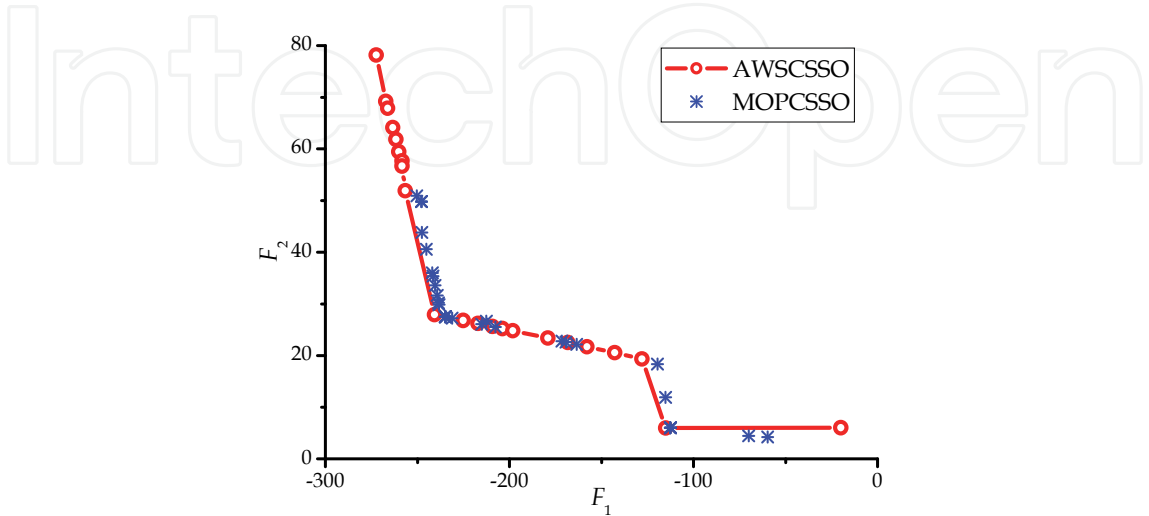


Fig. 13. Comparison of Pareto fronts obtained by using AWSCSSO and MOPCSSO methods

The comparison of Pareto front obtained by AWSCSSO and MOPCSSO is shown in Fig. 13. It is concluded that, for the problem with non-convex Pareto front, the more uniformly-spaced, more widely-distributed and smoother Pareto front is also obtained by the AWSCSSO method.

4.4.3 Example 3: Conceptual design of a subsonic passenger aircraft

The mathematical model of this problem is defined as

IntechOpen

$$\begin{aligned} &\text{Max } U \\ &\text{Max } L/D_C \\ &\text{s.t. } C_{d0L} \leq 0.2, C_{d0C} \leq 0.02 \\ &\quad R_f \geq 1 \\ &\quad q_{To} \geq 0.027, q_L \geq 0.024 \\ &\quad D_{To} \leq 1981, D_L \leq 1371 \end{aligned} \tag{23}$$

The objective functions in Eq. (23) are to maximize useful load fraction (U) and lift-to-drag ratio for the cruising condition (L/D_C). The constraints in Eq.(23) are as follows. (1) The drag coefficient for the take-off condition and landing condition (C_{d0L}) is no more than 0.2 and that for the cruising condition (C_{d0C}) is no more than 0.02. (2) The overall fuel balance coefficient (R_f) is no less than 1. (3) The achievable climb gradient for the take-off condition (q_{To}) is greater than 0.027 and that for the landing condition (q_L) is greater than 0.024. (4) The take-off field length (D_{To}) is less than 1981m and the landing field length (D_L) is less than 1371m. The overall fuel balance coefficient is defined as the ratio of the fuel weight required for mission to that available for mission. The design variables are listed in Table 3.

| Design Variable/unit | Symbol | Lower limit | Upper limit |
|--|----------|-------------|-------------|
| Wing area/m ² | S | 111.48 | 232.26 |
| Aspect ratio | AR | 9.5 | 10.5 |
| Design gross weight/10 ³ kg | W_{dg} | 63.504 | 113.400 |
| Installed thrust/10 ³ kg | T_i | 12.587 | 24.948 |

Table 3. List of design variables

Two disciplines, aerodynamics and weight, are considered in this problem. The dataflow between and in subsystems is analyzed in Fig. 14, where L/D_{To} , L/D_L , L/D_C are the lift-to-drag ratios for the take-off, landing and cruising conditions respectively, V_{br} is the cruise velocity with the longest range, R_{fr} is the fuel weight fraction required for mission, and C_{d0C} is the zero-lift drag coefficient for the cruising condition.

Two disciplines, aerodynamics and weight, are coupled. When the state variables in aerodynamics such as cruise velocity with the longest range, lift coefficients, zero-lift drag coefficients, skin-friction drag coefficients, lift-to-drag ratio are computed, some state variables in Weight such as R_{fr} should be known. Similarly, when the state variables in Weight such as useful load fraction, overall fuel balance coefficient, achievable climb gradient on take-off and landing, take-off field length and landing field length are computed, some state variables in Aerodynamics such as L/D_{To} , L/D_L , L/D_C and V_{br} should also be provided. In the Aerodynamics discipline, V_{br} is coupled with C_{d0C} . The dataflow between state variables and design variables can be seen in Fig. 15. Many details of equations in the aerodynamic discipline model and weight discipline model can refer to the

reference (Zhang et al., 2008). The full description of them can be found in the references (Lewis & Mistree, 1995; Lewis, 1997).

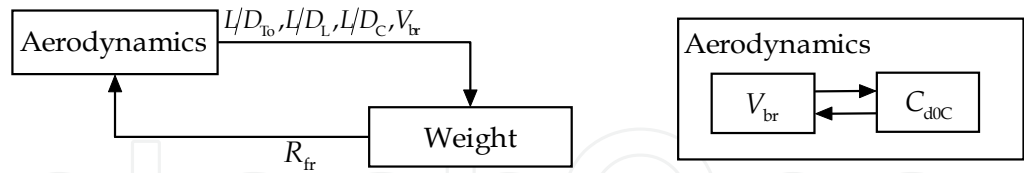


Fig. 14. Dataflow between and in subspaces

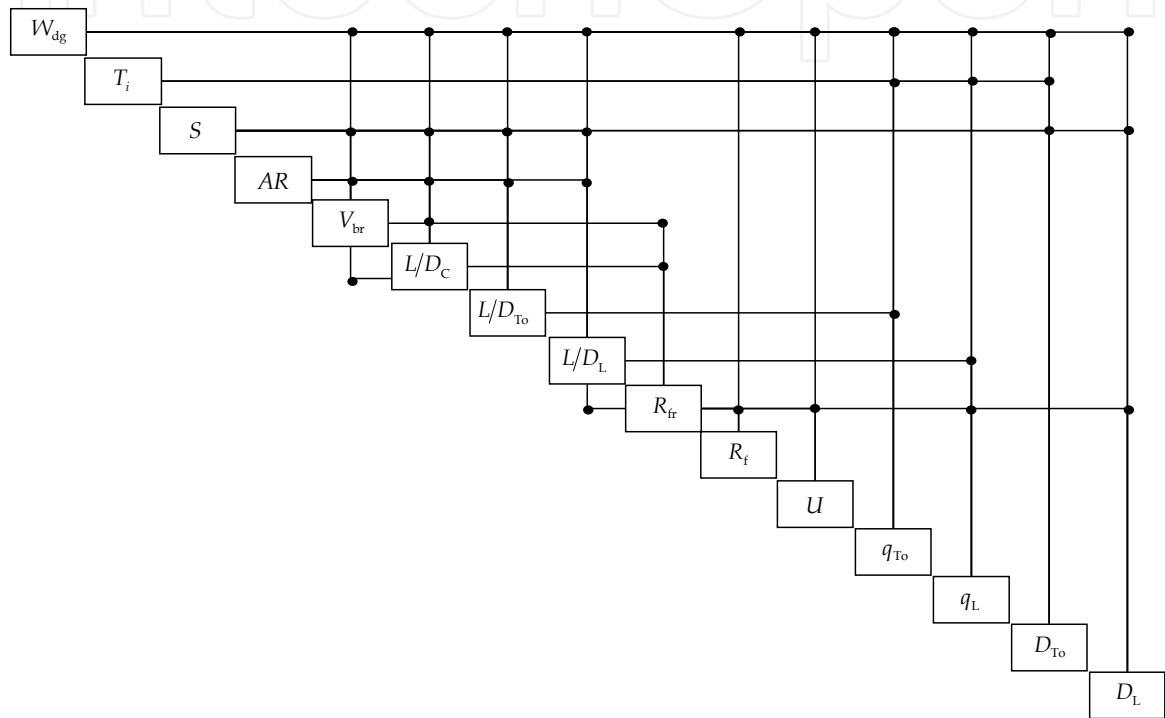


Fig. 15. Dataflow between state variables and design variables

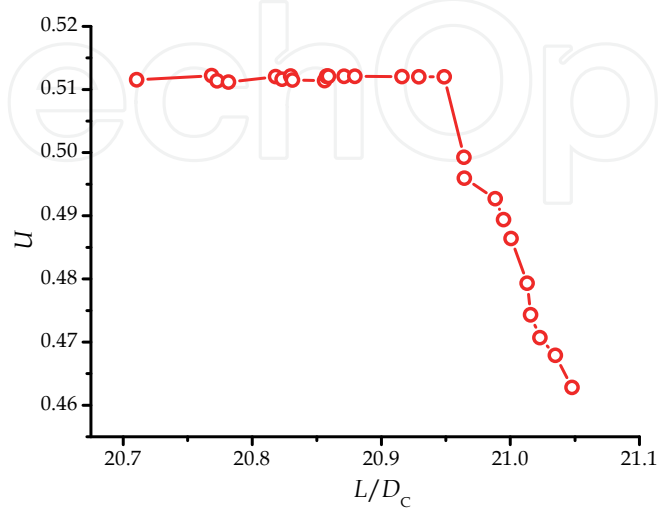


Fig. 16. Pareto front obtained by using AWSCSSO

The Pareto front obtained using AWSCSSO is shown in Fig. 16. Each solution on Pareto front is obtained using CSSO with iterative subspace optimizations. Taking one of the optimal designs as example, the values of the design variables are: $S=232.3\text{m}^2$, $AR=10.5$, $W_{dg}=113.4\times 10^3\text{kg}$, $T_i=16.75\times 10^3\text{kg}$. The performance parameters of the aircraft in optimal design are as follows: $C_{d0C}=0.01777$, $L/D_C=21.05$, $V_{br}=183.43\text{m/s}$, $q_{To}=0.03303$, $q_L=0.08804$, $D_{To}=1823\text{m}$ and $D_L=1086\text{m}$. Several conclusions can be made from these results. 1) The AWSCSSO method is primarily proved to be applicable for aircraft conceptual design. 2) The distribution of Pareto points is not so uniform as expected. These results are still very encouraging in general. The non-uniformity may be due to the additional constraint that changes the location to expected solution. Further study is still needed on how to achieve the balance between uniformity and convergence.

5. Conclusion

The CSSO method is one of the main bi-level MDO methods. Couples of CSSO methods for single- and multi-objective MDO problems are discussed in this chapter. It can be concluded that, (1) number of the system analysis can be greatly reduced by using the CSSO methods, which in turn improve the efficiency; (2) the CSSO methods enable concurrent design and optimization of different design groups, which can greatly improve efficiency; (3) the CSSO methods are effective and applicable in solving not only single-objective but also multi-objective MDO problems.

For the CSSO methods, although the RSCSSO method behaves more robust, it will actually reduce to a single-level surrogate modeling based MDO method since the subspace optimizations have little impact on the results. So the GSECSSO method is more promising as a bi-level method and worth further studying. The future study on the GSECSSO method will focus on improving its robustness and efficiency. For the multi-objective CSSO methods, the AWSCSSO method behaves better on obtaining widely-distributed Pareto points. The future work on the multi-objective CSSO methods will focus on improving the solution quality and also on testing it for more realistic engineering design problems.

6. References

- Aute, V. and Azarm, S. A, "Genetic Algorithms Based Approach for Multidisciplinary Multiobjective Collaborative Optimization," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Virginia, 2006, AIAA 2006-6953.
- Azarm, S. and Li, W.C., "Multi-level Design Optimization using Global Monotonicity Analysis," *ASME Journal of Mechanisms and Automation in Design*, 1989, Vol.111, No.2, pp.259-263.
- Bolebaum, C.L., "Formal and Heuristic System Decomposition in Structural Optimization," NASA-CR-4413, 1991.
- Huang, C. H., "Development of Multi-Objective Concurrent Subspace Optimization and Visualization Methods for Multidisciplinary Design," Ph.D. Dissertation, The State University of New York, New York, 2003.
- Huang, C. H. and Bloebaum, C. L., "Incorporation of Preferences in Multi-Objective Concurrent Subspace Optimization for Multidisciplinary Design," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, New York, 2004, AIAA 2004-4548.
- Huang, C. H. and Bloebaum, C. L., "Multi-Objective Pareto Concurrent Subspace Optimization for Multidisciplinary Design," *AIAA Journal*, Vol. 45, No. 8, 2007, pp. 1894-1906.

- Kreisselmeier, G., Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *IEAC Symposium on Computer Aided Design of Control Systems*, Zurich, Switzerland, 1979.
- Kim, I. Y. and de Weck, O. L., "Adaptive Weighted Sum Method for Multiobjective Optimization," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, New York, 2004, AIAA 2004-4322.
- Kim, I. Y. and de Weck, O. L., "Adaptive Weighted-sum Method for Bi-objective Optimization: Pareto Front Generation," *Structural and Multidisciplinary Optimization*, No. 29, 2005, pp. 149-158.
- Lewis, K., Mistree, F., "Designing Top-level Aircraft Specifications-A Decision-based Approach to A Multiobjective, Highly Constrained Problem," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA, AIAA 1995-1431.
- Lewis, K., "An Algorithm for Integrated Subsystem Embodiment and System Synthesis," Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia, August 1997.
- McAllister, C. D., Simpson, T. W. and Yukesh, M., "Goal Programming Applications in Multidisciplinary Design Optimization," *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, CA, 2000, AIAA 2000-4717.
- McAllister, C. D., Simpson, T. W., Lewis, K. and Messac, A., "Robust Multiobjective Optimization through Collaborative Optimization and Linear Physical Programming," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, New York, 2004, AIAA 2004-4549.
- Orr, S. A. and Hajela, P., "Genetic Algorithm Based Collaborative Optimization of A Tiltrotor Configuration," *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Texas, 2005, AIAA 2005-2285.
- Parashar, S. and Bloebaum, C. L., "Multi-objective Genetic Algorithm Concurrent Subspace Optimization (MOGACSSO) for Multidisciplinary Design," *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Rhode Island, 2006, AIAA 2006-2047.
- Renaud, J.E. and Gabriele, G.A., "Second Order Based Multidisciplinary Design Optimization Algorithm Development," *Advance in Design Automation*, 1993, Vol.65, No.2, pp.347-357.
- Renaud, J.E. and Gabriele, G.A., "Improved coordination in non-hierarchic system optimization," *AIAA Journal*, 1993, Vol.31, No.12, pp.2367-2373.
- Renaud, J.E. and Gabriele, G.A., "Approximation in non-hierarchic system optimization," *AIAA Journal*, 1994, Vol.32, No.1, pp.198-205.
- Sellar, R. S, Batill, S. M and Renaud, J. E., "Response surface based, concurrent subspace optimization for multidisciplinary system design," *34th AIAA Aerospace Sciences Meeting*, 1996, AIAA 96-0714.
- Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-hierarchic Systems," *Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, Hampton, 1988.
- Sobieszczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 153-160.
- Tappeta, R. V. and Renaud, J. E., "Multiobjective Collaborative Optimization," *ASME Journal of Mechanical Design*, No. 3. 1997, pp. 403-411.
- Zhang, K.S., Han, Z.H., Li, W.J., and Song, W.P., "Bilevel Adaptive Weighted Sum Method for Multidisciplinary Multi-Objective Optimization," *AIAA Journal*, 2008, Vol.46 No.10, pp. 2611-2622.



Aeronautics and Astronautics

Edited by Prof. Max Mulder

ISBN 978-953-307-473-3

Hard cover, 610 pages

Publisher InTech

Published online 12, September, 2011

Published in print edition September, 2011

In its first centennial, aerospace has matured from a pioneering activity to an indispensable enabler of our daily life activities. In the next twenty to thirty years, aerospace will face a tremendous challenge - the development of flying objects that do not depend on fossil fuels. The twenty-three chapters in this book capture some of the new technologies and methods that are currently being developed to enable sustainable air transport and space flight. It clearly illustrates the multi-disciplinary character of aerospace engineering, and the fact that the challenges of air transportation and space missions continue to call for the most innovative solutions and daring concepts.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ke-shi Zhang (2011). Concurrent Subspace Optimization for Aircraft System Design, Aeronautics and Astronautics, Prof. Max Mulder (Ed.), ISBN: 978-953-307-473-3, InTech, Available from:
<http://www.intechopen.com/books/aeronautics-and-astronautics/concurrent-subspace-optimization-for-aircraft-system-design>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen