

## Research Article

# A Universal MDO Framework Based on the Adaptive Discipline Surrogate Model

Hua Su , Chun-lin Gong , and Liang-xian Gu

*Shaanxi Aerospace Flight Vehicle Design Key Laboratory, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

Correspondence should be addressed to Chun-lin Gong; [leonwood@nwpu.edu.cn](mailto:leonwood@nwpu.edu.cn)

Received 15 July 2018; Accepted 19 September 2018; Published 27 November 2018

Academic Editor: Hikmat Asadov

Copyright © 2018 Hua Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

High time-consuming computation has become an obvious characteristic of the modern multidisciplinary design optimization (MDO) solving procedure. To reduce the computing cost and improve solving environment of the traditional MDO solution method, this article introduces a novel universal MDO framework based on the support of adaptive discipline surrogate model with asymptotical correction by discriminative sampling. The MDO solving procedure is decomposed into three parts: framework level, architecture level, and discipline level. Framework level controls the MDO solving procedure and carries out convergence estimation; architecture level executes the MDO solution method with discipline surrogate models; discipline level analyzes discipline models to establish adaptive discipline surrogate models based on a stochastic asymptotical sampling method. The MDO solving procedure is executed as an iterative way included with discipline surrogate model correcting, MDO solving, and discipline analyzing. These are accomplished by the iteration process control at the framework level, the MDO decomposition at the architecture level, and the discipline surrogate model update at the discipline level. The framework executes these three parts separately in a hierarchical and modularized way. The discipline models and disciplinary design point sampling process are all independent; parallel computing could be used to increase computing efficiency in parallel environment. Several MDO benchmarks are tested in this MDO framework. Results show that the number of discipline evaluations in the framework is half or less of the original MDO solution method and is very useful and suitable for the complex high-fidelity MDO problem.

## 1. Introduction

MDO solution method, also names MDO architecture or MDO strategy, is the most essential part of the MDO solving procedure [1]. It is a process to organize disciplines and format MDO problem into a mathematical model which could be solved. Such works include decomposing the MDO problem, interconnecting coupling, coordinating the system, and organizing implementation procedure of the MDO problem. With constant increases in product integration and complexity, the application of the MDO solution method is a significant challenge in complex industrial design [2]. Especially for aerospace engineering, modern aircraft design includes aerodynamic, structure, aerothermodynamics, control, trajectory, etc., which cause an extremely complex and time-consuming problem [3]. Traditional design methods with statistical data and empirical formula

are no longer applicable. So many high-precision numerical analysis methods such as computational fluid mechanics (CFD), computational structural mechanics (CSM), and computational electromagnetic (CEM) arise constantly. Time-consuming computation, huge information interchange, nonsmooth and nonlinear design space become new challenges to the traditional MDO solution method. Meanwhile, technologies such as parallel computing, distributed computing, and approximation method are increasingly important in the field of optimization. How to utilize these useful technologies to obtain a more efficient and robust MDO solution environment is a fascinating research area.

Wang et al. [4] proposed a new global optimization method for expensive black-box function. This global optimization method based on a novel mode-pursuing sampling (MPS) algorithm systematically generates more sample points

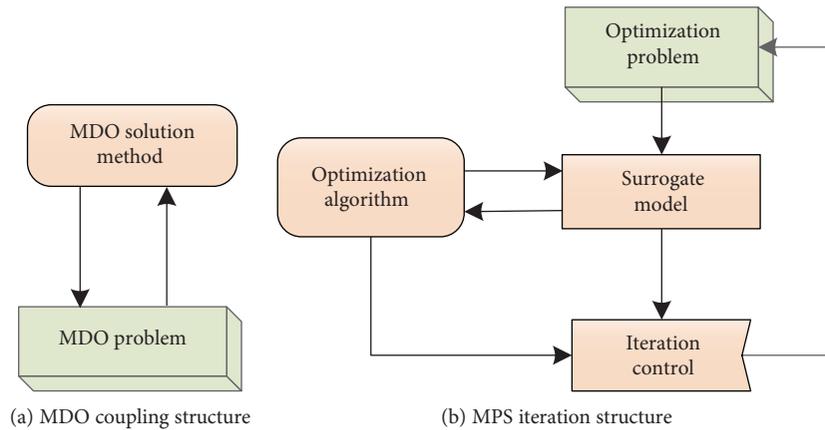


FIGURE 1: The comparison of MDO coupling structure and MPS iteration structure.

in the neighborhood of the function mode while statistically covering the entire search space. MPS has good convergence and does not depend on the derivative information, which is significantly suitable for computation-intensive optimization problems [4–7]. MPS is also applied to discrete variable optimization [8], multiobjective optimization [9], and hybrid optimization [10]. Wang et al. [11–15] proposed a new collaboration pursuing method (CPM) based on the MPS algorithm with the multidisciplinary analysis (MDA) process and aimed to search for the global optimum solution of a MDO problem with a short turnaround time. It maintains the consistency of coupling among state parameters by a collaboration model implemented with radial basis functions, and MPS is applied as a global optimizer module for CPM to search for the global optimum solution of the MDO problem. But the limitation of the MDA process hinders the use of the other MDO solution method and cannot satisfy the need for the flexible solution of the MDO solving procedure.

In this article, a novel universal MDO framework based on the idea of the stochastic asymptotical sampling method is presented. At first, the idea and composition of the universal MDO framework are introduced and the elementary theory of the MPS algorithm is simply reviewed. Then, based on the MPS iteration process, the solving procedure of the universal MDO framework is proposed. The detailed implementation of all three levels is discussed and used to construct the universal MDO framework. Finally, some benchmarks are utilized to give a verification and validation of the effectiveness and high efficiency.

## 2. A Universal MDO Framework

MPS algorithm has the ability of fast convergence which can be helpful for the time-consuming MDO problem. MDO problem is a combination of various disciplines with interdisciplinary coupling, and MDO solution method decouples this interdisciplinary relationship directly or indirectly to construct the optimization problem. Figure 1 gives a comparison between the MDO coupling structure and the MPS iteration structure. As shown in Figure 1(a), MDO

solving procedure is totally controlled by the MDO solution method, which includes execution sequence and iteration pattern of the discipline model. The traditional MDO solving procedure is opaque. Actually, because of the design complexity of the MDO algorithm, calling mechanism of the discipline model is very complex, and it is difficult to manage the discipline models centrally. The MPS iteration process is shown in Figure 1(b). It is an iterative structure. The optimization algorithm carries on surrogate models of the optimization problem. The iteration control is utilized to manage the surrogate model updating and the convergence estimation. The degree of freedom of the optimization process can be improved greatly due to the decoupling of the optimization problem and the optimization algorithm, and this structure also provides a good support to parallel computing.

The MPS iteration structure gives a fast convergence to the optimization problem; the results in [7] show that it has far fewer function evaluations than the GA algorithm, which can be a significant help to accelerate the MDO solving procedure. But it is not a good idea to change the optimization problem in Figure 1(b) into the MDO problem in Figure 1(a) directly. Actually, it is better to approximate the discipline model rather than the MDO problem because of the opacity relation of the MDO coupling structure; then, each discipline model can be evaluated and approximated separately. An independent iteration control to update each discipline surrogate model is also needed to improve the quality of approximation.

In this article, the discipline model is regarded as the optimization problem of the MPS iteration structure shown in Figure 1(b), and the corresponding surrogate model is called a discipline surrogate model. The MDO solving procedure is carried out with these discipline surrogate models, and then, the current optimal solution is obtained and used to update the discipline surrogate model. This iteration process goes on as the MPS process until it converges.

Based on the above design pattern, the MDO framework is decomposed into three parts: framework level, architecture level, and discipline level. The design skeleton of the

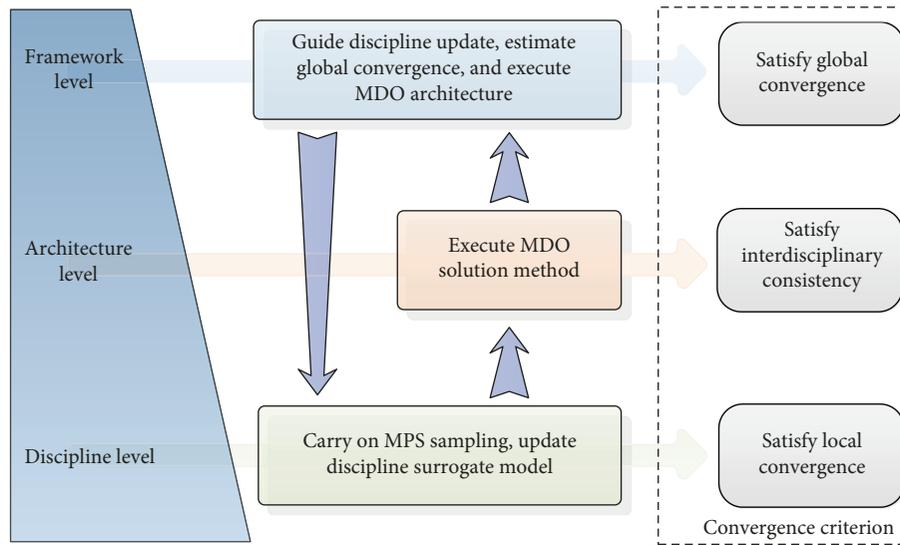


FIGURE 2: Design skeleton of the MDO framework.

universal MDO framework can be viewed as Figure 2. The roles of these three levels are as follows:

- Framework level: framework level is the system level of the MDO framework. It is responsible for managing the organizational relationship and implementation process of the framework and guiding the convergence process of the MDO problem. The main feature of the framework level includes global convergence detecting and MDO solution method constructing
- Architecture level: architecture level decouples the MDO problem into some optimization problems and solves it based on the discipline surrogate model under the control of the framework level. This process will repeat continuously with the updating of the adaptive discipline surrogate model. The computing efficiency of a single MDO solving procedure will gain great improvement by replacing the discipline model to a discipline surrogate model
- Discipline level: discipline level executes discipline analysis at sample points based on the MPS sampling process and the current optimal solution and updates the adaptive discipline surrogate model. The two-way interaction between the discipline model at the discipline level and the MDO solution method at the architecture level is simplified to one-way relation by introducing the framework level, which has a great flexibility to carry out as a distributed and parallel system

Based on the above definition, the whole MDO solving procedure is converted to a process with discipline surrogate model updating, MDO solution method solving, and discipline model analyzing. The MPS sampling method is used to establish and update the adaptive discipline surrogate model. Combining with these three levels, a novel MDO framework is

established, called it disciplinary model pursuing sampling-MDO framework (DMPS-MDOF). The detailed introduction will be given in Section 4. In the next section, we have a review of the MPS algorithm.

### 3. Review of the Mode-Pursuing Sampling Algorithm

MPS algorithm was proposed as a method to search for the global optimum of the black-box function problem by Wang et al. [4]. It is a discriminative sampling method that systematically generates more sample points in the neighborhood of the current minimum while statistically covering the entire search space. Quadratic regression is carried out to detect the region containing the global optimum and a sampling, and detection process iterates until the global optimum is obtained. MPS algorithm does not need for the gradient information and contains the huge potential for parallel computing.

MPS algorithm contains three main parts: random selection, global surrogate model, and global convergence criterion. First, some design points are generated around the current minimum while statistically covering the entire search space, and then, a surrogate model is built to fit the design space by the above points; at last, a quadratic response surface is evaluated for checking convergence and meanwhile gives a preference probability to the sampling process. The main procedure of MPS can be elaborated as follows.

*Step 1.* Set the number of uniform distributed basic sample points  $N$  and the size of contours  $K$  and initialize the speed control factor  $s$ , then generate  $m$  initial design points  $x^{(1)}, x^{(2)}, \dots, x^{(m)}$  by design of experiment (DOE) based on a uniform distributed sampling process. These design points are also used to build the design points set (DPS).

*Step 2.* Evaluate the unevaluated design points in DPS and fit a surrogate model with all design points in DPS, and then we have,

$$y^{(T)} = \tilde{f}(x^{(T)}), \quad (1)$$

where  $T$  is the size of DPS and  $\tilde{f}(\cdot)$  is the proper surrogate model selected to fit the design space.

*Step 3.* Uniformly create  $N$  sample points over the design space and evaluate the surrogate model  $\tilde{f}(x)$  to define the guidance function (GF); classify these sample points uniformly into  $K$  counters by the values of GF.

$$\text{GF} : p(x^{(K)}) = \max(\tilde{f}(x^{(K)})) - \tilde{f}(x^{(K)}). \quad (2)$$

*Step 4.* GF is a function between 0 and 1 and could be regarded as the probability density function (PDF) of the objective function. The cumulative distribution function (CDF) is also evaluated from the sample points and revised by speed control factor  $s$ . Then, apply the sampling strategy of Fu and Wang [16] to draw  $m$  random design points  $X_i^{\text{new}}$  and add these to DPS too.

*Step 5.* Evaluate the subregion of the convergence checking and build the local quadratic response surface. If the convergence criterion is not satisfied, update the speed control factor  $s$  and return to Step 2, else go to Step 6.

*Step 6.* Resample  $[n/2]$  new expensive design points randomly within the subregion, where  $[n/2]$  stand for the integer part of  $n/2$  and  $n$  is the dimension of  $x$ . Rebuild and reevaluate the local quadratic response surface. If the convergence criterion is satisfied, optimize the local quadratic response surface and get the optimum point as the global optimum, else add the  $[n/2]$  design points to DPS and return to Step 2.

## 4. DMPS-MDOF

For a better solution performance and execution control, a novel hierarchy design pattern with the framework level, architecture level, and discipline level is established in DMPS-MDOF to decompose the MDO solving procedure. The detailed implementation of all these three levels and the executing process of the DMPS-MDOF is introduced in this section.

### 4.1. Discipline Level Design

*4.1.1. Discipline Surrogate Model.* We define the discipline model as a black-box function with input variable  $X_i$  and output variable  $Y_i$ .

$$Y_i = \text{Disp}_i(X_i), \quad (3)$$

where subscript  $i$  is the  $i$ th discipline. The discipline surrogate model is expressed as

$$Y_i = \text{Disp}_i^*(X_i). \quad (4)$$

*4.1.2. Preference Function.* Discipline model has no objective function, so we need a preference function to guide the direction of the MPS sampling. Approximation capability of the discipline surrogate model has a direct effect on the convergence accuracy and efficiency of the MDO framework; the following factors should be considered:

- (a) Improve fitting accuracy by placing the new updated sampling design points near the current optimal region as much as possible
- (b) Preference function must be continuous and smooth. This will enhance the stability of the MPS sampling
- (c) MPS sampling process of each discipline must be independent and has no relevance with the other disciplines

Due to all the requirements above, the following quadratic function is used as the preference function of the discipline surrogate model:

$$D(X_i) = \|X_i - X_i^*\|_2^2, \quad (5)$$

where  $X_i^*$  is the current optimal solution of the MDO framework of discipline  $i$ . The guidance function of MPS sampling can be expressed as

$$p(X_i) = \max(D(X_i)) - D(X_i). \quad (6)$$

With the above definition, MPS sampling of each discipline will get closer to the optimal region as the iteration goes on in probabilistic meaning.

*4.1.3. Normalized Design Space.* Normalization standardizes the design space and unifies all design variables into the same level, which improves the computing efficiency and convergence precision. The following linear transformation is used to normalize the input variables of discipline:

$$\bar{X}_i^{(j)} = v_i^{(j)} X_i^{(j)} + w_i^{(j)}, \quad (7)$$

where  $X_i^{(j)}$  is the  $j$ th input variable of the discipline  $i$ ,  $\bar{X}_i^{(j)}$  is the normalized variable,  $v_i^{(j)}$  is the scale factor, and  $w_i^{(j)}$  is the offset value.  $v_i^{(j)}$  and  $w_i^{(j)}$  could be calculated by defining the proper boundary of  $\bar{X}_i^{(j)}$ .

*4.1.4. MPS Sampling Process.* For better understanding, some illustration must be listed. We define the sample point as the cheap point which only needs to evaluate the preference function in (5); the execution time can be ignored. And we define the design point as the expensive point needs time-consuming discipline analysis. The MPS sampling process for each discipline is as follows.

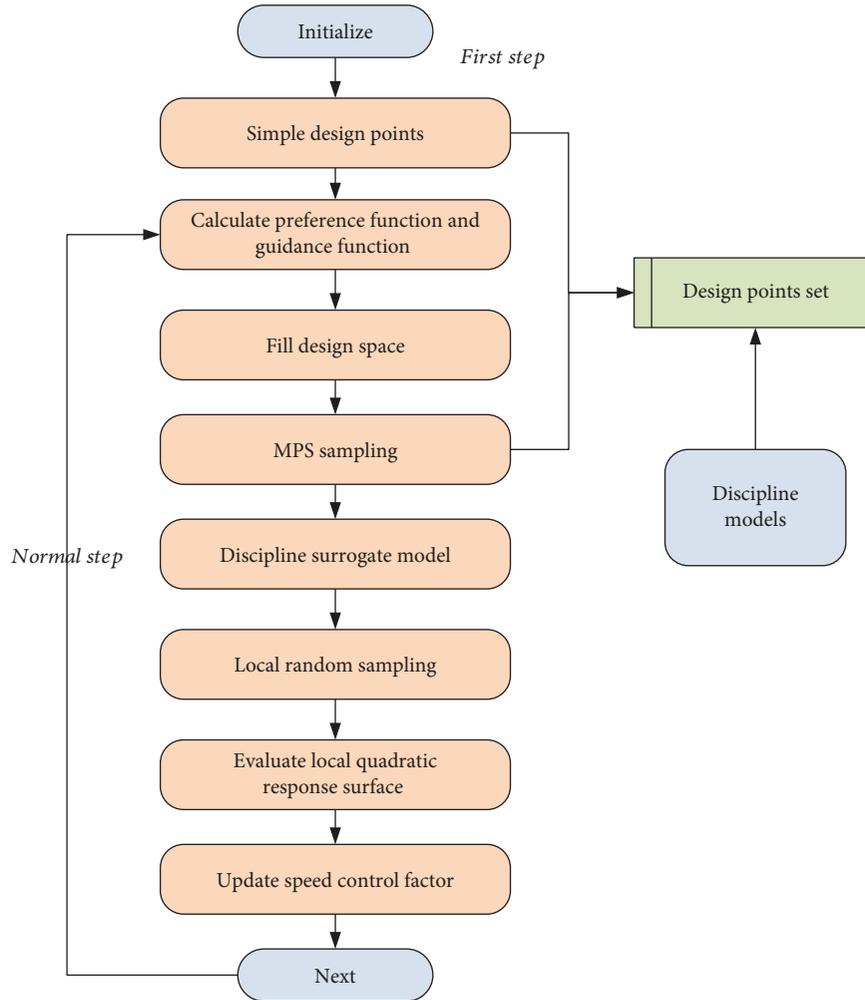


FIGURE 3: Flowchart of the adaptive discipline surrogate model.

*Step 7.* Sample  $N$  discrete sample points in disciplinary design space based on the uniform sampling method and calculate disciplinary preference function  $D(X)$  and MPS guidance function  $p(X)$ .

*Step 8.* Divide the  $N$  sample points into  $K$  discrete regions and calculate the distribution function PDF and CDF of every region by the following equation:

$$G_k(X) = \frac{\sum_{i=1}^k P_i(X)}{\sum_{i=1}^K P_i(X)}, \quad k = 1, 2, \dots, K. \quad (8)$$

*Step 9.* Sample disciplinary design points based on the rules in [16]. The number of sample  $N$  points influences the density of the disciplinary discrete design space. Selecting new design points depends on the position of the sample points. The smaller  $N$  will reduce the approximation precision, thus will increase the iterative times of the discipline surrogate model updating for converging; the bigger  $N$  will increase the execution time of the MPS sampling process but improves the approximation precision. The number of discrete regions  $K$  decides how many sample points are located

in each region. For ordinary problems,  $N = 10000$  and  $K = 100$  is recommended in [4, 7].

*4.1.5. Disciplinary Convergence Criterion.* The local quadratic response surface is used to evaluate the disciplinary convergence precision. The disciplinary convergence criterion will be carried out by the framework level based on the multiple correlation coefficient  $R^2$  of each discipline.

*4.1.6. Speed Control Factor.* The updating of the speed control factor of each discipline surrogate model is the same as the MPS algorithm, which is controlled by  $R^2$ .

*4.1.7. The Adaptive Discipline Surrogate Model.* The flowchart of the discipline level is illustrated in Figure 3. Detailed description of the adaptive discipline surrogate model is given below.

*Step 10.* Sample  $t * m$  initial design points  $X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(t*m)}$  in the disciplinary design space and build DPS by uniting all design points.  $t$  is a positive integer greater than or equal to 1 for providing enough design points to initialize the disciplinary design space reasonably at the first iteration.

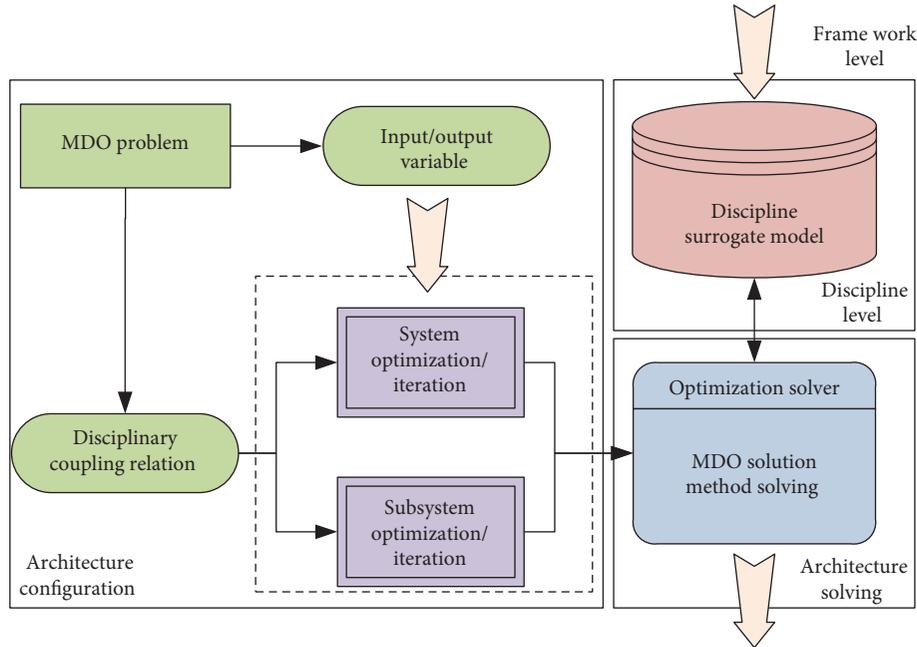


FIGURE 4: Flowchart of architecture level.

*Step 11.* Calculate the preference function  $D(X)$  and MPS guidance function  $p(X)$ .

*Step 12.* Generate the disciplinary discrete design space by sampling  $N$  sample points based on the uniform sampling criteria.

*Step 13.* Calculate CDF and correct it by a speed control factor, execute the MPS sampling process by the fixed CDF \* to get  $m$  new design points, and add it to DPS.

*Step 14.* Rebuild the discipline surrogate model by DPS.

*Step 15.* Generate local quadratic response surface by uniform sampling criteria in the local region around the new sampled design points and compare with discipline surrogate model to evaluate the disciplinary convergence precision  $R^2$ .

*Step 16.* Fix the speed control factor by  $R^2$  and repeat Steps 11–15 until the framework level convergence criterion is met.

**4.2. Architecture Level Design.** Architecture level has a high efficiency because of the replacement of discipline model with discipline surrogate model. The complex interaction between the MDO solution method and discipline model is avoided, and also with the continuity and smoothness characteristic of the discipline surrogate model, a better convergence performance can be achieved. Architecture level is designed as two parts: the architecture configuration and the architecture solving, as shown in Figure 4. The main function of these is as follows:

(a) **Architecture configuration:** according to the detail architecture configuration of the selected MDO

solution method, the MDO problem is transformed to a system level optimization/iteration problem and some subsystem level optimization/iteration problems, and then, the MDO model can be generated

(b) **Architecture solving:** the MDO model will be solved after finishing the architecture configuration based on the latest updated discipline surrogate models, and then, the current optimal result is returned to the framework level; a single MDO iteration is completed

The architecture solving has less computational requirements, so the success rate and the convergence precision are the primary purpose of the architecture level. The detail illustration of some common MDO solution methods can be found in [1, 3, 17, 18].

**4.3. Framework Level Design.** Framework level initializes the MDO solving procedure, manages the iteration of the discipline level and the architecture level, and provides the basic support with auxiliary modules and convergence criterion.

**4.3.1. Design Flowchart.** The flowchart of the framework level is shown in Figure 5. Framework level includes four parts: initialization, execution management, auxiliary modules, and convergence criterion. Initialization initializes the MDO solving procedure by user's configuration. Execution management controls the iteration of the discipline level and the architecture level alternately to drive the entire framework process. Auxiliary modules give elementary support which includes variable interface, approximation method, optimization algorithm, and MDO solution method. Variable interface gives a transition between disciplinary input/output variables and MDO variables; radial basis function (RBF) is chosen to

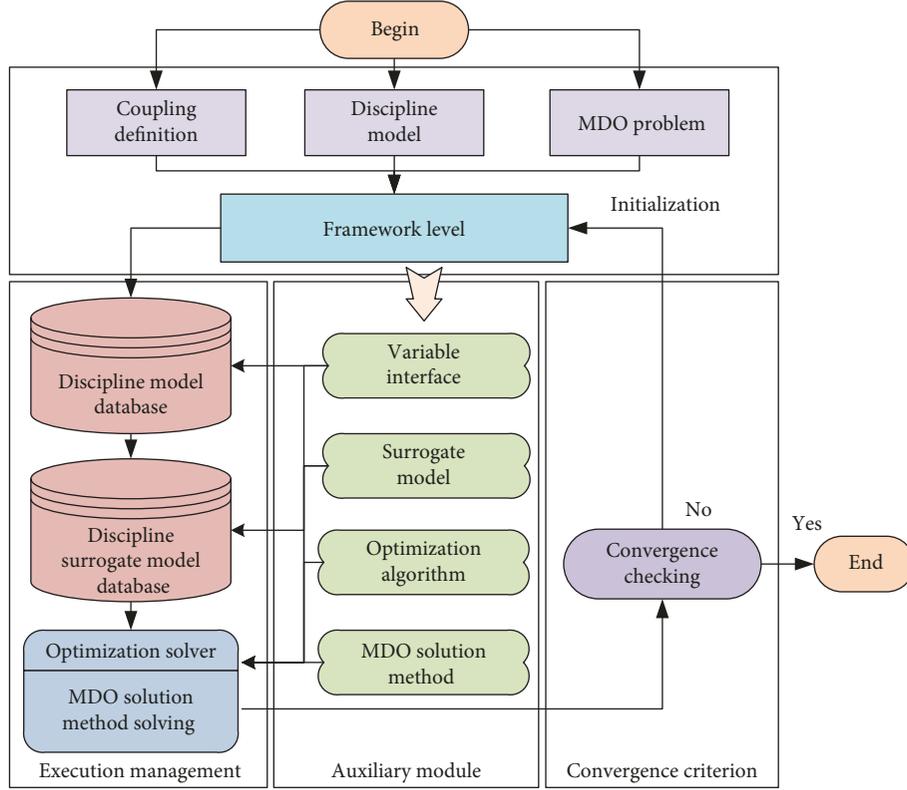


FIGURE 5: Flowchart of framework level.

build the discipline surrogate model; sequential quadratic programming (SQP) method is chosen as the optimization algorithm of the constructed system optimization/iteration problem and subsystem optimization/iteration problems. Three MDO solution methods are chosen to test the DMPS-MDOF, which includes individual discipline feasible (IDF), multidisciplinary feasible (MDF), and concurrent subspace optimization with response surface (CSSO-RS). Detail description of the above MDO solution methods can be found in [1, 3]. A hybrid convergence criterion is executed at every MDO iteration to carry a convergence checking, which will be described in the next.

**4.3.2. Convergence Criterion.** Both of the local response surface fitting precision of the discipline surrogate model and the state of the current MDO optimal solution are considered as the convergence criteria of the DMPS-MDOF. The multiple correlation coefficient  $R^2$  of the local response surface can only assess the fitting precision of the local region in each discipline. So, the following hybrid multistep method is used to check the convergence state of the MDO framework.

The convergence criterion of the discipline surrogate model is defined as

$$\begin{aligned} \varepsilon_i &= 1 - R_i^2, \\ \varepsilon_i &\leq \varepsilon_{\text{disp}}, \quad i = 1, 2, \dots, N. \end{aligned} \quad (9)$$

The convergence criterion of the MDO problem is defined as

$$\varepsilon_0 = \begin{cases} \left| \frac{f^{(k+1)} - f^{(k)}}{f^{(k)}} \right|, & \text{if } f^{(k)} \neq 0, \\ \left| f^{(k+1)} - f^{(k)} \right|, & \text{if } f^{(k)} = 0, \end{cases} \quad (10)$$

$$\varepsilon_0 \leq \varepsilon_{\text{fun}},$$

where  $f$  is the objective function of the MDO problem,  $\varepsilon_{\text{disp}}$  and  $\varepsilon_{\text{fun}}$  are the maximum error of local fitting precision of the discipline surrogate model and the rate of objective function change, and  $k$  is the  $k$ th MDO iteration. There are three conditions in convergence criterion.

- (a) When  $\varepsilon_0 \leq \varepsilon_{\text{fun}}$  is satisfied  $n_1$  times continuously, and  $\varepsilon_i \leq \varepsilon_{\text{disp}}$  is satisfied for all disciplines in current MDO iteration step, where  $i = 1, 2, \dots, N$ , it is considered that the fitting precision of the local response surface of all discipline surrogate models in the current optimal region is satisfied. Convergence criterion checking is successful
- (b) When  $\varepsilon_0 \leq \varepsilon_{\text{fun}}$  is satisfied  $n_1$  times continuously, the convergence criterion of discipline surrogate model is not entirely satisfied with all disciplines. This situation could be the reason for inaccuracy

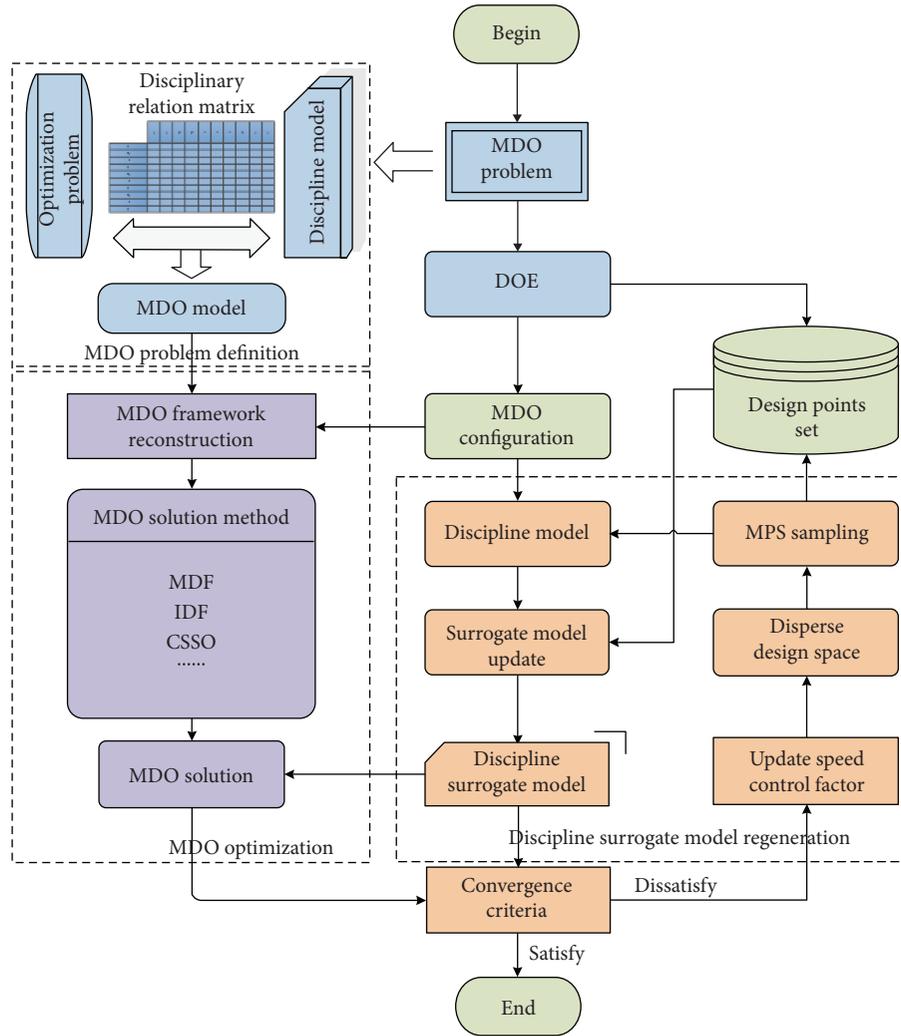


FIGURE 6: Flowchart of the DMPS-MDOF.

of disciplinary local fitting region in current MDO iteration step. The speed control factor is reinitialized to 0.5 to give discipline more chances to explore the global design space

- (c) When  $\varepsilon_i \leq \varepsilon_{\text{disp}}$  is satisfied  $n_2$  times continuously for all disciplines, the convergence criterion of objective function does not meet the above detection (a) and (b). It is considered that the MDO problem is converged into current optimal region, but the local fitting precision of discipline surrogate models is not satisfied. The speed control factor is set to  $S_{\text{max}}$  to accelerate convergence, and MDO iteration continues

The above hybrid multistep convergence criterion is conducive to accelerate convergence, improve convergence precision, and avoid premature convergence as much as possible.

**4.4. Executing Process.** With the three level design pattern, the flowchart of the DMPS-MDOF is shown in Figure 6. The entire implementation process can be described as three parts: MDO problem definition, MDO optimization, and

discipline surrogate model regeneration. MDO problem definition transforms the MDO problem into a corresponding MDO model based on disciplinary relation matrix (DRM) [17], which gives a standardized format to the MDO problem; MDO optimization decomposes the MDO model into a system optimization/iteration problem and some subsystem optimization/iteration problems depending on the selected MDO solution method and optimizes it with the discipline surrogate model instead of the discipline model. The framework level and architecture level are realized in this part; discipline surrogate model regeneration updates and corrects discipline surrogate models by the MPS sampling process, which is illuminated in Section 4.1. Detailed description of the solution process is given below.

*Step 17.* Initialize MDO problem. Define initial value and boundary of the disciplinary input and output variables, set objective function and constraint of the MDO problem, build disciplinary coupling relationship, and select and set up the MDO solution method and optimization algorithm.

*Step 18.* Reconstruct MDO framework. Depending on the MDO problem and the selected MDO solution method defined in Step 17, construct and initialize the MDO framework.

*Step 19.* Generate initial design points. Sample  $t * m$  design points based on the optimal Latin hypercube sampling (LHS) method and build and evaluate the DPS. The size of  $t$  depends on the dimension of disciplinary input variables.

*Step 20.* Disperse disciplinary design space. Each discipline sample  $N$  uniformly distributed sample points to fill disciplinary design space, and evaluate the disciplinary preference function, guidance function, and CDF.

*Step 21.* MPS sampling. Each discipline sample  $m$  design points using the MPS sampling process, and add these points into the DPS.

*Step 22.* Analyze discipline model. Analyze all discipline models which are new added into DPS. This is the most time-consuming process of the MDO solving procedure, but for the independence of each design point, this process can be speeded up greatly by parallel computing.

*Step 23.* Set up/update the discipline surrogate model. Set up or update the discipline surrogate model based on the DPS and calculate the multiple correlation coefficient  $R^2$  of each discipline.

*Step 24.* Solve the MDO problem. Solve the MDO problem with the MDO framework constructed in Step 18 and the discipline surrogate model set up in Step 23. It has a small amount of computation due to the use of the surrogate model.

*Step 25.* Update the speed control factor. The speed control factor  $s$  is updated based on lasted evaluated disciplinary multiple correlation coefficient  $R^2$ .

*Step 26.* Check the convergence criterion. The framework level carries convergence criterion checking based on the multiple correlation coefficient and the rate of objective function change; if the convergence criterion is satisfied, MDO iteration exits; otherwise, go to Step 20.

For some simple discipline models, there is no need to use the discipline surrogate model. So, the discipline surrogate model can be replaced with the true discipline model in Step 24 to get a better solution performance.

## 5. MDO Benchmarks

In this section, we will give a performance analysis to the proposed DMPS-MDOF with some typical MDO benchmarks.

*5.1. A Single Example.* First, a single MDO example with two disciplines is used to illustrate the DMPS-MDOF solving procedure. The optimization problem is as follows:

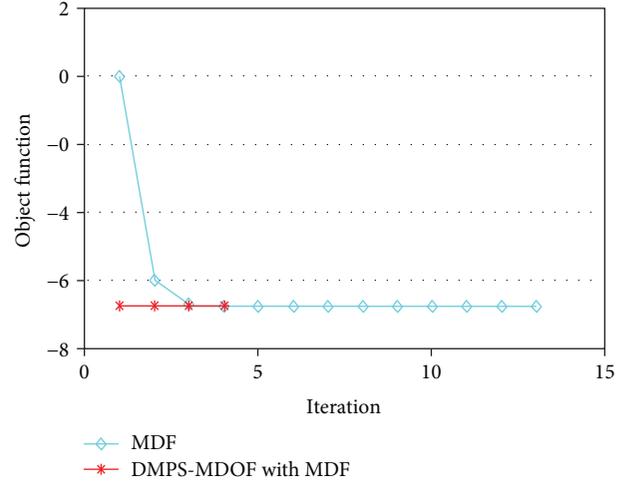


FIGURE 7: Iteration history on the objective function.

$$\begin{aligned}
 \min \quad & 0.25(x_1^4 + x_2^4 + x_3^4) - 2.0(x_1^3 + x_2^3 + x_3^3) \\
 & + 5.5(x_1^2 + x_2^2 + x_3^2) \\
 & - 6.0(4.0y_1 - 2y_2 - 2x_2 - 3x_3) \\
 \text{w.r.t.} \quad & x_1, x_2, x_3 \\
 \text{s.t.} \quad & 0 \leq x_1 \leq 4 \\
 & 0 \leq x_2 \leq 4 \\
 & 0 \leq x_3 \leq 4.
 \end{aligned} \tag{11}$$

Discipline 1:

$$\begin{aligned}
 & \text{Given } x_1, x_3, y_2, \\
 & y_1 = x_1 + x_3 - y_2, \\
 & \text{Return } y_1.
 \end{aligned} \tag{12}$$

Discipline 2:

$$\begin{aligned}
 & \text{Given } x_2, x_3, y_1, \\
 & y_2 = y_1 - x_2 - x_3, \\
 & \text{Return } y_2.
 \end{aligned} \tag{13}$$

The parameters of DMPS-MDOF are configured as follows:  $K = 100$ ,  $N = 10000$ ,  $m = 6$ ,  $t = 4$ , and the MDF is selected as the MDO solution method. Convergence precision of the objective function and the discipline surrogate model is set to  $1e-6$  and  $1e-4$ . The analytical solution is  $-6.75$  with optimal position at  $(3.0, 1.0, 1.0)$ . A comparison is given with the original MDF method. The iteration history on the objective function of the two methods can be found in Figure 7 and Table 1. All of the two methods converged to the optimal solution. The relative error of the objective function and the design variables are less than  $1.6e-6$ , slightly worse than the original MDF method, but it is very close to the optimal analytical solution. The number of discipline evaluations

TABLE 1: Comparison between DMPS-MDOF with MDF and the original MDF method.

	DMPS-MDOF with MDF	Original MDF method
Objective function	-6.75000016	-6.75
Consistency	$3.20408e-8$	$2.41121e-8$
$x_1$	3.00000066	2.99999986
$x_2$	0.99999843	0.99999922
$x_3$	0.99999952	0.99999992
Number of discipline 1 evaluations	48	603
Number of discipline 2 evaluations	48	603
CPU time (s)	3.16	0.5

has a significant difference between the DMPS-MDOF and the original MDF method. DMPS-MDOF only needs less than a tenth of the total discipline counts of the original MDF method. For this simple example with cheap discipline analysis, DMPS-MDOF needs more CPU time than the original MDF method because of the generation of the surrogate model and the multiple optimization processes. But generally the modern MDO problem has more complex disciplines with time-consuming analysis which is far more expensive than the surrogate model. Compared to the time-consuming discipline model evaluations, the time cost by the generation of the surrogate model and the multiple optimization processes are almost negligible.

The iteration history on the distribution of discrete design points can be found in Figure 8 and Table 2. The closed lines are the convex hulls of the current sampling design points per iteration. Design points at each iteration are shown with different colors and different marks. As iteration goes on, the MPS sampling design points move round the current optimal solution. Meanwhile, fitting precision of the local response surface improves with the correction process of the discipline surrogate model and the region of the current sampling design points also becomes smaller. Then, the smaller region prompts the current local response surface to improve the fitting precision. With the change of the optimal solution, value becomes smaller; the global convergence criterion will be satisfied finally.

The first iteration in Table 2 is utilized to generate more design points to give a better global initialization of the discipline surrogate model. For small problem, 4 times of the normal sampling design points are compatible. The asymptotic randomness of the MPS sampling process will influence the size of the updating region. The region of design points becomes larger when the discipline surrogate model has a lower local fitting precision or the discipline convergence cannot be satisfied. Another interesting phenomenon in Figure 8 is that the region of the last iteration in red was larger than others. The reason is that the convergence criterion (c) is met at the third iteration and the speed control factor is reinitialized to give more chances to explore the global design space. This characteristic can avoid prematurity of the

searching process and prevent the illogical convergence checking with low global fitting precision.

5.2. *Test Cases.* The above single example gave an intuitive understanding of the characteristic of DMPS-MDOF. Another three test cases are used to analyze comprehensive performance. The first one is an analytic example has been previously solved by Sellar et al. [19]. It is a two-discipline MDO problem and has low dimensionality. The second is adapted from NASA's MDO test suite [20] and represents the design of a simple gearbox. As the problem has originally been solved as a single discipline optimization problem, the MDO problem is created by splitting up the single discipline problem into gear 1, gear 2, and gear shaft three parts with no coupling. The third example is an aircraft problem with aerodynamic, mass, and performance disciplines. The three MDO test cases are listed as follows.

(a) Analytic problem

$$\begin{aligned}
 \min \quad & x_2^2 + x_3 + y_1 + e^{-y_2} \\
 \text{w.r.t.} \quad & x_1, x_2, x_3 \\
 \text{s.t.} \quad & 1 - \frac{y_1}{3.16} \leq 0 \\
 & \frac{y_2}{24} - 1 \leq 0 \\
 & -10 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 10 \\
 & 0 \leq x_3 \leq 10.
 \end{aligned} \tag{14}$$

Discipline 1:

$$\begin{aligned}
 \text{Given } & x_1, x_2, x_3, y_2, \\
 y_1 = & x_1^2 + x_2 + x_3 - 0.2y_2, \\
 \text{Return } & y_1.
 \end{aligned} \tag{15}$$

Discipline 2:

$$\begin{aligned}
 \text{Given } & x_1, x_3, y_1, \\
 y_2 = & \sqrt{y_1} + x_1 + x_3, \\
 \text{Return } & y_2.
 \end{aligned} \tag{16}$$

(b) Golinski's speed reducer

$$\begin{aligned}
 \min \quad & C_1 x_1 x_2^2 (C_2 x_3^2 + C_3 x_3 - C_4) - C_5 x_1 (x_6^2 + x_7^2) \\
 & + C_6 (x_6^3 + x_7^3) + C_7 (x_4 x_6^2 + x_5 x_7^2) \\
 \text{w.r.t.} \quad & x_1, x_2, x_3, x_4, x_5, x_6, x_7.
 \end{aligned} \tag{17}$$

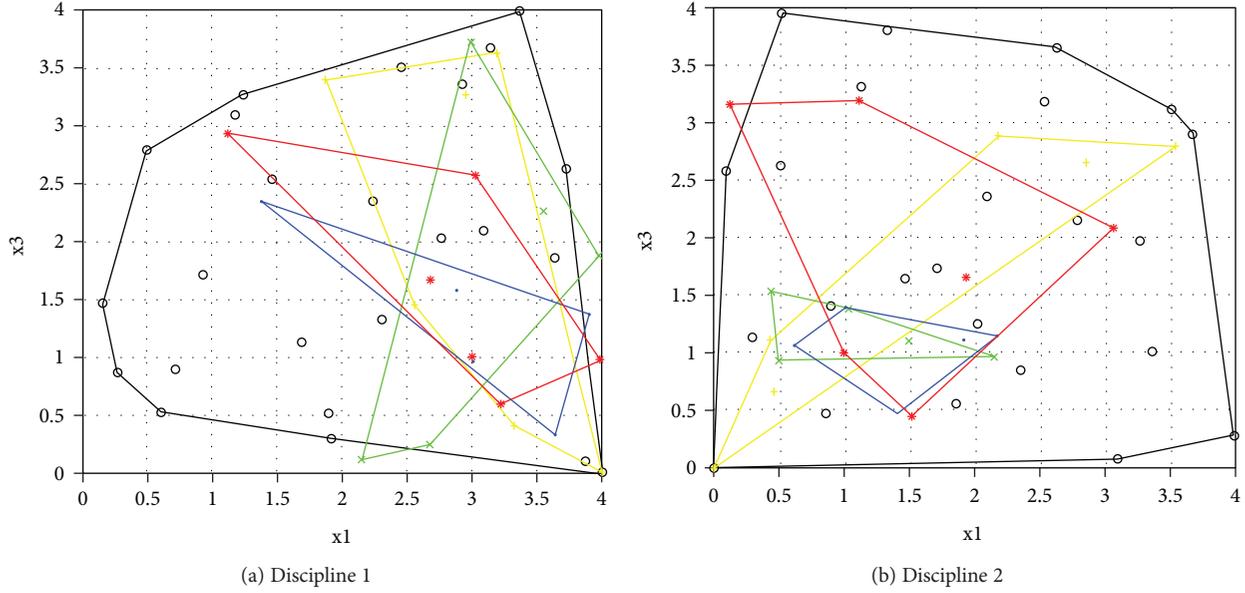


FIGURE 8: Iteration history on the distribution of discrete design points of DMPS-MDOF.

TABLE 2: Iteration history on variables of discipline 1 and discipline 2.

Iteration	Marks	Number of discipline evaluations	
		Discipline 1	Discipline 2
0	O (black)	24	24
1	+ (yellow)	6	6
2	X (green)	6	6
3	. (blue)	6	6
4	* (red)	6	6

Discipline 1:

$$\begin{aligned} &\text{Given } x_2, x_3, x_4, x_6, \\ &g_3, g_5, g_{10} \leq 0. \end{aligned} \quad (18)$$

Discipline 2:

$$\begin{aligned} &\text{Given } x_2, x_3, x_5, x_7, \\ &g_4, g_6, g_{11} \leq 0. \end{aligned} \quad (19)$$

Discipline 3:

$$\begin{aligned} &\text{Given } x_1, x_2, x_3, \\ &g_1, g_2, g_7, g_8, g_9 \leq 0. \end{aligned} \quad (20)$$

(c) Aircraft problem

$$\begin{aligned} &\min W_{to} \\ &\text{s.t. } R \geq 901.23 \text{ km} \\ &V_s \leq 21.336 \text{ m/s}. \end{aligned} \quad (21)$$

Discipline 1:

$$\begin{aligned} &\text{Given } AR, S_w, L_f, D_f, \\ &L/D = 10.0 + 4.0 \left( \frac{AR}{S_{wet}/S_w} - 1 \right), \\ &S_{wet} = S_{fuse} + 2S_w + 3 \times 0.2S_w, \\ &S_{fuse} = \pi L_f D_f, \\ &\text{Return } L/D. \end{aligned} \quad (22)$$

Discipline 2:

$$\begin{aligned} &\text{Given } L_f, D_f, \rho_c, V_c, L/D, W_{fuel}, \\ &W_{to} = 3(W_{fuse} + W_{fuel}), \\ &W_{fuse} = 2.862L_f^{1.035} D_f^{1.086} \rho_c^{0.241} V_c^{0.482} (L/D)^{-0.072}, \\ &\text{Return } W_{to}. \end{aligned} \quad (23)$$

Discipline 3:

$$\begin{aligned} &\text{Given } L/D, W_{to}, W_{fuel}, \rho_c, S_w, \\ &R = \eta \frac{L/D}{c} \ln \left( \frac{W_{to}}{W_{to} - 0.85W_{fuel}} \right), \\ &V_s = \sqrt{\frac{2W_{to}}{\rho_c S_w C_{L_{max}}}}, \\ &\text{Return } R, V_s. \end{aligned} \quad (24)$$

The parameters of DMPS-MDOF are set as follows:  $K = 100$ ,  $N = 10000$ ,  $m = 6$ , and  $t = 4$ . Convergence precision

TABLE 3: Result comparison of the analytic problem.

Benchmark	MDO solution method	Framework	Number of discipline evaluations		Optimal value
			Discipline 1	Discipline 2	
Analytic problem	MDF	Origin	203	203	3.18341
		DMPS-MDOF	78	81	3.18339
	IDF	Origin	139	139	3.18339
		DMPS-MDOF	57	60	3.18339
	CSSO	Origin	339	432	3.18340
		DMPS-MDOF	258	264	3.18339

TABLE 4: Result comparison of the Golinski's speed reducer problem.

Benchmark	MDO solution method	Framework	Number of discipline evaluations			Optimal value
			Discipline 1	Discipline 2	Discipline 3	
Golinski's speed reducer	MDF	Origin	73	73	73	2994.35
		DMPS-MDOF	54	51	54	2994.35
	IDF	Origin	73	73	73	2994.35
		DMPS-MDOF	54	51	54	2994.35
	CSSO	Origin	773	1021	1071	2994.35
		DMPS-MDOF	54	54	54	2994.35

TABLE 5: Result comparison of the aircraft problem.

Benchmark	MDO solution method	Framework	Number of discipline evaluations			Optimal value
			Discipline 1	Discipline 2	Discipline 3	
Aircraft problem	MDF	Origin	638	638	638	875.654
		DMPS-MDOF	210	219	231	875.654
	IDF	Origin	213	213	213	875.654
		DMPS-MDOF	87	78	81	875.654
	CSSO	Origin	5667	5401	6815	875.654
		DMPS-MDOF	69	81	77	875.654

of the objective function and the discipline surrogate model is set to  $1e-6$  and  $1e-4$ . SNOPT [21], an efficient optimization library with the SQP method, is used as the system and sub-system's optimization solver. MDF, IDF, and CSSO-RS are chosen as the MDO solution methods.

Results of the three test cases are listed in Tables 3 to 5. All of the three test cases with different MDO solution methods found out the optimal solution, but there is an obvious difference in the number of discipline evaluations between the DMPS-MDOF and the original MDO solution method. Some conclusions are given from computing efficiency, parallel computing, architecture, and add-on capability as follows.

**5.2.1. Computing Efficiency.** From the results, we know that at the same precision level, DMPS-MDOF with the discipline surrogate model has higher computing efficiency than the original MDO solution method with the discipline model directly. The number of discipline evaluations of DMPS-MDOF is half or less of the original MDO solution method in the three test cases. Especially for distributed architecture,

such as CSSO, there is a significant reduction. The MDO solving process has a lightweight computation because of the full use of discipline surrogate models instead of the time-consuming original discipline models. The discipline models can be evaluated separately from the MDO solving process in the independent discipline level. This is helpful for the complex MDO optimization with high-fidelity discipline models.

**5.2.2. Parallel Computing.** DMPS-MDOF is a hierarchic framework based on the independent discipline models. The discipline models and disciplinary design points are all irrelevant; parallel computing could be utilized to improve the computing efficiency. For example, in a MDO problem with 5 disciplines, we suppose that each iteration uses 6 design points for every discipline, then a total 30 times of discipline model evaluations are needed for each iteration in this framework. If every discipline evaluation cost 5 minutes, the total CPU time will cost 2 hours and a half for one iteration. But the 30 times discipline model evaluations in each iteration are all independent. It only needs about 5

minutes if we have enough computing resources. So, the computing efficiency will be improved greatly by using parallel computing in DMPS-MDOF.

**5.2.3. Architecture.** Another obvious phenomenon is that the discipline evaluations reduction between DMPS-MDOF and the original MDO solution method are not consistent for each MDO solution method. For example, in the analytic problem, CSSO is the most inefficient one with the original MDO solution method and DMPS-MDOF and in the aircraft problem, CSSO is also the most inefficient one with the original MDO solution method but the most efficient one is with DMPS-MDOF. Benefited from high computing efficiency in DMPS-MDOF, it is acceptable and comfortable to select the most suitable MDO solution method for a given MDO problem in DMPS-MDOF.

**5.2.4. Add-On Capability.** The MDO solution method, approximation method, and optimization algorithm are kept separated from the discipline model; these are designed as some algorithm libraries with generic interfaces in DMPS-MDOF. It is convenient to add new method to the algorithm library for extension. And also it is convenient for designers to participate in the MDO solving procedure with DMPS-MDOF. Designers can offer the following assistances: (1) Designers can modify, add, or remove design points per iteration according to the convergence state of the current iteration; (2) The MDO solution method, approximation method, and optimization algorithm can be changed per iteration manually by designers when the current one is hard to converge or has low computing efficiency; (3) Designers even can adjust the MDO problem per iteration to obtain a more efficient and more easy-solved way by reducing design space or removing inactive constraint condition.

## 6. Conclusion

In this article, the adaptive discipline surrogate model with the MPS sampling process is used to replace the time-consuming discipline model and a new universal MDO framework is developed to implement an effective and parallel MDO solving environment. This framework is based on a universal MDO model with standard coupling relationship definition, which is discussed in greater detail in [17, 18]. So, an automatic decomposing and solving procedure with different MDO solution methods is realized to give a universal and comfortable way to carry out this framework.

Based on the adaptive discipline surrogate model, DMPS-MDOF has an excellent computing capability and clear framework architecture. The independent disciplines and MPS sampling process give a remarkable adaptation to parallel computing environment, which will bring further improvement for the computing efficiency. It is easy to change MDO solution methods or approximation methods in DMPS-MDOF. Designers could choose the most appropriate one flexibly to get a better performance. Designer's experience and decision-making capacity could also be realized easily in DMPS-MDOF. These attractive characteristics of DMPS-MDOF make it extremely useful for solving

the complex MDO problem with high-fidelity and time-consuming discipline models.

There are also some limits. DMPS-MDOF can solve the MDO problem more efficiently and carry out in parallel way with various MDO solution methods, and it has no obvious difference on the convergence characteristic compare to the original MDO solving procedure, so DMPS-MDOF cannot give a right answer to the problem that the original MDO solution method cannot solve. A possible way to improve the optimization performance is to carry out the MDO iteration process with different MDO solution methods in a nested manner. This will give a more flexible way to organize the MDO iteration process and may further increase the convergence performance. For simple MDO problem with analytic discipline model, DMPS-MDOF may likely cost more CPU time because of the surrogate model generation and the multiple optimization processes, so it is more appropriate for complex MDO problems with high-fidelity discipline models.

Future research will focus on the hybrid process combining with monolithic architecture and distributed architecture. More tests will be given to evaluate the performance of the DMPS-MDOF, especially for some complex MDO problems with high-fidelity discipline models. The proper distributed parallel computing environment with workstation clusters will be carried out to test and verify the parallel solving performance of the DMPS-MDOF.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the funding from the National Natural Science Foundation of China (no. 51505385), the Shanghai Aerospace Science Technology Innovation Foundation (no. SAST2015010), and the Defense Basic Research Program (JCKY2016204B102 and JCKY2016208C001). The authors are thankful to the National Key Laboratory of Aerospace Flight Dynamics of NPU.

## References

- [1] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.
- [2] T. W. Simpson and J. R. R. A. Martins, *The Future of Multidisciplinary Design Optimization (MDO): Advancing the Design of Complex Engineered Systems*, NSF Workshop Report, Fort Worth, TX, USA, 2010.
- [3] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette, "A survey of multidisciplinary design optimization methods in launch vehicle design," *Structural and Multidisciplinary Optimization*, vol. 45, no. 5, pp. 619–642, 2012.

- [4] L. Wang, S. Shan, and G. G. Wang, "Mode-pursuing sampling method for global optimization on expensive black-box functions," *Engineering Optimization*, vol. 36, no. 4, pp. 419–438, 2004.
- [5] Y. M. Deng, D. Zheng, and Y. A. Zhang, "A modified mode-pursuing-sampling based optimization method for minimization of injection molding warpage," in *2009 Fifth International Conference on Natural Computation*, vol. 5, pp. 317–321, Tianjian, China, 2009.
- [6] L. An, Z. Wang, G. Gary Wang, and Z. Li, "Design optimization of base widths of transmission tower using mode-pursuing sampling global optimization method," in *2010 International Conference on Computer Application and System Modeling (ICCA SM 2010)*, vol. 6, pp. V8–V257, Taiyuan, China, 2010.
- [7] X. Duan, G. G. Wang, X. Kang, Q. Niu, G. Naterer, and Q. Peng, "Performance study of mode-pursuing sampling method," *Engineering Optimization*, vol. 41, no. 1, pp. 1–21, 2009.
- [8] B. Sharif, G. Gary Wang, and T. Y. ElMekkawy, "Mode pursuing sampling method for discrete variable optimization on expensive black-box functions," *Journal of Mechanical Design*, vol. 130, no. 2, article 021402, 2008.
- [9] S. Shan and G. Gary Wang, "An efficient Pareto set identification approach for multiobjective optimization on black-box functions," *Journal of Mechanical Design*, vol. 127, no. 5, p. 866, 2005.
- [10] Y. Deng, Y. Zhang, and Y. C. Lam, "A hybrid of mode-pursuing sampling method and genetic algorithm for minimization of injection molding warpage," *Materials and Design*, vol. 31, no. 4, pp. 2118–2123, 2010.
- [11] D. Wang, G. Wang, and G. Naterer, "Advancement of a collaboration pursuing method (CPM)," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2006.
- [12] D. Wang, G. G. Wang, and F. N. Greg, "Collaboration pursuing method for MDO problems," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Austin, TX, USA, 2005.
- [13] D. Wang, G. G. Wang, and G. F. Naterer, "Collaboration pursuing method for multidisciplinary design optimization problems," *AIAA Journal*, vol. 45, no. 5, pp. 1091–1103, 2007.
- [14] D. Wang, "Collaboration pursuing method using Latin hypercube sampling," in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Structures, Structural Dynamics, and Materials and Co-located Conferences*, Newport, RI, USA, 2006.
- [15] D. Wang, G. G. Wang, and G. F. Naterer, "Extended collaboration pursuing method for solver larger multidisciplinary design optimization problems," *AIAA Journal*, vol. 45, no. 6, pp. 1208–1221, 2007.
- [16] J. C. Fu and L. Wang, "A random-discretization based Monte Carlo sampling method and its applications," *Methodology and Computing in Applied Probability*, vol. 4, no. 1, pp. 5–25, 2002.
- [17] H. Su, L. Gu, and C. Gong, "A disciplinary relation matrix based universal multidisciplinary optimization architecture," *Computer Integrated Manufacturing Systems*, vol. 20, no. 4, pp. 731–738, 2014.
- [18] H. Su, *Study on High Fidelity Multidisciplinary Design Optimization Technique of Vehicles*, School of Astronautics Northwestern Polytechnical University, Xian, China, 2014.
- [19] R. S. Sellar, S. M. Batill, and J. E. Renaud, "Response surface based, concurrent subspace optimization for multidisciplinary system design," in *34th Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 1996.
- [20] S. L. Padula, N. M. Alexandrov, and L. L. Green, "MDO test suite at NASA Langley Research Center," in *6th Symposium on Multidisciplinary Analysis and Optimization, Multidisciplinary Analysis Optimization Conferences*, pp. 410–420, Bellevue, WA, USA, 1996.
- [21] E. Philip, E. Wong, W. Murray, and M. A. Saunders, *User's Guide for SNOPT Version 7.4: Software for Large-Scale Nonlinear Programming*, 2015.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

