Mode-Pursuing Sampling Method Using Discriminative Coordinate Perturbation for High-Dimensional Expensive Black-Box Optimization

Yufei Wu
School of Aerospace Engineering,
Beijing Institute of Technology,
Beijing 100081, China
e-mail: wuyufei@bit.edu.cn

Teng Long
School of Aerospace Engineering,
Beijing Institute of Technology,
Beijing 100081, China;
Key Laboratory of Dynamics and Control
of Flight Vehicle,
Ministry of Education,
Beijing 100081, China
e-mail: tenglong@bit.edu.cn

Renhe Shi[1]
School of Aerospace Engineering,
Tsinghua University,
Beijing 100084, China
e-mails: srenhe@163.com;
shirenhe@bit.edu.cn

G. Gary Wang
School of Mechatronic Systems Engineering,
Simon Fraser University,
Surrey, BC V3T 0A3, Canada
e-mails: gwa5@sfu.ca;
gary_wang@sfu.ca

This article presents a novel mode-pursuing sampling method using discriminative coordinate perturbation (MPS-DCP) to further improve the convergence performance of solving high-dimensional, expensive, and black-box (HEB) problems. In MPS-DCP, a discriminative coordinate perturbation strategy is integrated into the original mode-pursuing sampling (MPS) framework for sequential sampling. During optimization, the importance of variables is defined by approximated global sensitivities, while the perturbation probabilities of variables are dynamically adjusted according to the number of optimization stalling iterations. Expensive points considering both optimality and space-filling property are selected from cheap points generated by perturbing the current best point, which balances between global exploration and local exploitation. The convergence property of MPS-DCP is theoretically analyzed. The performance of MPS-DCP is tested on several numerical benchmarks and compared with state-of-the-art metamodel-based design optimization methods for HEB problems. The results indicate that MPS-DCP generally outperforms the competitive methods regarding convergence and robustness performances. Finally, the proposed MPS-DCP is applied to a stepped cantilever beam design optimization problem and an all-electric satellite multidisciplinary design optimization (MDO) problem. The results demonstrate that MPS-DCP can find better feasible optima with the same or less computational cost than the competitive methods, which demonstrates its effectiveness and practicality in solving real-world engineering problems.

[DOI: 10.1115/1.4047909]

Keywords: metamodel-based design optimization, metamodeling, multidisciplinary design and optimization

1 Introduction

Expensive black-box simulation models (e.g., finite element analysis, computational fluid dynamics, and computational electromagnetics) have been widely applied to modern engineering systems design practices. The application of expensive simulations significantly increases the computational cost, which becomes a critical challenge for engineering design optimization [1]. To alleviate the computational burden, a series of metamodel-based design optimization (MBDO) methods have been developed [2]. In MBDO methods, cheap-to-evaluate metamodels are constructed based on limited sample points to predict the responses of expensive analysis models. Commonly used metamodels include polynomial response surface (PRS), radial basis function (RBF), Kriging, support vector regression (SVR), and so on [3]. Moreover, a number of novel metamodeling techniques such as multi-model fusion [4–6], ensemble of metamodels [7,8], and gradient-enhanced surrogates [9] have been developed to further improve the approximation accuracy and robustness.

In recent years, a number of novel MBDO methods using adaptive metamodeling techniques have been developed to improve the optimization efficiency and convergence performance [10–13]. In the adaptive MBDO methods, metamodels are constructed and adaptively refined to lead the search to the optimum. The mode-pursuing sampling (MPS) method presented by Wang et al. [14] is a typical adaptive MBDO method, where sequential samples are generated toward the global optimum according to a constructed probability density function. Wang et al. [14] theoretically prove the global convergence of MPS. Owing to the promising global optimization capacity of the mode-pursuing sampling mechanism, a series of MPS variants have been tailored to tackle different kinds of black-box optimization problems. For instances, Kazemi et al. [15] proposed the Constraint-importance Mode-Pursuing Sampling (CiMPS) method for solving constrained optimization problems; using a novel double-sphere strategy, Sharif et al. [16] developed the discrete variable MPS (D-MPS) for solving discrete variable optimization problems.

The effectiveness of MBDO methods for solving middle and low dimensional engineering design optimization problems (i.e., problems with no more than ten variables [17]) have been widely approved in the literature [18–21]. However, the high-dimensional, expensive, and black-box (HEB) optimization problems are still challenging due to the "curse of dimensionality." Since the complexity of optimization grows exponentially with the problem dimensionality, the high-dimensional design space can hardly be thoroughly explored. It is well acknowledged that the efficiency and convergence capability of MBDO methods need to be further improved for HEB optimization problems. As for the original MPS algorithm, it would require tens of thousands expensive function evaluations for solving HEB problems, which generally results in computational memory overflow and unacceptable computational expenses [22,23]. To relieve the computational expense of MPS for solving HEB problems, Cheng et al. [22] developed the trust region-based MPS (TRMPS) method. In TRMPS, two trust

[1] Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received March 15, 2020; final manuscript received June 24, 2020; published online October 12, 2020. Assoc. Editor: James T. Allison.

regions are defined within two hypercubes, respectively, whose sizes are adjusted dynamically. Sequential sampling is performed alternately in the two trust regions. However, the convergence and robustness performances of TRMPS need to be further improved according to the authors' literature survey [3,24].

To address the challenge of solving HEB problems, a novel MPS variant called mode-pursuing sampling method using discriminative coordinate perturbation (MPS-DCP) is proposed in this paper. Different from the trust region method in TRMPS, the coordinate perturbation (CP) strategy from dynamic coordinate search using response surface models (DYCORS) [25] is introduced and customized to guide the discriminative sampling in MPS. Different from the original DYCORS, the proposed discriminative coordinate perturbation technique assigns different priorities to variables in perturbation according to the approximate global sensitivity analysis results to improve the global exploration and local exploitation capacity, which is a new endeavor for solving high-dimensional expensive black-box optimization problems. In this work, the proposed MPS-DCP mainly focuses on HEBs with 10–30 variables, which is the most common in engineering practices. Moreover, the convergence performance and convergence rate of MPS-DCP are theoretically analyzed, which is a contribution for metamodel-based design optimization research.

The rest of the paper is organized as follows. Section 2 briefly reviews the original MPS and DYCORS. Section 3 presents the methodology of the proposed MPS-DCP, especially for the sensitivity analysis based discriminative coordinate perturbation strategy. And the convergence analysis and a customized constraint handling strategy are also presented. In Sec. 4, the proposed MPS-DCP is tested on several numerical benchmarks and compared with state-of-the-art MBDO methods for HEB problems, including MPS, DYCORS, TRMPS, and optimization on metamodeling-supported iterative decomposition (OMID). The effectiveness of MPS-DCP in solving real-world engineering design optimization problems is demonstrated in Sec. 5 through a stepped cantilever optimization problem and an all-electric satellite multidisciplinary design optimization (MDO) problem. The conclusions and future work are summarized in Sec. 6.

2 Review of Mode-Pursuing Sampling Method and Dynamic Coordinate Search

In this section, the original MPS and DYCORS are briefly reviewed, which form the theoretical foundations of the proposed MPS-DCP. Several concepts in MPS and DYCORS are clarified as follows [14,25].

DEFINITION 1 (cheap points). *Cheap points refer to points whose responses are obtained by evaluating a metamodel. Since no expensive simulations is invoked, the computational cost of generating cheap points can be neglected.*

DEFINITION 2 (expensive points). *Expensive points refer to points whose responses are obtained by evaluating the original HEB function. The number of expensive points n_e is regarded as the number of expensive function evaluations. Expensive points are used to construct the RBF metamodel and quadratic PRS metamodel.*

DEFINITION 3 (current best point). *Current best point refers to the expensive point whose objective value is the lowest at the current iteration.*

DEFINITION 4 (current best objective). *Current best objective refers to the objective value of the current best point.*

DEFINITION 5 (optimization improves). *The optimization improves when the current best objective is lower than the best objective of the last iteration.*

DEFINITION 6 (optimization stalls). *The optimization stalls when the current best objective equals to the best objective of the last iteration.*

DEFINITION 7 (step size). *Step size refers to the standard deviation of Gaussian distributed random perturbation.*

2.1 Mode-Pursuing Sampling Method.

The procedure of MPS includes design of computational experiments, approximation models construction and update, and local optimization on the metamodel, which is introduced as follows [14].

The number of initial expensive points equals to [14]

$$n_0 = \frac{(n_v+1)(n_v+2)}{2} + 1 - n_v \tag{1}$$

where n_v is the number of variables. A linear RBF metamodel is constructed. Then, numerous cheap points $\boldsymbol{x}_c$ are uniformly generated in the entire design space, and their responses are evaluated by the RBF metamodel. The cheap points are sorted in ascending order by RBF responses and then evenly divided into K groups. The average response values of cheap points in each group are calculated to form a cumulative distribution function on the groups, which defines the probabilities of the groups to be chosen. The distribution function is adjusted using a speed control factor r to bias toward groups with lower average responses. Then, expensive points are selected from cheap points based on the distribution function. A quadratic PRS metamodel is built using a specific number of expensive points around the current best point $\boldsymbol{x}_e^*$ and is expressed as

$$\hat{f}_{\mathrm{PRS}}(\boldsymbol{x}) = \beta^{(0)} + \sum_{i=1}^{n_v}\beta^{(i)}x^{(i)} + \sum_{i=1}^{n_v}\beta^{(ii)}\left(x^{(i)}\right)^2 + \sum_{i=1}^{n_v}\sum_{j>i}^{n_v}\beta^{(ij)}x^{(i)}x^{(j)} \tag{2}$$

where coefficients $\beta^{(0)}$, $\beta^{(i)}$, $\beta^{(ii)}$, and $\beta^{(ij)}$ can be calculated via least squares approximation. As mentioned in Ref. [14], if the quadratic PRS metamodel could accurately approximate the expensive objective within a small sub-region around the current optimum, it is reasonable to believe that the optimization reaches the unimodal area around the global optimum, in which the local optimization is performed. The coefficient of determination R^2 and the maximum absolute error in terms of L_∞-norm *Diff* are used to evaluate metamodel approximation accuracy, as calculated below:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_t}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{\sum_{i=1}^{n_t}\left(y^{(i)} - \bar{y}^{(i)}\right)^2}, \quad Diff = \max\left\{\left|y^{(i)} - \hat{y}^{(i)}\right|, i = 1, \ldots, n_t\right\} \tag{3}$$

where $\hat{y}^{(i)}$ and $y^{(i)}$ are the approximated value and objective value of the expensive points, respectively, $\bar{y}^{(i)}$ is the mean value of the objective values, and n_t is the number of test points. n_t is calculated by Eq. (4) [14]:

$$n_t = \frac{(n_v+1)(n_v+2)}{2} + 1 + \frac{n_v}{2} \tag{4}$$

The local optimization is performed on the quadratic PRS metamodel, and the resulting best point is evaluated by the expensive objective. When the optimization termination criterion [14] is reached, the algorithm terminates.

2.2 Dynamic Coordinate Search Using Response Surface Models.

The contour-based discriminative sampling strategy in the original MPS is replaced by a discriminative coordinate perturbation strategy in the proposed MPS-DCP. In DYCORS, numerous cheap points are generated by applying perturbations on some variables of the current best point. The perturbed variables are randomly selected at each iteration. The perturbation obeys the normal distribution with a mean value of zero and a specific standard deviation (referred to as the step size). The step size is adjusted according to the number of improving or stalling iterations. The perturbation probability of coordinates decreases with the number of expensive points, which is calculated by Eq. (5) [25]:

$$p = \varphi(n_e) = \min\left\{\frac{20}{n_v}, 1\right\} \cdot \left(1 - \frac{\ln(n_e - n_0 + 1)}{\ln(N_{\max} - n_0)}\right) \tag{5}$$

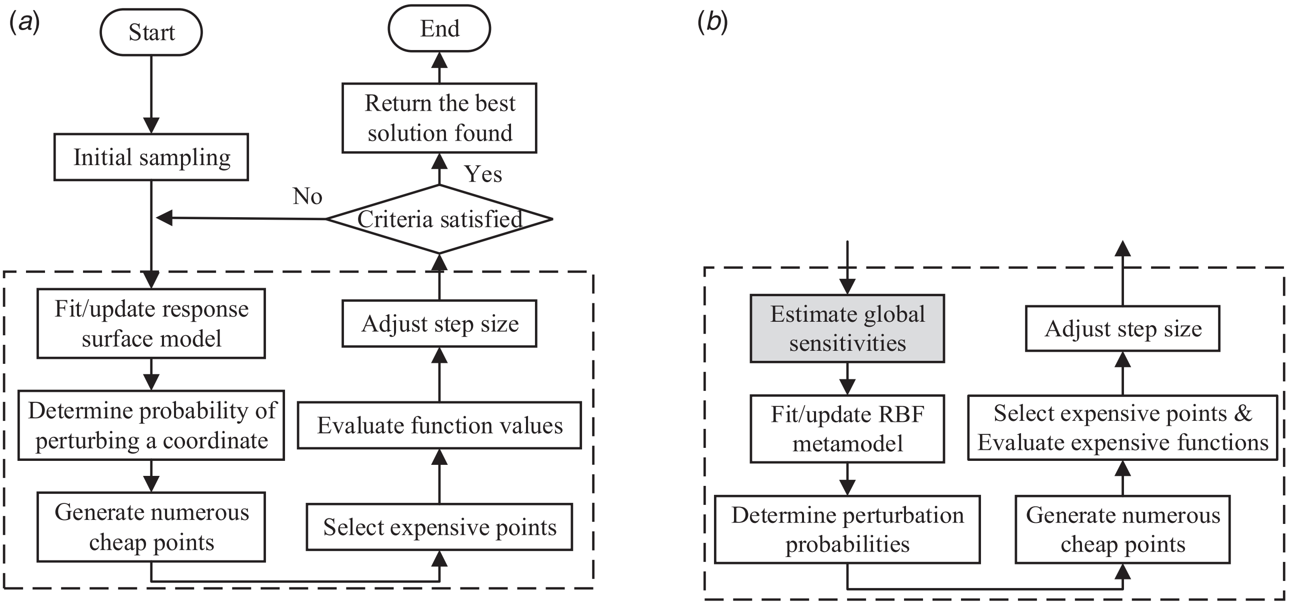
(a)
Start
Initial sampling
End
Return the best solution found
Yes
No
Criteria satisfied
Fit/update response surface model
Adjust step size
Determine probability of perturbing a coordinate
Evaluate function values
Generate numerous cheap points
Select expensive points
(b)
Estimate global sensitivities
Adjust step size
Fit/update RBF metamodel
Select expensive points & Evaluate expensive functions
Determine perturbation probabilities
Generate numerous cheap points

Fig. 1 Flowcharts of the coordinate perturbation processes: (*a*) coordinate perturbation in DYCORS and (*b*) coordinate perturbation in MPS-DCP

Expensive points are selected from the cheap points with lower approximated value evaluated by multiquadric RBF metamodels and larger distance from existing expensive points. The weighted sum of the normalized RBF prediction and the opposite of the minimum distance from existing expensive points is shown in Eq. (6) [25]:

$$c(\boldsymbol{x}_c) = w \cdot \frac{\hat{f}_{\mathrm{RBF}}(\boldsymbol{x}_c) - \min\limits_{\boldsymbol{x}_c \in X_e} \{\hat{f}_{\mathrm{RBF}}(\boldsymbol{x}_c)\}}{\max\limits_{\boldsymbol{x}_c \in X_e} \{\hat{f}_{\mathrm{RBF}}(\boldsymbol{x}_c)\} - \min\limits_{\boldsymbol{x}_c \in X_e} \{\hat{f}_{\mathrm{RBF}}(\boldsymbol{x}_c)\}} + (1-w) \cdot \frac{\max\limits_{\boldsymbol{x}_c \in X_e} \{-d_{\min}(\boldsymbol{x}_c)\} + d_{\min}(\boldsymbol{x}_c)}{\max\limits_{\boldsymbol{x}_c \in X_e} \{-d_{\min}(\boldsymbol{x}_c)\} - \min\limits_{\boldsymbol{x}_c \in X_e} \{-d_{\min}(\boldsymbol{x}_c)\}} \tag{6}$$

where w is the weight coefficient. w cycles through the set of {0.3, 0.5, 0.8, 0.95} for each sequentially generated expensive point to help the sampling process maintain the balance of local exploitation and global exploration [25].

The proposed discriminative coordinate perturbation strategy is customized from the native DYCORS algorithm. Figure 1 compares the flowcharts of coordinate perturbation processes (enclosed in dashed lines) in DYCORS and MPS-DCP. Different from DYCORS, the coordinate perturbation probabilities are discriminated by the estimated global sensitivities, i.e., the probabilities of perturbing globally sensitive variables are increased when local exploitation is expected to be enhanced, and vice versa. The discriminative coordinate perturbation in MPS-DCP helps the proposed MPS-DCP to balance the global exploration and local exploitation, as detailed in Sec. 3.2.

2.3 Discussions. As pointed out in Refs. [22,24], MPS becomes time-consuming when tackling high-dimensional problems. The reasons why MPS is not suitable for HEB problems are explained as follows. The basic theory of the contour-based discriminative sampling process assumes that the RBF metamodel can globally approximate the trend of the expensive objective to guide the sampling toward optimum. However, due to “the curse of dimensionality,” it is difficult for the metamodels to achieve good global accuracy with a limited computational budget for HEB problems. Therefore, MPS and its vairiants are generally inefficient when tackling high-dimensional problems. Considering the merits of discriminative coordinate perturbation techniques in exploring the high-dimensional design space, it is promising to incorporate DYCORS with MPS to address the challenges in solving HEB problems. In this paper, a novel MPS variant using discriminative coordinate perturbation is proposed by using concepts of DYCORS, as detailed in Sec. 3.

3 Methodology of Mode-Pursuing Sampling Using Discriminative Coordinate Perturbation

In this section, the overall procedure of the proposed MPS-DCP is presented first. Then, the approach to the sensitivity analysis based DCP is detailed to describe the perturbed coordinates selection process and the improvement in the optimization efficiency via the perturbation operation.

3.1 Overall Procedure. To improve the performance of the algorithm, the design space is normalized to [0, 1]. The overall flowchart of the proposed MPS-DCP is illustrated in Fig. 2. The procedure of MPS-DCP is introduced as follows.

Step 1: Configuration.

The initial parameters of MPS-DCP are configured, including the number of initial expensive points n_0, the number of sequential expensive points n_s generated at each iteration, the number of cheap points n_c generated at each iteration, and the maximum number of expensive function evaluations $N_{\max}$. The default settings of n_0, n_s, and n_c are shown in Table 1.

Step 2: Initial sampling.

Initial sampling is performed using the Latin hypercube design with the maximin criterion (Maximin-LHD). The expensive objective is evaluated on the initial points.

Step 3: RBF metamodel construction.

The RBF metamodel is constructed or refitted with a number of promising expensive points whose objective values are the lowest. In this work, the number of promising expensive points for fitting RBF metamodel is set to be $\min\{n_e, 10 \cdot n_v\}$. A two-dimensional identification process of $10 \cdot n_v$ expensive points is depicted in Fig. 3(*a*), where pentagram, filled circles, and hollow circles represent the current best point, the promising expensive points, and the other expensive points, respectively. In this way, the approximation accuracy of the RBF metamodel in the vicinity of the current best point can be enhanced to improve the convergence performance of the metamodel-based optimization.

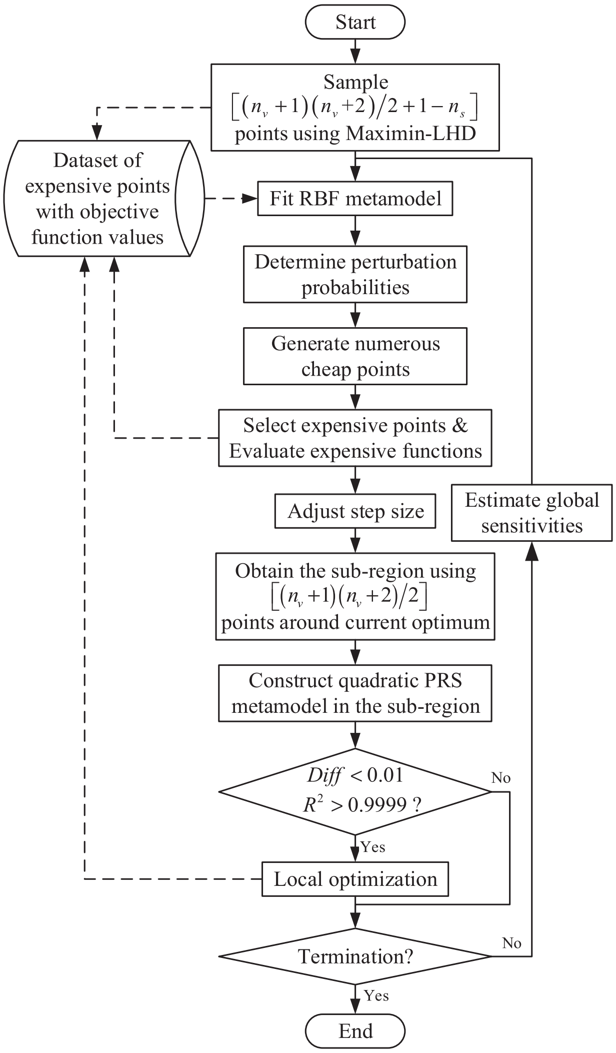

Fig. 2 Flowchart of MPS-DCP

Table 1 Default settings of several algorithm parameters

Parameter	n_0	n_s	n_c
Default value	$(n_v+1)(n_v+2)/2+1-n_s$	round$(n_v/3)$	$\min\{100 \cdot n_v, 5000\}$

Step 4: Sensitivity analysis based DCP.

n_s sequential expensive points are generated using the sensitivity analysis based discriminative coordinate perturbation method at each iteration to explore the design space. For the first iteration, the initial step size σ_0 is set to be 0.2. Since the quadratic PRS metamodel has not been constructed at the first iteration, the sensitivity indicator for guiding discriminative coordinate perturbation is not available at the beginning of optimization. Thus, the perturbation probabilities of different design variables are identical, as calculated by Eq. (5) [25]. As the optimization proceeds, the perturbation probabilities are influenced by the calculated sensitivity indicator vector $\boldsymbol{s}$. Tolerances T_{improve}, $T_{\text{stall,1}}$, and $T_{\text{stall,2}}$ are adopted for deciding when to adjust the step size σ_k. The methods to determine the sensitivity indicators, the perturbation probabilities, and the step size are detailed in Sec. 3.2.

Step 5: Quadratic PRS metamodel construction.

An interpolation quadratic PRS metamodel is constructed with Eq. (2) using $n_k=(n_v+1)(n_v+2)/2+2$ expensive points with minimum Euclidean distance to the current best point $\boldsymbol{x}_e^*$. As illustrated in Fig. 3(*b*), the pentagram stands for $\boldsymbol{x}_e^*$, and the filled circles stand for the selected n_k points. If the R^2 value of the quadratic PRS metamodel is larger than 0.9, $n_v/6$ expensive points are generated using the Maximin-LHD method in the local exploitation region defined by the envelope of the n_k points. All the existing expensive points are used to refine the quadratic PRS metamodel, whose approximation accuracy is examined by evaluating the R^2 and *Diff* values. If $Diff<0.01$ and $R^2>0.9999$, the local exploitation area is regarded to be unimodal [14]; thus, the optimization process turns to *Step 6*. Otherwise, go to *Step 7*.

Step 6: Local search.

A local search is performed on the PRS metamodel using the sequential quadratic programming (SQP) method. The current best point $\boldsymbol{x}_e^*$ is set as the starting point of SQP. The optimum found by SQP is evaluated by the objective function and added to the expensive point set.

Step 7: Termination criteria checking

MPS-DCP adopts two different termination criteria, i.e., the maximum number of function evaluations $N_{\max}$ criterion and the maximum number of stalling iterations $C_{\text{stall}}^{\max}$ criterion. If $N_{\max}$ or $C_{\text{stall}}^{\max}$ is reached, the optimization terminates and the current best point is regarded as the optimized solution. Otherwise, the process turns to *Step 3*.

3.2 Sensitivity Analysis Based Discriminative Coordinate Perturbation.

In the proposed MPS-DCP, sequential expensive points are generated by using the sensitivity analysis based DCP method. The process of the proposed DCP method is summarized in Algorithm 1. The DCP method consists of four parts, i.e., determination of perturbation probabilities, cheap points generation, sequential expensive points selection, and step size adjustment.

Algorithm 1 DCP method

Input: expensive points set $\boldsymbol{X}_e$, current best point $\boldsymbol{x}_e^*$, current best objective y_e^*, number of initial expensive points n_0, number of existing expensive points n_e, maximum number of expensive function evaluations $N_{\max}$, number of iterations n_{iter}, number of cheap points n_c, coefficients of the quadratic PRS metamodel β, initial step size σ_0, minimum step size $\sigma_{\min}$, current step size σ_k, cheap points coincident tolerance T_{coincide}, step size tolerances T_{improve}, $T_{\text{stall,1}}$, and $T_{\text{stall,2}}$, number of improving iterations C_{improve}, and number of stalling iterations C_{stall}.

Output: expensive points sets $[\boldsymbol{X}_e,\ \boldsymbol{Y}_e]$, current best point $\boldsymbol{x}_e^*$, current best objective y_e^*, number of improving iterations C_{improve}, number of stalling iterations C_{stall}, step size in the next iteration σ_{k+1}.

```
Begin
If $n_{\text{iter}} = 1$ then
    $\boldsymbol{p} \leftarrow$ CalculatePerturbationProbability($n_0,\ n_e,\ N_{\max}$);
else
    $\boldsymbol{s} \leftarrow$ CalculateSensitivityIndicators($\beta$);
    $\boldsymbol{p} \leftarrow$ CalculatePerturbationProbability($\boldsymbol{s},\ n_0,\ n_e,\ N_{\max},\ n_{\text{iter}},\ C_{\text{stall}}$);
end
$\boldsymbol{X}_c \leftarrow$ GenerateCheapPoints($\boldsymbol{p},\ \boldsymbol{X}_e,\ n_c,\ \sigma_k$);
$\boldsymbol{X}_{\text{new}} \leftarrow$ SelectSequentialPoints($\boldsymbol{X}_c,\ \boldsymbol{X}_e,\ T_{\text{coincide}}$);
$\boldsymbol{Y}_{\text{new}} \leftarrow$ EvaluateFunction($\boldsymbol{X}_{\text{new}}$)
$\boldsymbol{X}_e \leftarrow \boldsymbol{X}_e \cup \boldsymbol{X}_{\text{new}}$; $\boldsymbol{Y}_e \leftarrow \boldsymbol{Y}_e \cup \boldsymbol{Y}_{\text{new}}$
If Exist($\boldsymbol{Y}_{\text{new}} < \boldsymbol{y}_e^*$) then
    $C_{\text{improve}} = C_{\text{improve}} + 1$; $C_{\text{stall}} = 0$;
    $\boldsymbol{x}_e^* \leftarrow$ ArgumentOfTheMinimum($\boldsymbol{Y}_{\text{new}}$); $y_e^* =$ Minimum($\boldsymbol{Y}_{\text{new}}$);
else
    $C_{\text{improve}} = 0$; $C_{\text{stall}} = C_{\text{stall}} + 1$;
end
$\sigma_{k+1} \leftarrow$ AdjustStepSize($\sigma_k,\ \sigma_0,\ \sigma_{\min},\ C_{\text{improve}},\ C_{\text{stall}},\ T_{\text{improve}},\ T_{\text{stall1}},\ T_{\text{stall2}}$);
return [$\boldsymbol{X}_e,\ \boldsymbol{Y}_e,\ \boldsymbol{x}_e^*,\ \boldsymbol{y}_e^*,\ C_{\text{improve}},\ C_{\text{stall}},\ \sigma_{k+1}$]
```

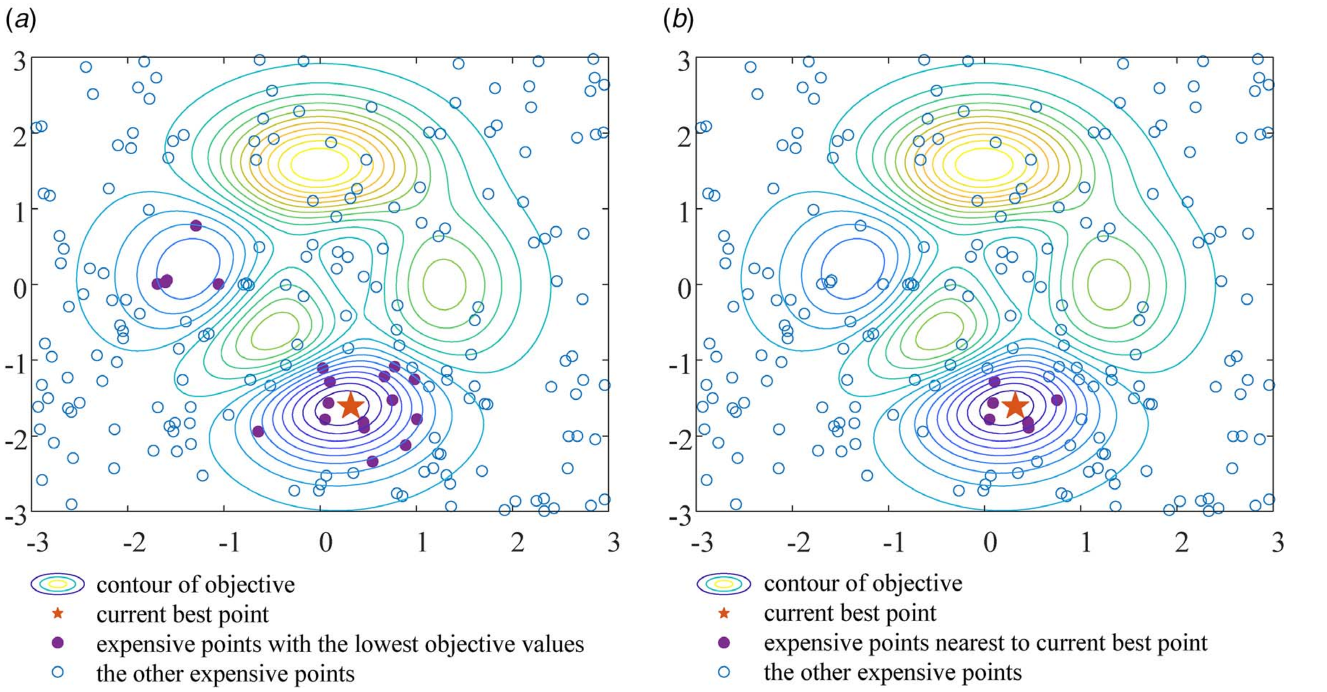

Fig. 3 Schematic diagram of identifying specific expensive points using different criteria: (*a*) identifying 10•n_v expensive points with the lowest objective values and (*b*) identifying n_k expensive points nearest to current best point

3.2.1 Determination of Perturbation Probabilities. In the original DYCORS method, the perturbation probability is the same for each variable. Since the importance of different variables is different, it is necessary to identify the critical variables in terms of the perturbation probability. As demonstrated in Ref. [26], increasing the perturbation probabilities of sensitive variables is beneficial to local exploitation; otherwise, the global exploration ability can be enhanced. From the aforementioned discussions, the perturbation probabilities are determined based on the global sensitivities and objective improvement at the current iteration. The global sensitivities are approximated by averaging the absolute value of coefficients of the quadratic PRS metamodel corresponding to each variable. The process of determining the perturbation probabilities is detailed as below.

Step 1: For the first iteration, the perturbation probabilities for coordinates are calculated by $\varphi(n_e)$ [25] as formulated in Eq. (5) and the process terminates; otherwise, calculate the sensitivity indicators of the design variables, i.e., the global sensitivities represented by the coefficients of the quadratic PRS model. The indicator of global sensitivity of the *i*th design variable$x^{(i)}$ is expressed as

$$s^{(i)} = \frac{\left|\beta^{(i)} + \beta^{(ii)} + \sum_{j\neq i}^{n_v} \beta^{(ij)}\right|}{n_v + 1}, \quad i = 1, \ldots, n_v \tag{7}$$

where $s^{(i)}$ is the sensitivity indicator of $x^{(i)}$ and $\beta^{(i)}$, $\beta^{(ii)}$, and $\beta^{(ij)}$ are the associated quadratic PRS metamodel coefficients for $x^{(i)}$ as shown in Eq. (2). The sensitivity indicators are expressed as $\boldsymbol{s} = \{s^{(i)}\},\ i = 1, 2, \ldots, n_v$.

Step 2: Modify the sensitivity indicators according to the number of stalling iterations. Once the optimization improves, it indicates that the sensitivity indicators of variables with larger global sensitivities should be shrunk to improve the global exploration capability. In this condition, each indicator $s^{(i)}$ is modified to be its reciprocal, as shown in Eq. (8). Note that if the optimization stalls for consecutive two or more iterations, it is indicated that the local exploitation capability of the algorithm should be improved. In this condition, the modified indicator $s'(i)$ equals to $s^{(i)}$, as shown in Eq. (8):

$$s'^{(i)} = \begin{cases} \dfrac{1}{s^{(i)}}, & \text{if optimization improves} \\ s^{(i)}, & \text{if optimization stalls for more than one iteration} \end{cases}, \quad i = 1, \ldots, n_v \tag{8}$$

Step 3: Calculate the perturbation probability of each variable. The perturbation probability $p^{(i)}$ for the *i*th variable is calculated by Eq. (9):

$$p^{(i)} = \frac{s'^{(i)} - \min\{s'^{(i)}\}}{\max\{s'^{(i)}\} - \min\{s'^{(i)}\}} \cdot \varphi(n_e), \quad i = 1, \ldots, n_v \tag{9}$$

3.2.2 Cheap Points Generation. Numerous cheap points are generated at each iteration by perturbing some of the variables of the current best point $\boldsymbol{x}_e^*$. A random coefficient $r^{(i)}$ obeying [0, 1] uniform distribution is generated to determine whether $x^{(i)}$ would be disturbed as shown in Eq. (10):

$$x_c^{(i)} = \begin{cases} x_e^{*(i)} + z, & \text{if } r^{(i)} < p^{(i)} \\ x_e^{*(i)}, & \text{if } r^{(i)} \geq p^{(i)} \end{cases} \tag{10}$$

where $z \sim \mathcal{N}(0, \sigma_k)$ follows the Gaussian distribution. The step size is updated iteratively to control the distribution range of cheap points in the design space. The detailed cheap points generation procedure is described in Ref. [25]. When expensive points are overcrowded in the promising region, the coefficient matrices of RBF are likely to be ill-conditioned, which may cause the numerical difficulties or even the failure for RBF construction. To address this issue, the cheap points whose minimum distances from other expensive points, notated as $d_{\min}(\cdot)$, are less than the predefined tolerance T_{coincide}, should be eliminated.

3.2.3 Sequential Expensive Points Selection. Two criteria are used to select expensive points from the numerous cheap points, i.e., the RBF criterion and the distance criterion. Cheap points are sorted in ascending order with respect to the weighted sum of its normalized RBF prediction and the opposite of its minimum distance from existing expensive points, $c(\boldsymbol{x})$, as shown in Eq. (6) [25]. The first n_s cheap points with lowest $c(\boldsymbol{x}_c)$ values are selected as the sequential expensive points for RBF refinement. Responses of the sequential expensive points are obtained by evaluating the expensive function. The RBF metamodel is constructed in *Step 3* of Sec. 3.1. A detailed procedure can be found in Ref. [25].

3.2.4 Step Size Adjustment. The process of step size adjustment is exhibited in Algorithm 2 and described as follows. When the number of improved optimization iterations exceeds T_{improve}, the step size σ_k is doubled to enhance global exploration. When optimization stalls, if $0 < C_{\text{stall}} \le T_{\text{stall},1}$, the step size σ_k is reduced by half to narrow the spread area of cheap points, so that the local exploitation is facilitated. If $T_{\text{stall},1} < C_{\text{stall}} \le T_{\text{stall},2}$, σ_k is enlarged by two times to avoid being trapped in a local optimum. If $C_{\text{stall}} > T_{\text{stall},2}$, which means optimization fails to find a sub-region better than the current by exploring a wider area, then σ_k is halved iteratively to improve the efficiency of exploiting the local area. Moreover, the minimum step size $\sigma_{\min}$ is set to be $10 \cdot T_{\text{coincide}} \cdot \sqrt{n_v}$ to avoid overcrowding. In this work, tolerances T_{improve}, $T_{\text{stall},1}$, and $T_{\text{stall},2}$ are set as 2, 2, and 6, respectively.

Algorithm 2 Step size adjustment method

Input: initial step size σ_0, current step size σ_k, minimum step size $\sigma_{\min}$, cheap points coincident tolerance T_{coincide}, step size tolerances T_{improve}, $T_{\text{stall},1}$, and $T_{\text{stall},2}$, number of optimization improving iterations C_{improve}, and number of optimization stalling iterations C_{stall}.
Output: step size in the next iteration σ_{k+1}, number of improving iterations C_{improve}, and number of stalling iterations C_{stall}.

```
Begin
    If Exist($\boldsymbol{Y}_{\text{new}} < y_e^*$) then
        $C_{\text{improve}} = C_{\text{improve}} + 1$; $C_{\text{stall}} = 0$;
        if $C_{\text{improve}} \ge T_{\text{improve}}$ then
            $\sigma_{k+1} \leftarrow \min(2\sigma_k,\ \sigma_0)$;
            $C_{\text{improve}} \leftarrow 0$; $C_{\text{stall}} \leftarrow 0$;
        end
    else
        $C_{\text{improve}} = 0$; $C_{\text{stall}} = C_{\text{stall}} + 1$;
        If $C_{\text{stall}} \le T_{\text{stall},1}$ then
            $\sigma_{k+1} \leftarrow \max(\sigma_k/2,\ \sigma_{\min})$;
        elseif $T_{\text{stall},1} < C_{\text{stall}} \le T_{\text{stall},2}$ then
            $\sigma_{k+1} \leftarrow \min(2\sigma_k,\ \sigma_0)$;
        else
            $\sigma_{k+1} \leftarrow \max(\sigma_k/2,\ \sigma_{\min})$;
        end
    end
    return [$\sigma_{k+1},\ C_{\text{improve}},\ C_{\text{stall}}$]
```

3.2.5 Convergence Analysis. In this section, the convergence property of MPS-DCP is theoretically analyzed and numerically demonstrated, which contributes to the mathematical foundation of the proposed method.

Proof of convergence: ■

DEFINITION 8 (the best sequence of a stochastic algorithm converges with probability 1) [27]. *Let y_n^* be the objective value of the best point in the first n iterations. The best sequence of a stochastic algorithm is defined to converge with probability 1, if $P\left(\lim_{n\to\infty} y_n^* = y^{**}\right) = 1$, where y^{**} is a real number.*

THEOREM 1 [27]. *The best sequence of a stochastic algorithm converges with probability 1, if the algorithm satisfies the following two conditions.*

(1) *The algorithm adopts elite reserve strategy, that is*

$$y_{k+1}^* = \begin{cases} f(\boldsymbol{x}_{k+1}), & f(\boldsymbol{x}_{k+1}) < y_k^* \\ y_k^*, & \text{otherwise} \end{cases} \tag{11}$$

where y_k^* is the current best objective at the kth iteration, which is stored in the sample database, and $\boldsymbol{x}_{k+1}$ is the best point among all the newly added points at the kth iteration.

(2) *Let $\boldsymbol{x}'$ be any point other than the global optimal point, the probability of transforming from $\boldsymbol{x}'$ to its corresponding level set, $L(\boldsymbol{x}') = \{\boldsymbol{x} | f(\boldsymbol{x}) < f(\boldsymbol{x}'),\ \boldsymbol{x} \in S\}$, is not equal to zero. S represents the entire design space.*

The convergence of MPS-DCP is proved using Theorem 1, which leads to Theorem 2.

THEOREM 2. *The best sequence of MPS-DCP converges with probability 1.*

Proof. The current best point $\boldsymbol{x}_e^*$ at each iteration are added into existing expensive points set $\boldsymbol{X}_e$, and it is reserved during the optimization procedure. Thus, MPS-DCP adopts the elite reserve strategy. During MPS-DCP procedure, at the k-th iteration, massive cheap points are generated by adding normally distributed perturbation on some variables of current best point $\boldsymbol{x}_{e,k}^*$. The probability of generating an arbitrary cheap point $\boldsymbol{x}_c$, which is denoted by $p_{\text{cheap}}(\boldsymbol{x}_c)$, is determined by perturbation probabilities of variables $\boldsymbol{p}_{\text{var}}$ and probabilities of cheap points coordinates $\boldsymbol{p}_{\text{cord}}(\boldsymbol{x}_c)$. $p_{\text{cheap}}(\boldsymbol{x}_c)$ is expressed as Eq. (12):

$$p_{\text{cheap}}(\boldsymbol{x}_c) = \prod_{i=1}^{n_v} \left[p_{\text{var}}^{(i)} \cdot p_{\text{cord}}^{(i)}(x_c^{(i)}) \right] \tag{12}$$

The perturbation probability of the i-th variable $p_{\text{var}}^{(i)}$, is calculated by Eq. (9), which is positive for $i = 1, 2, \ldots, n_v$.

Note that the design space is normalized to $[0,\ 1]^{n_v}$, so that if $x_c^{(i)}$ exceeds [0, 1], it is reflected into the design space. As a result, the probability of generating the ith coordinate of the cheap point is calculated by

$$p_{\text{cord}}^{(i)}(x_c^{(i)}) = \sum_{j=0}^{\infty} \left[f_{\text{Gaussian}}(2j + x_c^{(i)};\ x_e^{*(i)},\ \sigma_k) \right. \\ \left. + f_{\text{Gaussian}}(2j + 2 - x_c^{(i)};\ x_e^{*(i)},\ \sigma_k) \right] \tag{13}$$

where $f_{\text{Gaussian}}(\cdot)$ is the probability density function of the Gaussian distribution. $p_{\text{cord}}^{(i)}(x_c^{(i)})$ is positive for $x_c^{(i)} \in [0,\ 1]$.

Based on Eqs. (12) and (13), $p_{\text{cheap}}(\boldsymbol{x}_c)$ satisfies

$$p_{\text{cheap}}(\boldsymbol{x}_c) > 0, \quad \forall \boldsymbol{x}_c \in [0,\ 1]^{n_v} \tag{14}$$

Expensive points are selected from cheap points according to the RBF criterion and distance criterion. As shown in Eq. (6), cheap points with lowest $c(\boldsymbol{x}_c)$ are selected. The selection probability of $\boldsymbol{x}_c$ is denoted as $p_{\text{select}}(\boldsymbol{x}_c)$.

On the basis of aforementioned discussion, the probability of generating an arbitrary expensive point is expressed as Eq. (15):

$$p(\boldsymbol{x}) = p_{\text{cheap}}(\boldsymbol{x}) \cdot p_{\text{select}}(\boldsymbol{x}) \tag{15}$$

Let $\boldsymbol{x}'$ be any point other than the global optimal point, and its corresponding level set is $L(\boldsymbol{x}') = \{\boldsymbol{x} | f(\boldsymbol{x}) < f(\boldsymbol{x}'),\ \boldsymbol{x} \in S\}$. Since the RBF metamodel has the ability of approximating the expensive objective function, the probability of points in $L(\boldsymbol{x}')$ to be selected from cheap points is positive, i.e., $p_{\text{select}}(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \in L(\boldsymbol{x}')$. Thus, $p(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \in L(\boldsymbol{x}')$.

In other words, the probability of transforming from $\boldsymbol{x}'$ to its level set $L(\boldsymbol{x}')$ is not equal to zero. According to Theorem 1, the best sequence of MPS-DCP converges with probability 1. ■

3.2.6 Convergence Rate Analysis. Via comparing with SQP, the convergence rate of MPS-DCP is investigated on PUR-T1-13 problems (notated as PUR problems) [37] as displayed in Appendix. Note that SQP is proved to converge superlinearly under reasonable assumptions near a strongly stationary point [28]. Convergence curves of SQP and MPS-DCP on PUR problems for ten runs are illustrated in Fig. 4.

Figure 4(*a*) shows that the converge rate of SQP and MPS-DCP are similar on PUR10 problem. Figures 4(*b*) and 4(*c*) indicate that for PUR20 and PUR30 problems, MPS-DCP converges slower than SQP at the early stage of optimization. As the optimization proceeds, MPS-DCP gradually converges faster than SQP. Thus, MPS-DCP is expected to converge superlinearly via the comparison results with SQP.

3.2.7 Constraint Handling Strategy. The fundamental MPS-DCP mainly aims at solving HEBs with bounds on the variables, namely, unconstrained problems. Since real-world engineering design optimization problems generally involve expensive constraints, MPS-DCP customizes a constraint handling strategy to accomplish constrained optimization, which is described as follows.

(1) At the initial sampling stage (*Step 2* in Sec. 3.1), if initial expensive points are all infeasible, a sub-optimization to improve sample point feasibility is executed until at least one feasible point is obtained or the computational budget is exhausted.

(2) During the metamodeling process, the RBF metamodels for constraints are also constructed or updated using the $\min\{n_e, 10 \cdot n_v\}$ expensive points with lowest constraints responses. And the fitted RBF metamodels are employed to replace the original expensive black-box constraints for global optimization in *Step 3* and local search in *Step 6* in Sec. 3.1.

(3) In the DCP strategy, cheap points with nonpositive predicted responses of RBF constraints are regarded as potential feasible cheap points, and expensive points are selected from potential feasible cheap points followed by the same mechanism in *Step 4* in Sec. 3.1.

3.3 Illustration of MPS-DCP on Two-Dimensional Problem. To intuitively illustrate the optimization process of MPS-DCP, a 2D Peaks problem (PK) shown in Eq. (16) is investigated in this section. In this study, n_s and $N_{\max}$ are set to be 3 and 20, respectively. The optimization process of MPS-DCP is graphically depicted in Fig. 5

$$\begin{aligned} \min \quad f(\boldsymbol{x}) = & \, 3(1 - x^{(1)})^2 \exp\left[-x^{(1)2} - (x^{(2)} + 1)^2\right] \\ & -10\left(\frac{x^{(1)}}{5} - x^{(1)3} - x^{(2)5}\right) \exp\left[-x^{(1)2} - x^{(2)2}\right] \\ & -\frac{1}{3}\exp\left[-(x^{(1)} + 1)^2 - x^{(2)2}\right] \\ & \boldsymbol{x} \in [-3, 3]^2 \end{aligned} \tag{16}$$

As depicted in Fig. 5(*a*), five expensive points are generated during initial sampling to construct the RBF metamodel of the objective function. At the first iteration, the perturbation step size is initialized

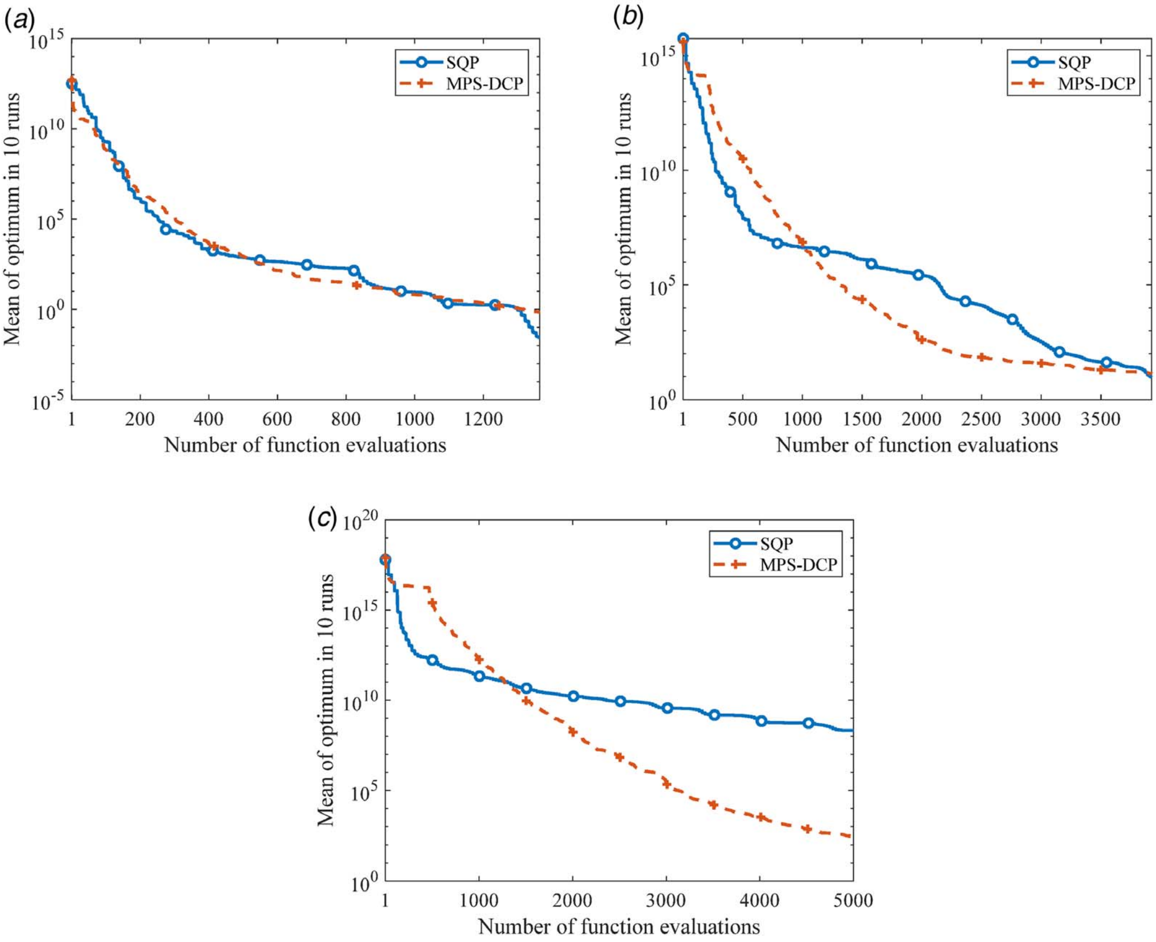

Fig. 4 Convergence curves of SQP and MPS-DCP for PUR problems: (*a*) PUR10 problem, (*b*) PUR20 problem, and (c) PUR30 problem

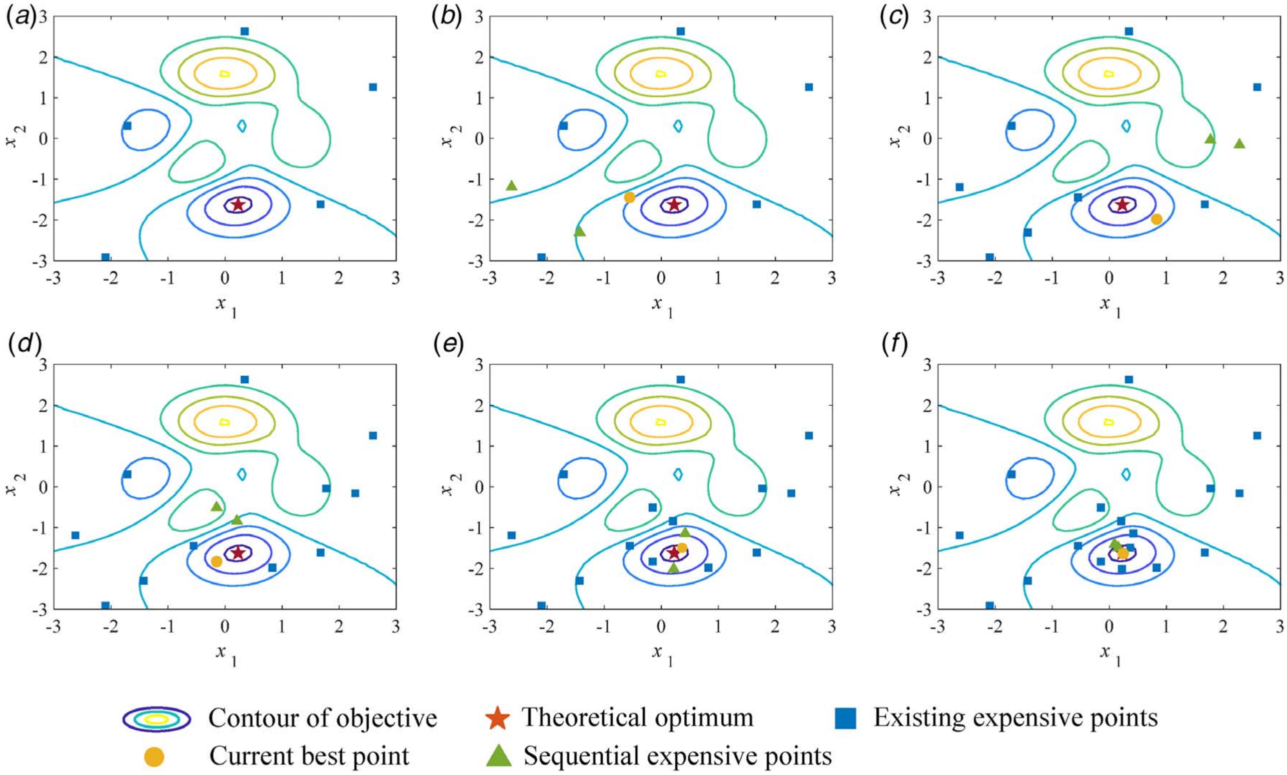

Fig. 5 Optimization process of MPS-DCP for the PK problem: (*a*) initial sampling, (*b*) after the first iteration, (*c*) after the second iteration, (*d*) after the third iteration, (*e*) after the fourth iteration, and (*f*) after the fifth iteration

as 0.2 to explore the design space, and three expensive points are generated by coordinate perturbation, as shown in Fig. 5(*b*). Figures 5(*c*)–5(*e*) indicate that as the optimization process proceeds, the sequential expensive points are generated toward the global optimum. After the fifth iteration, the maximum number of function evaluations is reached, and the global optimum is successfully reached as shown in Fig. 5(*f*). The optimization process for the PK problem illustrates the effectiveness of MPS-DCP.

4 Tests on Numerical Benchmark Problems

In this section, the performance of the proposed MPS-DCP method is compared with several MBDO algorithms for HEB problems, including MPS, DYCORS, TRMPS, and OMID [24]. Sixteen high-dimensional numerical benchmarks with the dimensionality from 10 to 30 are adopted for the tests. Although the benchmarks are analytical, they are assumed to represent HEB problems to test the optimization performance of different methods. The optimization results are compared and discussed in detail to illustrate the merits of MPS-DCP.

4.1 Test Framework. The numerical benchmarks are summarized in Table 2, including the dimensionality (n_v), theoretical optimum, and maximum numbers of expensive function evaluations ($N_{\max}$). The mathematical formulas of the benchmarks are given in Appendix. For the convenience of comparison, the maximum number of function evaluations termination criterion is used for different numerical benchmarks in this study. Note that $N_{\max}$ is determined according to Ref. [22] for fair comparison, and $C_{\text{stall}}^{\max}$ is set to be empty in MPS-DCP.

For all the algorithms, ten runs are carried out to alleviate random variation on simulation results. DYCORS is implemented using codes presented by Mueller [29]. The number of sequential expensive points is set to be round(n_v/3) in DYCORS, which is the same as that of MPS-DCP. Optimization results of MPS, TRMPS, and OMID are directly cited from Ref. [24]. The mean value and standard deviation of the optimized objective value for each benchmark problem are recorded to show the convergence and robustness performances of the five algorithms.

4.2 Comparison of Optimization Results. The optimization results of MPS-DCP compared with MPS, DYCORS, TRMPS,

Table 2 Information of numerical benchmark problems

Problem	F16	R10	R20	R30	SUR10	SUR20	SUR30	PUR10
n_v	16	10	20	30	10	20	30	10
Theoretical optimum	25.875	0	0	0	0	0	0	0
$N_{\max}$	700	3828	5000	5000	5000	5000	5000	4153

Problem	PUR20	PUR30	GR10	GR20	GR30	ZF10	ZF20	ZF30
n_v	20	30	10	20	30	10	20	30
Theoretical optimum	0	0	0	0	0	0	0	0
$N_{\max}$	5000	5000	2352	5000	5000	3532	5000	5000

Table 3 Mean and standard deviation of the best objective values obtained by MPS, DYCORS, TRMPS, OMID, and MPS-DCP

	MPS		DYCORS		TRMPS		OMID		MPS-DCP	
Problem	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F16	29.07	28.9	25.8754	1.0283×10^{-4}	25.928	0.148	25.8852	0.0064	**25.8750**	**1.0550×10^{-6}**
R10	229.5	177.8	6.6509	0.3190	5.548	1.99	19.7823	29.4380	**4.2172**	**0.7094**
R20	3.315×10^{4}	3.49×10^{4}	17.5883	0.8006	17.98	1.41	41.2	29.7582	**14.5436**	**0.2993**
R30	9.45×10^{4}	9.65×10^{5}	42.9148	19.4130	**21.53**	7.54	117	59.4922	26.4711	**1.0311**
SUR10	20.11	15.1	0.9961	0.0944	1.086	0.201	1.2207	0.2734	**0.9547**	**0.0059**
SUR20	6765	6476	2.1689	1.9171	2.192	0.848	2.41	1.4306	**1.4032**	**0.4828**
SUR30	3.42×10^{4}	3.38×10^{4}	88.0583	140.7335	3.536	1.536	54.8	106.0589	**2.0394**	**1.2022**
PUR10	4.237×10^{7}	6.675×10^{5}	0.0334	0.0358	9.84	26.2	**0**	**0**	**3.7679×10^{-12}**	**5.5796×10^{-12}**
PUR20	2.182×10^{13}	1.572×10^{13}	4.0997×10^{3}	6.3766×10^{3}	8.399×10^{5}	$1.45\text{e}\times10^{6}$	0.805	1.6064	**0.0426**	**0.0841**
PUR30	7.18×10^{15}	6.345×10^{15}	3.6909×10^{8}	2.6788×10^{8}	3.189×10^{8}	3.55×10^{8}	657	1184.42	**286.7752**	**435.3574**
GR10	0.184	0.042	0.5336	0.0973	0.133	0.0901	N/A	N/A	**0.0342**	**0.0203**
GR20	205.9	209.44	0.7883	0.1013	0.138	0.0370	N/A	N/A	**0.0214**	**0.0134**
GR30	396.6	405.2	0.8802	0.1756	0.252	0.0506	N/A	N/A	**0.0194**	**0.0103**
ZF10	2.73×10^{6}	1.65×10^{6}	0.5166	0.3130	0.371	0.764	0.203	0.1661	**1.3802×10^{-5}**	**7.4179×10^{-6}**
ZF20	2.36×10^{11}	2.31×10^{11}	2.6858	3.4750	**0.235**	**0.189**	123	66.4936	6.4981	4.2109
ZF30	1.112×10^{13}	1.18×10^{13}	70.7065	27.8559	**31.03**	**14.71**	284	72.0840	109.8613	30.2526

Note: The best results are expressed in bold. Std represents the standard deviation of optimized objective value. Reference [24] did not provide results of OMID for GR10, GR20, and GR30.

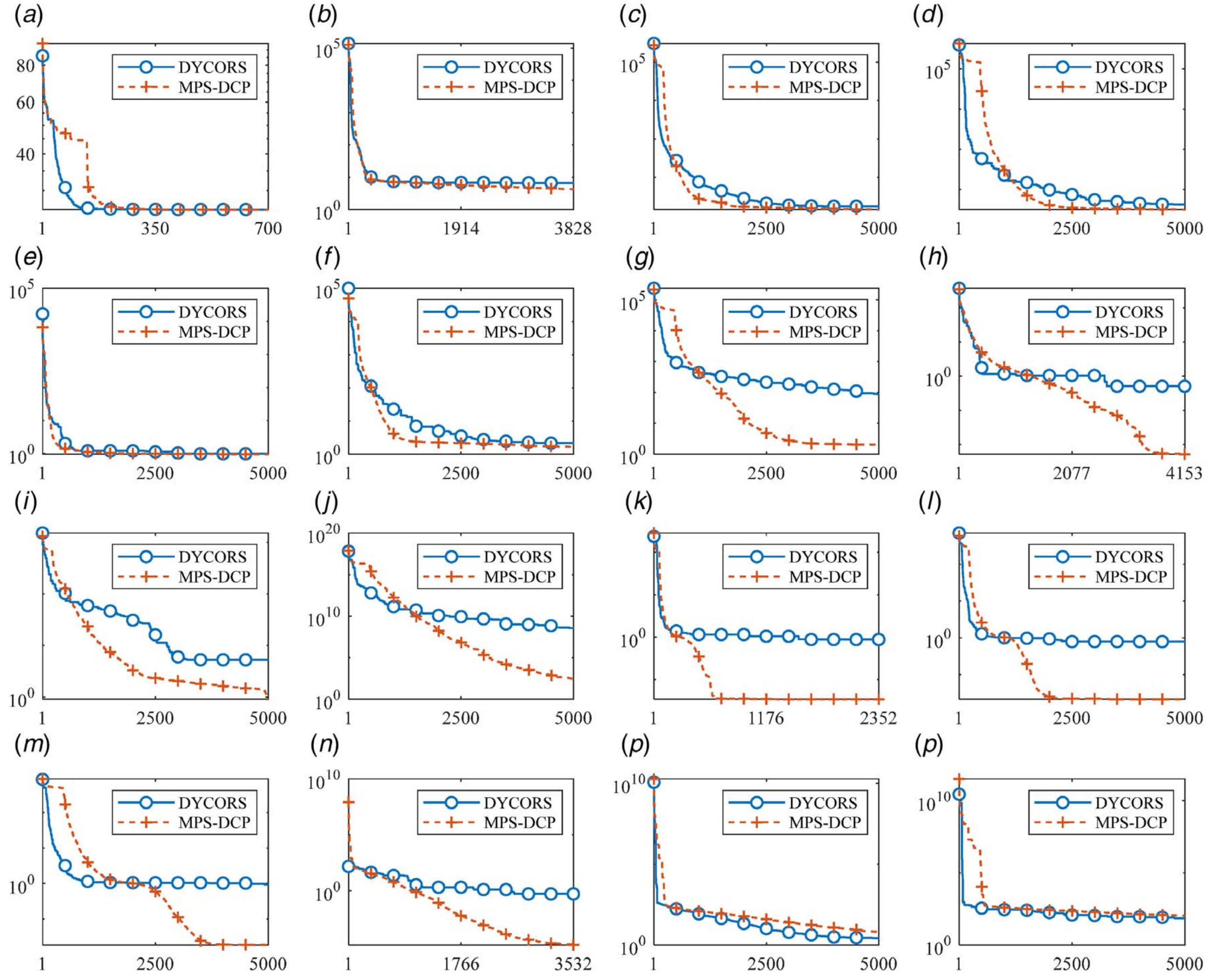

Fig. 6 Convergence curves of DYCORS and MPS-DCP: (*a*) F16, (*b*) R10, (*c*) R20, (*d*) R30, (*e*) SUR10, (*f*) SUR20, (*g*) SUR30, (*h*) PUR10, (*i*) PUR20, (*j*) PUR30, (*k*) GR10, (*l*) GR20, (*m*) GR30, (*n*) ZF10, (*o*) ZF20, and (*p*) ZF30

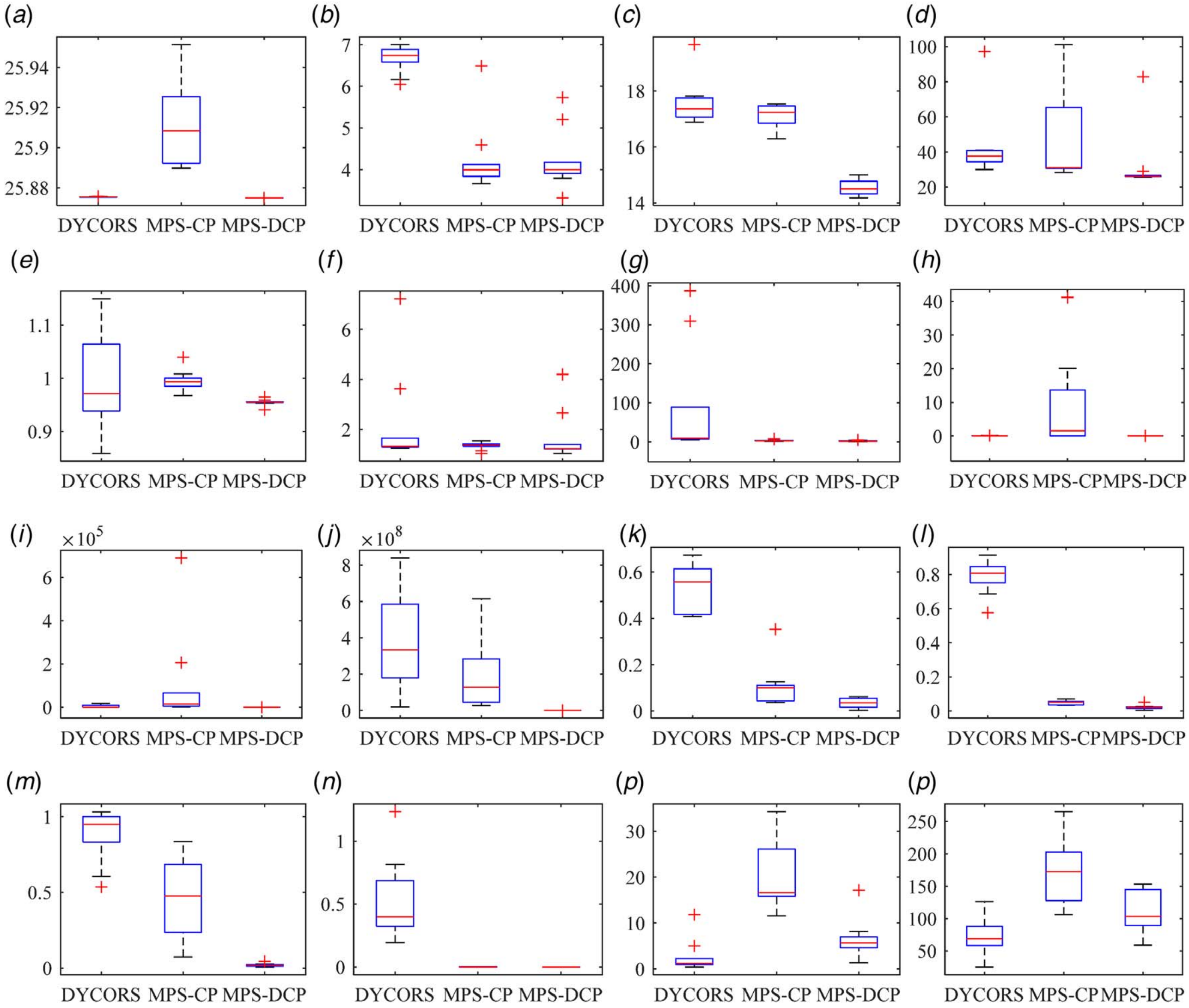

Fig. 7 Boxplots of the best objective values obtained by DYCORS, MPS-CP, and MPS-DCP: (*a*) F16, (*b*) R10, (*c*) R20, (*d*) R30, (*e*) SUR10, (*f*) SUR20, (*g*) SUR30, (*h*) PUR10, (*i*) PUR20, (*j*) PUR30, (*k*) GR10, (*l*) GR20, (*m*) GR30, (*n*) ZF10, (o) ZF20, and (*p*) ZF30

and OMID are shown in Table 3, and the convergence curves of DYCORS and MPS-DCP are illustrated in Fig. 6. In Fig. 6, the x-axis label is the number of function evaluations. Results indicate that MPS-DCP outperforms the competitive algorithms in terms of converging to the global optimum for most benchmarks except for R30, ZF20, and ZF30. In these cases, MPS-DCP still finds better global solutions than MPS and OMID. Besides, apart from the early stage of the optimization process, MPS-DCP generally converges faster than DYCORS except for F16, ZF20, and ZF30. The detailed analysis of the optimization results is presented as follows.

According to the benchmark test, MPS yields the worst results among the algorithms in terms of optimality and robustness. On the contrary, MPS-DCP generally consistently finds better solutions than DYCORS, TRMPS, and OMID for all the problems (except R30, ZF20, and ZF30) with the same computational budget. Especially for SUR30, PUR20–PUR30, and ZF10 problems, the advantage of MPS-DCP becomes more obvious compared with other algorithms since the mean optimized objective values produced by MPS-DCP are several orders of magnitudes lower. This superiority of MPS-DCP is largely due to its unique capability to capture the relatively significant improvement caused by trivial variations in variable coordinates, which is further discussed in Sec. 4.3.2. Moreover, MPS-DCP also shows the best performance in solving GR10–GR30 problems with massive widespread local optima [30], which illustrates that MPS-DCP can escape from being trapped in local optima.

In conclusion, with the same computational cost, MPS-DCP generally outperforms the competitors in terms of optimization effectiveness and robustness, which demonstrates the effectiveness of MPS-DCP for solving HEB problems.

4.3 Discussions

4.3.1 Comparison With Mode-Pursuing Sampling Using Conventional Coordinate Perturbation Strategy. In the original DYCORS using conventional CP strategy, the variable perturbation probabilities simply change with n_e during the optimization, as shown in Eq. (5). Different from CP, note that the variable perturbation probabilities in our proposed DCP technique are determined by n_e, the approximated global sensitivities of design variables, and the number of stalling iterations during the optimization simultaneously, as shown in Eqs. (7)–(9). According to Ref. [26], increasing the perturbation probabilities of sensitive variables is beneficial to local exploitation; otherwise, the global exploration ability can be enhanced. Thus, the customized perturbation coordinate probability is expected to improve the optimization performance of MPS-DCP compared with the algorithm using CP.

Table 4 Parameter setting in different cases

	Case I	Case II	Case III	Case IV	Case V	Case VI	Case VII
n_s	n_v	round(n_v/3)	round(n_v/3)	round(n_v/3)	round(n_v/3)	round(n_v/3)	round(n_v/3)
T_{coincide}	$5 \times 10^{-5} \cdot \sqrt{n_v}$	$1 \times 10^{-3} \cdot \sqrt{n_v}$	$5 \times 10^{-5} \cdot \sqrt{n_v}$	$5 \times 10^{-5} \cdot \sqrt{n_v}$	$5 \times 10^{-5} \cdot \sqrt{n_v}$	$5 \times 10^{-5} \cdot \sqrt{n_v}$	$5 \times 10^{-5} \cdot \sqrt{n_v}$
$T_{\text{Stall},1}$	2	2	max{5, n_v}	2	2	2	2
T_{improve}	2	2	2	3	2	2	2
$T_{\text{stall},2}$	6	6	6	6	4	8	6

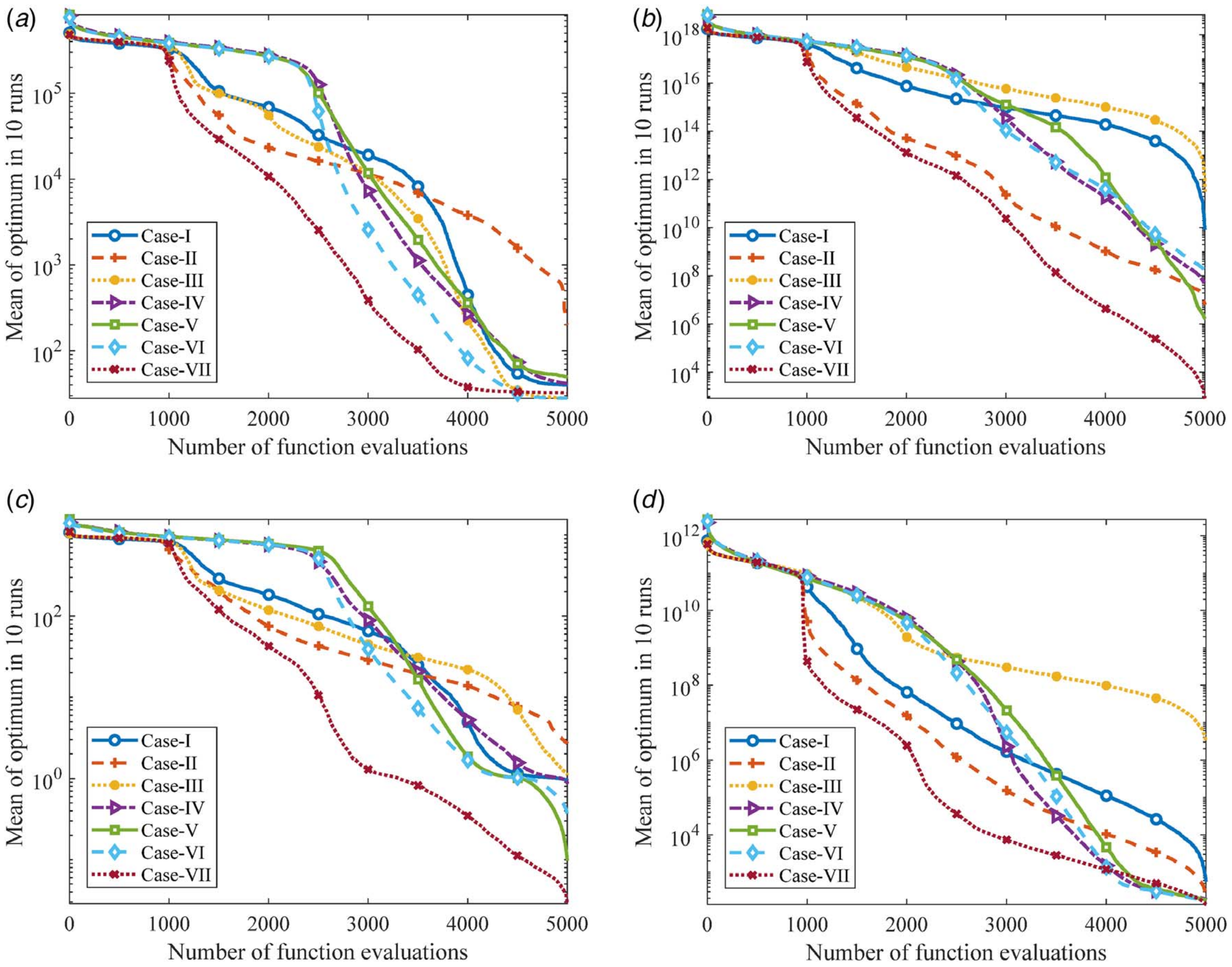

Fig. 8 Tuning parameters sensitivity analysis results: (*a*) R30 problem, (*b*) PUR30 problem, (c) GR30 problem, and (*d*) ZF30 problem

To further illustrate the effectiveness of the proposed DCP technique, the performance of a simple combination of the MPS framework and CP technique (notated as MPS-CP) is also investigated. Figure 7 shows the boxplots of the best objective values obtained by DYCORS, MPS-CP, and MPS-DCP for ten runs. It is shown that MPS-CP yields better results than DYCORS only in R10–20, SUR20–30, PUR30, GR10–30, and ZF10. By using the DCP technique, MPS-DCP significantly outperforms DYCORS and MPS-CP in terms of optimality and robustness in all the mentioned problems except SUR20, PUR10, ZF20, and ZF30. It is concluded that the proposed DCP technique helps MPS-DCP to yield better results compared with classical algorithms.

4.3.2 Effects of Tuning Parameters and Termination Criteria in MPS-DCP. In this part, effects of several critical tuning parameters (i.e., the number of sequential expensive points n_s and the tolerance parameters T_{coincide}, $T_{\text{stall},1}$, T_{improve}, and $T_{\text{stall},2}$) on the optimization results of R30, PUR30, GR30, and ZF30 are investigated at first, in order to find reasonable default settings for MPS-DCP. The parameter settings in different cases are summarized in Table 4. The parameters in Case VII are our recommended default settings of MPS-DCP. The mean objective values in ten runs (sorted in descending order) with different tuning parameters are graphically expressed in Fig. 8.

The first tuning parameter is the number of sequential expensive points n_s. The case adopting a larger n_s is referred to as Case I, where n_s equals to n_v, the same as the original MPS. As shown in Fig. 8, for the first 1000 largest objective values, the convergence performances of MPS-DCP with different n_s are almost the same. However, after that, MPS-DCP with the default settings (Case VII) significantly converges faster, especially for R30, PUR30, and GR30.

The expensive point coincidence tolerance T_{coincide} is examined. It is also modified by the value in the original DYCORS, i.e.,

$T_{\text{coincide}} = 1 \times 10^{-3} \cdot \sqrt{n_v}$, to see its effects on convergence performance. The new case is named as Case II. Figure 8 shows that although the sensitivity of T_{coincide} on algorithm performance varies for different problems, to tolerate denser cheap points in design space under the premise of not leading to the failure of RBF construction generally benefits finding promising solutions. It is because that this modification might capture the improvement caused by trivial variations in variable coordinates.

The optimization stalling tolerance $T_{\text{stall},1}$ and improving tolerance T_{improve} are also examined. Original DYCORS suggests $T_{\text{stall},1} = \max\{5,\ n_v\}$ and $T_{\text{improve}} = 3$ [25], while in MPS-DCP, $T_{\text{stall},1} = 2$ and $T_{\text{improve}} = 2$. MPS-DCP was also run with $T_{\text{stall},1} = \max\{5,\ n_v\}$ and $T_{\text{improve}} = 3$, which are referred to as Case III and Case IV, respectively. Figure 8 shows substantial deteriorations in the performance of MPS-DCP on these problems when using a larger stalling tolerance $T_{\text{stall},1}$ and a larger improving tolerance T_{improve}. This indicates that the original value of $T_{\text{stall},1}$ and T_{improve} suggested by DYCORS are rather conservative and the convergence performance could be enhanced when adopting a greedier step size reduction strategy.

The last discussed tuning parameter is the other stalling tolerance $T_{\text{stall},2}$ introduced by MPS-DCP. When the number of stalling iterations reaches $T_{\text{stall},2}$, it is indicated that the MPS-DCP fails to find better points in global exploration, so that MPS-DCP focuses on exploiting the local area. A smaller and a larger tolerance value, i.e., 4 and 8, are also tested, which are referred to as Case V and Case VI, respectively. Figure 8 illustrates that for R30, PUR30, and GR30, the usage of improper $T_{\text{stall},2}$ values slow down the pace of convergence. For the final stage of optimizing ZF30, the variants of MPS-DCP in Case V and Case VI yield slightly better results. However, the default settings of current MPS-DCP are still recommended in most cases.

In addition, the effects of $C_{\text{stall}}^{\max}$ criterion are demonstrated on GR30 problem. The termination criteria in different cases are summarized in Table 5. The convergence curves of these cases are graphically expressed in Fig. 9.

Figure 9 indicates that adopting a relatively small $C_{\text{stall}}^{\max}$ (20 for GR30 problem) leads to a premature termination, while adopting a relatively larger $C_{\text{stall}}^{\max}$ (100 for GR30 problem) yields better solutions at the expense of large computational cost. Besides, Cases XI and XII indicate that MPS-DCP generally finds better solutions as $N_{\max}$increasing.

Table 5 Termination criteria in different cases

	Case VIII	Case IX	Case X	Case XI	Case XII
$N_{\max}$	–	–	–	3000	5000
$C_{\text{stall}}^{\max}$	20	50	100	–	–

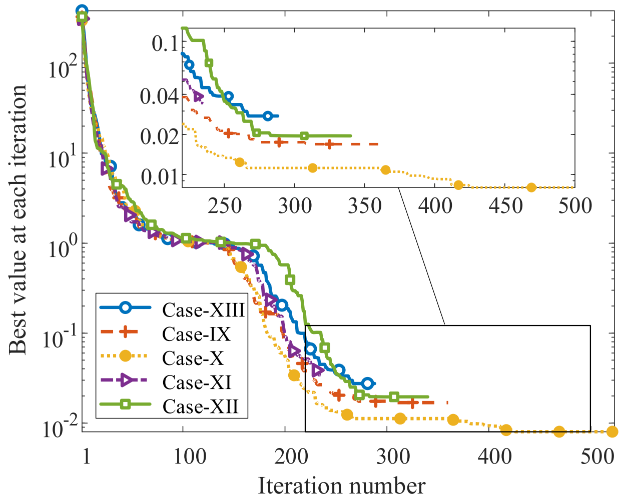

Fig. 9 Convergence curves of MPS-DCP with different termination criteria

5 Engineering Design Optimization Problem

Engineering design optimization problems with the objective and constraints arising from black-box computation-intensive simulations are challenging in practices. In literature, many state-of-the-art MBDO methods such as adaptive response surface method using intelligent space exploration strategy (ARSM-ISES) [21], constrained optimization by radial basis function interpolation (COBRA) [31], surrogate-based constrained global optimization using space reduction (SCGOSR) [32], and general sequential constraints updating approach based on the confidence intervals from the Kriging surrogate model (SCU–CI) [33] have been developed and successfully applied to practical engineering optimization. To further demonstrate the effectiveness and practicality of MPS-DCP in solving real-world engineering design optimization problems, a stepped cantilever beam design problem and an all-electric geostationary orbit (GEO) satellite (AEGS) MDO problem are investigated in this section.

5.1 Stepped Cantilever Beam Design Problem

5.1.1 Problem Description. The 30D stepped cantilever beam design problem (CP30) is referred from Ref. [34]. The stepped cantilever beam has ten sections, as shown in Fig. 10. A 50 kN load, P, is applied at the tip. Young's modulus E and the maximum allowed stress σ_{allow} in each section are 200 GPa and 350 MPa, respectively. There are three variables in each section: width $b^{(i)}$, height $h^{(i)}$, and length $l^{(i)}$. Then, the 30 design variables are presented as follows:

$$\boldsymbol{X} = [b^{(1)},\ h^{(1)},\ l^{(1)},\ b^{(2)},\ h^{(2)},\ l^{(2)},\ldots,\ b^{(10)},\ h^{(10)},\ l^{(10)}] \qquad (17)$$

The objective is to minimize the tip deflection of the beam, δ, expressed as below [34]:

$$\begin{aligned}\delta &= \int_0^{l^{(10)}} \frac{Px^{(10)2}}{EI^{(10)}}\mathrm{d}x^{(10)} + \int_0^{l^{(9)}} \frac{P(x^{(9)}+l^{(10)})^2}{EI^{(9)}}\mathrm{d}x^{(9)} + \cdots \\ &\quad + \int_0^{l^{(1)}} \frac{P(x^{(1)}+l^{(2)}+l^{(3)}+\cdots+l^{(9)})}{EI^{(1)}}\mathrm{d}x^{(1)} \\ &= \frac{P}{3E}\sum_{i=1}^{10}\left[\frac{12}{b^{(i)}h^{(i)3}}\left(\left(\sum_{j=i}^{10} l^{(j)}\right)^3 - \left(\sum_{j=i+1}^{10} l^{(j)}\right)^3\right)\right] \end{aligned} \qquad (18)$$

where $I^{(i)}$ is the rotary inertia about a neutral axis for each section. Stress concentration in the beam is not taken into account.

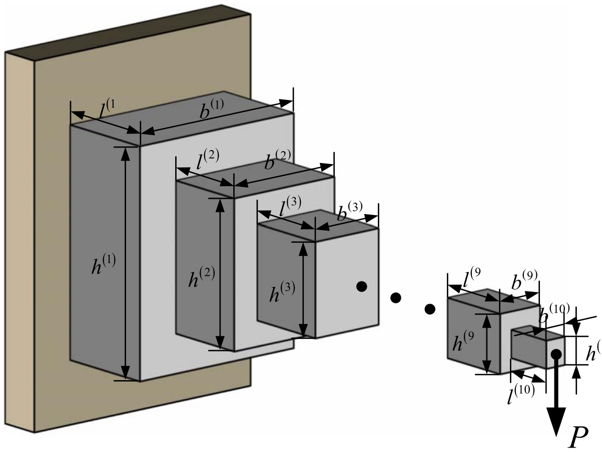

Fig. 10 Illustration of the stepped cantilever beam

Constraints consist of maximum bending stress, maximum aspect ratio, and minimum beam length.

The stepped cantilever beam design problem can be formulated as follows:

$$\text{find}\quad \boldsymbol{x} = [b^{(1)},\ h^{(1)},\ l^{(1)},\ b^{(2)},\ h^{(2)},\ l^{(2)},\ldots,\ b^{(10)},\ h^{(10)},\ l^{(10)}]$$

$$\min\quad \delta = \frac{P}{3E}\sum_{i=1}^{10}\left[\frac{12}{x^{(3i-2)}x^{(3i-1)2}}\left(\left(\sum_{j=i}^{10}x^{(3j)}\right)^3 - \left(\sum_{j=i+1}^{10}x^{(3j)}\right)^3\right)\right]$$

$$\text{s.t.}\quad \frac{6P}{x^{(3i-2)}x^{(3i-1)2}}\sum_{j=i}^{10}x^{(3j)} \le \sigma_{\text{allow}},\qquad \frac{x^{(3i-1)}}{x^{(3i-2)}} \le AR,$$

$$\sum_{j=1}^{10}x^{(3i-2)}x^{(3i-1)}x^{(3j)} \le V_{\max},\qquad -\sum_{j=1}^{10}x^{(3j)} \le -L_{\min},$$

$$0.01\,\text{m} \le b^{(i)} \le 0.05\,\text{m},\qquad 0.30\,\text{m} \le h^{(i)} \le 0.65\,\text{m},$$

$$0.50\,\text{m} \le l^{(i)} \le 1.00\,\text{m}\quad i = 1, 2, \ldots, 10 \tag{19}$$

Although the objective and constraints are explicitly formulated, they are assumed to be black-box models for illustrative and benchmarking purposes.

5.1.2 Optimization Results. To examine the merit of the proposed method, MPS-DCP is also compared with MPS and TRMPS, as shown in Table 6. The results of MPS and TRMPS are directly cited from Ref. [22]. Because DYCORS does not inherently support constraint handling, it is not involved in the comparison. To make a fair comparison, $N_{\max}$ is set as 990 for MPS-DCP, the same as the settings in Ref. [22]. And $C_{\text{stall}}^{\max}$ is set to be 20. Figure 11 plots the objective convergence history of MPS-DCP. Note that the current optima at each iteration during MPS-DCP are all feasible.

Figure 11 exhibits that MPS-DCP yields a 60.62% reduction in the feasible optimum, compared with the best point from initial sampling, 0.0387. As shown in Table 6, in the 30-dimensional stepped cantilever beam design problem, MPS-DCP and TRMPS produce nearly comparable results, among which MPS-DCP is the best. The two algorithms significantly outperform MPS in convergence performance and efficiency. In conclusion, MPS-DCP is expected to be efficient and effective in solving constrained HEB problems.

Table 6 n_e and mean tip deflection obtained by MPS, TRMPS, and MPS-DCP

	MPS	TRMPS	MPS-DCP
n_e	5000	990	990
Mean tip deflection	0.0320	0.0153	0.0150

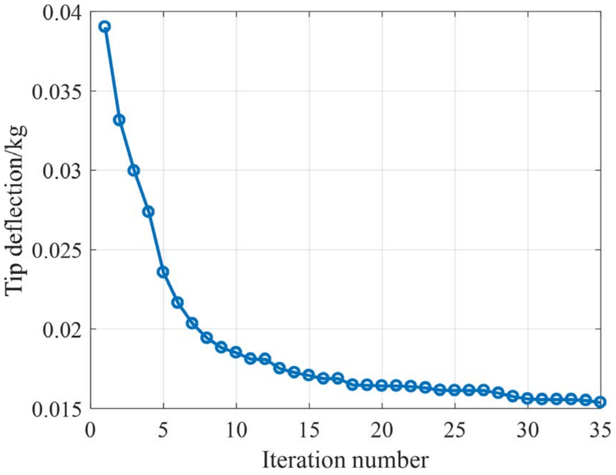

Fig. 11 Convergence history for the stepped cantilever beam design problem using MPS-DCP

5.2 All-Electric GEO Satellite Multidisciplinary Design Optimization Problem

5.2.1 Problem Description. The studied AEGS is comprised of payload module, service module, and solar arrays, as shown in Fig. 12 [35]. This problem involves four disciplines, ten design variables, and seven constraints. The coupling relationship of this MDO problem is organized in the design structure matrix (DSM) and shown in Fig. 13, and the disciplinary modeling approach is detailed in Ref. [35]. The objective of the MDO problem is to minimize the total mass of the satellite [35]. The MDO problem is formulated in Eq. (20). The details of the design variables and constraints are presented in Ref. [35]

$$\text{find } \boldsymbol{x} = [\alpha,\ d_{\text{T}},\ d_{\text{N}},\ A_{\text{sa}},\ H_{\text{S}},\ H_{\text{C}},\ H_{\text{TB}},\ P_{\text{S}},\ P_{\text{C}},\ P_{\text{TB}}]$$

$$\min M_{\text{satellite}} = m_{\text{payload}} + m_{\text{control}} + m_{\text{power}} + m_{\text{structure}} + m_{\text{others}}$$

$$\text{s.t.}\begin{cases} t_{\text{f}} \le 180\text{Day},\ \lambda_{\max} \le 0.05\ \text{deg},\ i_{\max} \le 0.05\ \text{deg}, \\ P_{\text{BOL}} \ge 22.90\,\text{kW},\ P_{\text{EOL}} \ge 16.30\,\text{kW} \\ f_{\text{X}} \ge 12\,\text{Hz},\ f_{\text{Y}} \ge 12\,\text{Hz} \end{cases} \tag{20}$$

The implementation of time-consuming calculation models such as orbit dynamics model and structural finite element analysis model combined with the iterative process of multidisciplinary analysis (MDA) makes the AEGS MDO problem extremely computationally expensive. The mean running time of the MDA process is about 10 min on a computer equipped with Intel Xeon 3.40 GHz CPU and 8 GB memory.

5.2.2 Optimization Results. The maximum number of expensive points $N_{\max}$ is set to be 500, and the maximum number of stall iterations $C_{\text{stall}}^{\max}$ is set to be 20. The initial design variables are determined according to Ref. [35]. The design variables, objective, and constraints of the initial design and the optimized designs are listed in Tables 7–9. The optimization result obtained by MPS-DCP is compared with that of MPS and that of the authors'

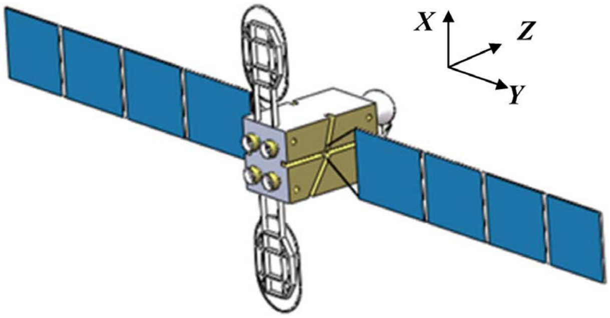

Fig. 12 Illustration of the AEGS

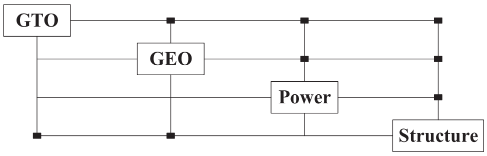

Fig. 13 DSM of the AEGS MDO problem

Table 7 Initial and optimized design variables of the AEGS MDO problem

Design variable	Symbol	Unit	Range	Initial design	Optimized design (MPS)	Optimized design (ARSM-ISES)	Optimized design (MPS-DCP)
Yaw angle in first stage of geostationary transfer orbit (GTO)	α	deg	[0, 60]	0	30.56	29.79	29.61
T position of thruster	d_T	mm	[500, 1180]	1180	655.22	503.28	503.68
N position of thruster	d_N	mm	[800, 1050]	1050	1032.58	962.40	1042.32
Area of solar arrays	A_{sa}	m^2	[90, 120]	110	119.01	117.49	117.41
Core thickness of service cabin plates	H_S	mm	[14, 26]	20	16.77	16.64	15.93
Core thickness of communication cabin plates	H_C	mm	[14, 26]	20	18.17	17.09	16.55
Core thickness of central cylinder	H_{TB}	mm	[14, 26]	20	24.90	22.36	14.61
Ply thickness of service cabin plates	P_S	μm	[0.07, 0.13]	0.1	0.118	0.095	0.071
Ply thickness of communication cabin plates	P_C	μm	[0.07, 0.13]	0.1	0.104	0.081	0.070
Ply thickness of bearing cylinder	P_{TB}	μm	[0.07, 0.13]	0.1	0.091	0.078	0.077

Table 8 Initial and optimized objective of the AEGS MDO problem

Objective	Initial design	Optimized design (MPS)	Optimized design (ARSM-ISES)	Optimized design (MPS-DCP)
Total mass/kg	2552.4	2513.98	2497.1	**2465.3**

Note: The best result is expressed in bold.

Table 9 Constraints of initial design and optimized design of the AEGS MDO problem

Constraint	Symbol	Unit	Range	Initial design	Optimized design (MPS)	Optimized design (ARSM-ISES)	Optimized design (MPS-DCP)
Total orbit transfer time	t_f	Day	≤180	166.11	131.27	130.10	128.63
East/west station keeping accuracy	λ_{max}	deg	≤0.05	0.035	0.029	0.027	0.028
North/south station keeping accuracy	i_{max}	deg	≤0.05	0.036	0.035	0.036	0.036
Beginning-of-life power	P_{BOL}	kW	≥22.90	21.41	23.10	22.90	22.90
Ending-of-life power	P_{EOL}	kW	≥16.30	19.86	21.43	21.20	21.25
First-order rotational modal frequency round X	f_X	Hz	≥12	13.48	13.03	12.25	12.12
First-order rotational modal frequency round Y	f_Y	Hz	≥12	13.39	12.93	12.16	12.00

previously developed efficient ARSM-ISES [21]. MPS is implemented using available open-source codes [36]. DYCORS and TRMPS are not involved in the comparison due to the incapability of handling constraints and the inaccessibility of source codes, respectively.

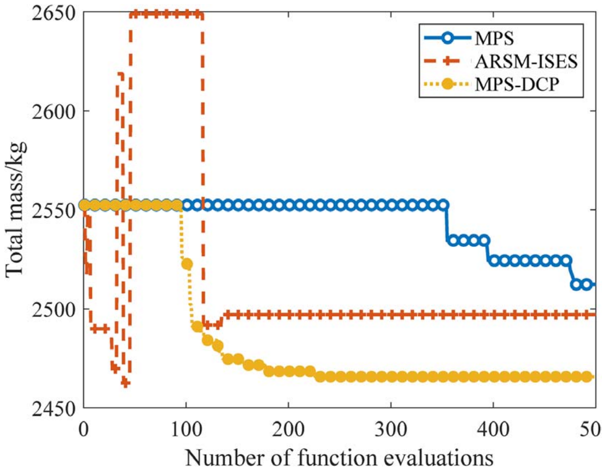

Fig. 14 Convergence history curve for the AEGS MDO problem using MPS-DCP

As shown in Table 8, the optimized design obtained by MPS-DCP yields an 87.1 kg decrease in total mass, which is 7.1% of the satellite components being optimized (i.e., fuel mass, solar array mass, and structure mass) [35]. Figure 14 illustrates the convergence curves of MPS, ARSM-ISES, and MPS-DCP. It is indicated that the MPS-DCP shows better convergence performance than MPS and ARSM-ISES within the same maximum number of expensive points. The optimized design obtained by MPS-DCP is feasible according to Table 9, compared with the initial design that violates the constraint of the beginning-of-life power P_{BOL}. The optimality of the optimized design is further demonstrated by the fact that some of the constraints are active, including the beginning-of-life power and the first-order rotational modal frequencies. Besides, the optimized solution of MPS-DCP is also much better than those of MPS and ARSM-ISES. The comparison results indicate that MPS-DCP shows better convergence performance within the same computational cost compared with MPS and ARSM-ISES in the studied AEGS MDO problem, which demonstrates the effectiveness and practicality of MPS-DCP in solving real-world HEB engineering design optimization problems.

6 Conclusions and Future Work

Referring to the dynamic coordinate perturbation strategy proposed in DYCORS, a novel sensitivity analysis based DCP strategy is developed. Based on the novel DCP strategy, an enhanced MPS method for high-dimensional expensive black-box problems, notated as MPS-DCP, is proposed. In MPS-DCP, the sensitivity indicators are identified using the coefficients of a polynomial response surface and used to generate the discriminative perturbations. The extent of coordinate perturbation is automatically enlarged or shrunk during the optimization so that cheap points are effectively generated toward global optimum from which the expensive points are picked. The convergency of MPS-DCP is also proved based on the convergency analysis theory of stochastic algorithms. The optimization results of numerical benchmarks show promising merits of MPS-DCP compared with the original MPS, DYCORS, TRMPS, and OMID in terms of convergence and robustness performances. All the simulation results suggest that MPS-DCP is a promising global optimization algorithm for HEB problems. Finally, a stepped cantilever beam design optimization problem and an all-electric GEO satellite MDO problem are investigated to evaluate its effectiveness and practicality for real-world engineering design problems. The results show that MPS-DCP yields better feasible optima with the same or less computational cost compared with the competitive optimization methods.

However, due to the inherent limitations of the MPS framework, the convergence and robustness performances of MPS-DCP on extremely high-dimensional problems with hundreds of variables still need to be enhanced. In future work, MPS-DCP is expected to be enhanced by the space mapping and high-dimensional model representation techniques to further improve its capacity for solving extremely high-dimensional problems. Besides, constructing PRS metamodel requires a large number of expensive points when solving high-dimensional problems. More flexible metamodeling techniques such as RBF and Kriging will be adopted in MPS-DCP to further reduce the high-dimensional optimization cost. In addition, more effective constraints handling technique such as Kreisselmeier–Steinhauser (KS) function is expected to be applied to enhance the applicability of MPS-DCP in solving real-world engineering design optimization problem.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 51675047 and 11372036), Chinese Postdoctoral Science Foundation (Grant No. 2019M660668), the Aeronautic Science Foundation of China (Grant No. 2015ZA72004), the International Cooperation Program Fund of Beijing Institute of Technology (Grant No. GZ2018015101), and the Graduate Technological Innovation Project of Beijing Institute of Technology (Grant No. 2018CX100011).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request. The authors attest that all data for this study are included in the paper.

Appendix

The formulations of the numerical benchmark problems are presented as follows.

(1) 16-variable function (F16)

$$f(\boldsymbol{x}) = \sum_{i=1}^{16} \sum_{j=1}^{16} a^{(ij)} (x^{(i)2} + x^{(i)} + 1)\,(x^{(j)2} + x^{(j)} + 1) \tag{A1}$$

$$a^{(ij)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A2}$$

Number of variables: $n_v = 16$.

Design space: $[-1, 1]^{16}$.

Global optimum value: 25.875 [14].

(2) 10D, 20D, and 30D Rosenbrock functions (R10, R20, and R30)

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_v - 1} \left(100\left(x^{(i)} - x^{(i)2}\right)^2 + \left(x^{(i)} - 1\right)^2\right) \tag{A3}$$

Number of variables: $n_v = 10$, 20, or 30.

Design space: $[-5, 5]^{10, 20, \text{ or } 30}$.

Global optimum value: 0 [23].

(3) 10D, 20D, and 30D SUR-T1-14 functions (SUR10, SUR20, and SUR30)

$$f(\boldsymbol{x}) = (x^{(1)} - 1)^2 + (x^{(n_v)} - 1)^2 + n_v \sum_{i=1}^{n_v - 1} (n_v - i)\left(x^{(i)2} - x^{(i+1)}\right)^2 \tag{A4}$$

Number of variables: $n_v = 10$, 20, or 30.

Design space: $[-3, 2]^{10, 20, \text{ or } 30}$.

Global optimum value: 0 [37].

(4) 10D, 20D, and 30D PUR-T1-13 functions (PUR10, PUR20, and PUR30)

$$f(\boldsymbol{x}) = \left[\sum_{i=1}^{n_v} i^3 \left(x^{(i)} - 1\right)^2\right]^3 \tag{A5}$$

Number of variables: $n_v = 10$, 20, or 30.

Design space: $[-3, 3]^{10, 20, \text{ or } 30}$.

Global optimum value: 0 [37].

(5) 10D, 20D, and 30D Griewank functions (GR10, GR20, and GR30)

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_v} \frac{x^{(i)2}}{4000} - \prod_{i=1}^{n_v} \cos\left(\frac{x^{(i)}}{\sqrt{i}}\right) + 1 \tag{A6}$$

Number of variables: $n_v = 10$, 20, or 30.

Design space: $[-600, 600]^{10, 20, \text{ or } 30}$.

Global optimum value: 0 [22].

(6) 10D, 20D, and 30D Zakharov functions (ZF10, ZF20, and ZF30)

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_v} x^{(i)2} + \left(\sum_{i=1}^{n_v} 0.5 i x^{(i)}\right)^2 + \left(\sum_{i=1}^{n_v} 0.5 i x^{(i)}\right)^4 \tag{A7}$$

Number of variables: $n_v = 10$, 20, or 30.

Design space: $[-5, 10]^{10, 20, \text{ or } 30}$.

Global optimum value: 0 [22].

References

[1] Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P. N., and Yang, R.-J., 2004, "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion," Struct. Multidisp. Optim., **27**(5), pp. 302–313.

[2] Long, T., Liu, J., Wang, G. G., Liu, L., Shi, R., and Guo, X., 2016, "Discuss on Approximate Optimization Strategies Using Design of Computer Experiments and Metamodels for Flight Vehicle Design," Chin. J. Mech. Eng., **52**(14), pp. 79–105.

[3] Yondo, R., Andrés, E., and Valero, E., 2018, "A Review on Design of Experiments and Surrogate Models in Aircraft Real-Time and Many-Query Aerodynamic Analyses," Prog. Aerosp. Sci., **96**, pp. 23–61.

[4] Chen, S., Jiang, Z., Yang, S., and Chen, W., 2016, "Multimodel Fusion Based Sequential Optimization," AIAA J., **55**(1), pp. 1–14.

[5] Zhang, Y., Kim, N. H., Park, C., and Haftka, R. T., 2018, "Multifidelity Surrogate Based on Single Linear Regression," AIAA J., **56**(12), pp. 1–9.

[6] Zimmermann, R., 2013, "Gradient-Enhanced Surrogate Modeling Based on Proper Orthogonal Decomposition," J. Comput. Appl. Math., **237**(1), pp. 403–418.

[7] Viana, F. A. C., Haftka, R. T., and Steffen, V., 2009, "Multiple Surrogates: How Cross-Validation Errors Can Help Us to Obtain the Best Predictor," Struct. Multidisp. Optim., **39**(4), pp. 439–457.

[8] Palar, P. S., and Shimoyama, K., 2019, "Efficient Global Optimization With Ensemble and Selection of Kernel Functions for Engineering Design," Struct. Multidisp. Optim., **59**(1), pp. 93–116.

[9] Han, Z., Zhang, Y., Song, C., and Zhang, K., 2017, "Weighted Gradient-Enhanced Kriging for High-Dimensional Surrogate Modeling and Design Optimization," AIAA J., **55**(12), pp. 1–17.

[10] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," J. Glob. Optim., **13**(4), pp. 455–492.

[11] Peng, L., Liu, L., Long, T., and Yang, W., 2014, "An Efficient Truss Structure Optimization Framework Based on CAD/CAE Integration and Sequential Radial Basis Function Metamodel," Struct. Multidisp. Optim., **50**(2), pp. 329–346.

[12] Park, C., Haftka, R. T., and Kim, N. H., 2017, "Remarks on Multi-Fidelity Surrogates," Struct. Multidisp. Optim., **55**(3), pp. 1029–1050.

[13] Shi, R., Liu, L., Long, T., Wu, Y., and Tang, Y., 2018, "Filter-Based Sequential Radial Basis Function Method for Spacecraft Multidisciplinary Design Optimization," AIAA J., **57**(3), pp. 1–13.

[14] Wang, L., Shan, S., and Wang, G. G., 2004, "Mode-Pursuing Sampling Method for Global Optimization on Expensive Black-Box Functions," Eng. Optim., **36**(4), pp. 419–438.

[15] Kazemi, M., Wang, G. G., Rahnamayan, S., and Gupta, K., 2010, "Constraint Importance Mode Pursuing Sampling for Continuous Global Optimization," ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, QC, Canada, Aug. 15–18, pp. 325–334.

[16] Sharif, B., Wang, G. G., and ElMekkawy, T. Y., 2008, "Mode Pursuing Sampling Method for Discrete Variable Optimization on Expensive Black-Box Functions," ASME J. Mech. Des., **130**(2), p. 021402.

[17] Cai, X., Qiu, H., Gao, L., and Shao, X., 2017, "Metamodeling for High Dimensional Design Problems by Multi-Fidelity Simulations," Struct. Multidisp. Optim., **56**(1), pp. 151–166.

[18] Forrester, A., Sobester, A., and Keane, A., 2008, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, Chichester.

[19] Shi, R., Liu, L., Long, T., and Liu, J., 2016, "Sequential Radial Basis Function Using Support Vector Machine for Expensive Design Optimization," AIAA J., **55**(1), pp. 214–227.

[20] Shi, R., Liu, L., Long, T., Wu, Y., and Tang, Y., 2019, "Filter-Based Adaptive Kriging Method for Black-Box Optimization Problems With Expensive Objective and Constraints," Comput. Method. Appl. Mech. Eng., **347**, pp. 782–805.

[21] Long, T., Wu, D., Guo, X., Wang, G. G., and Liu, L., 2015, "Efficient Adaptive Response Surface Method Using Intelligent Space Exploration Strategy," Struct. Multidisp. Optim., **51**(6), pp. 1335–1362.

[22] Cheng, G. H., Younis, A., Hajikolaei, K. H., and Wang, G. G., 2015, "Trust Region Based Mode Pursuing Sampling Method for Global Optimization of High Dimensional Design Problems," ASME J. Mech. Des., **137**(2), p. 021407.

[23] Duan, X., Wang, G. G., Kang, X., Niu, Q., Naterer, G., and Peng, Q., 2009, "Performance Study of Mode-Pursuing Sampling Method," Eng. Optim., **41**(1), pp. 1–21.

[24] Hajikolaei, K. H., Cheng, G. H., and Wang, G. G., 2016, "Optimization on Metamodeling-Supported Iterative Decomposition," ASME J. Mech. Des., **138**(2), p. 11.

[25] Regis, R. G., and Shoemaker, C. A., 2013, "Combining Radial Basis Function Surrogates and Dynamic Coordinate Search in High-Dimensional Expensive Black-Box Optimization," Eng. Optim., **45**(5), pp. 529–555.

[26] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, Chichester.

[27] Xin, B., and Chen, J., 2017, *Intelligent Optimization Methods for Solving Complex Optimization Problems*, Beijing Institute of Technology Press, Beijing, (in Chinese).

[28] Fletcher, R., Leyffer, S., Ralph, D., and Scholtes, S., 2006, "Local Convergence of SQP Methods for Mathematical Programs With Equilibrium Constraints," SIAM J. Optim., **17**(1), pp. 259–286.

[29] Mueller, J.,2016, "User Guide for DYCORS Algorithm- MATLAB," https://ccse.lbl.gov/people/julianem/Manual_DYCORS_matlab.pdf, Accessed January 19, 2017.

[30] Surjanovic, S., and Bingham, D.,2016, "Virtual Library of Simulation Experiments: Test Functions and Datasets," https://www.sfu.ca/~ssurjano/index.html, Accessed January 15, 2015.

[31] Regis, R. G., 2014, "Constrained Optimization by Radial Basis Function Interpolation for High-Dimensional Expensive Black-Box Problems With Infeasible Initial Points," Eng. Optim., **46**(2), pp. 218–243.

[32] Dong, H., Song, B., Dong, Z., and Wang, P., 2018, "SCGOSR: Surrogate-Based Constrained Global Optimization Using Space Reduction," Appl. Soft Comput., **65**, pp. 462–477.

[33] Qian, J., Yi, J., Cheng, Y., Liu, J., and Zhou, Q., 2020, "A Sequential Constraints Updating Approach for Kriging Surrogate Model-Assisted Engineering Optimization Design Problem," Eng. Comput., **36**, pp. 993–1009.

[34] Hajikolaei, K. H., Pirmoradi, Z., Cheng, G. H., and Wang, G. G., 2015, "Decomposition for Large-Scale Global Optimization Based on Quantified Variable Correlations Uncovered by Metamodelling," Eng. Optim., **47**(4), pp. 429–452.

[35] Shi, R., Liu, L., Long, T., Liu, J., and Yuan, B., 2017, "Surrogate Assisted Multidisciplinary Design Optimization for an All-Electric GEO Satellite," Acta Astronaut., **138**, pp. 301–317.

[36] Product Design and Optimization Laboratory, 2019, "Mode Pursuing Sampling (MPS) Method," http://www.sfu.ca/~gwa5/software.html, Accessed May 4, 2016.

[37] Schittkowski, K., 2012, *More Test Examples for Nonlinear Programming Codes*, Springer Science & Business Media, New York.