A Fast-Converging Ensemble Infilling Approach Balancing Global Exploration and Local Exploitation: The Go-Inspired Hybrid Infilling Strategy

Liye Lv
School of Mechanical Engineering, Dalian University of Technology
No.2 Linggong Road, Ganjingzi District, Dalian, 116024, China
lvhexiaoye@mail.dlut.edu.cn

Maolin Shi
School of Mechanical Engineering, Dalian University of Technology
No.2 Linggong Road, Ganjingzi District, Dalian, 116024, China
shl5985336@126.com

Xueguan Song
School of Mechanical Engineering, Dalian University of Technology
No.2 Linggong Road, Ganjingzi District, Dalian, 116024, China
sxg@dlut.edu.cn

Wei Sun
School of Mechanical Engineering, Dalian University of Technology
No.2 Linggong Road, Ganjingzi District, Dalian, 116024, China
sunwei@dlut.edu.cn

Jie Zhang
Department of Mechanical Engineering, The University of Texas at Dallas
Richardson, TX 75080, USA
jiezhang@utdallas.edu

ABSTRACT

1 Corresponding author
Corresponding author: Xueguan Song Email: sxg@dlut.edu.cn
Infilling strategies have been proposed for decades and are widely used in engineering problems. It is still challenging to achieve an effective trade-off between global exploration and local exploitation. In this paper, a novel decision-making infilling strategy named the Go-inspired hybrid infilling (Go-HI) strategy is proposed. The Go-HI strategy combines multiple individual infilling strategies, such as the mean square error (MSE), expected improvement (EI), and probability of improvement (PoI) strategies. The Go-HI strategy consists of two major parts. In the first part, a tree-like structure consisting of several subtrees is built. In the second part, the decision value for each subtree is calculated using a cross-validation (CV)-based criterion. Key factors that significantly influence the performance of the Go-HI strategy, such as the number of component infilling strategies and the tree depth, are explored. Go-HI strategies with different component strategies and tree depth are investigated and also compared with four baseline adaptive sampling strategies through three numerical functions and one engineering case. Results show that the number of component infilling strategies exerts a larger influence on the global and local performance than the tree depth; the Go-HI strategy with two component strategies performs better than the ones with three; the Go-HI strategy always outperforms the three component infilling strategies and the other four benchmark strategies in global performance and robustness, and saves much computational cost.

Keywords: hybrid infilling strategy, Go-inspired, adaptive sampling strategy, surrogate model,

1. Introduction

Surrogate modeling techniques have been widely applied in complex engineering systems because of their high efficiency and low user expertise requirements. Design of experiment (DoE) methods, in which samples are generated in the region of interest, play an essential role in the construction of surrogate models. DoE methods can be broadly classified into two major categories, namely static [1] (or one-shot [2]) sampling and adaptive (or sequential) sampling.

In static or one-shot sampling strategies, the samples used to build surrogate models are all generated at once. To date, several popular static sampling strategies have been proposed, such as orthogonal arrays (OAs) [3], Latin hypercube design (LHS) [4], central composite design (CCD) [5], and fractional factorial design (FFD) [6]. In some cases, the original LHS approach cannot fulfill the space-filling criterion, which states that the sample points should be evenly distributed in the design space. Hence, a number of DoE
methods based on the original LHS method in combination with certain space-filling criteria and optimization algorithms have been developed [7-10]. Overall, static sampling strategies are model-independent methods, meaning that they cannot learn from known samples. However, it is quite challenging to determine the appropriate number and locations of samples without any prior information about the real model.

Adaptive or sequential sampling strategies are characterized as dynamic DoE methods, meaning that the samples used to build surrogate models are generated sequentially. Adaptive sampling strategies are model-dependent methods, meaning that they need to obtain information by learning from a surrogate model constructed from the initial samples, and then generate infilling points. In theory, it is reasonable to expect more points to be placed in complex/nonlinear regions, while fewer points are expected to be placed in simple/linear regions. There are several typical adaptive sampling strategies, including the statistical lower bound (SLB) [11], goal seeking (GS) [12], probability of improvement (PoI) [13] and expected improvement (EI) [14] strategies. In the PoI strategy, points are selected to maximize the probability of improving the function beyond a certain target. The EI strategy, as the name suggests, refers to the amount of improvement we expect to achieve when a new point is added.

In the past two decades, great efforts have been directed toward developments in adaptive sampling strategies, and some achievements have been reported. Jin et al. [15] used the cross-validation (CV) error to exploit local regions and used the maximin distance to explore global regions. Sacks and Welth [16] developed the integral MSE (IMSE) which has a better global performance than MSE. Lam [17] proposed an adaptive design criteria based on the cross validation approach and on the EI strategy. Sóbester et al. [18] proposed a selection criterion based on the EI strategy and investigated the proper initial sample size. Xu [19] developed a CV-Voronoi adaptive sampling strategy based on the leave-one-out (LOO) CV approach and the use of a Voronoi diagram to partition the design space into a set of Voronoi cells. Craud et al. [20] proposed a smart sampling algorithm that optimizes the crowding distance metric (CDM) and the departure function (DF). Liu et al. [21] used an expected prediction error criterion to develop a novel adaptive sampling approach. Xiao et al. [22] proposed an adaptive surrogate model based on an efficient reliability method. To reduce the overall cost of simulation and micromodeling, Deschrijver et al. [23] combined the Voronoi tessellation and local linear approximation (LOLA) algorithms to develop an
adaptive sampling strategy. Crombecq et al. [24] proposed a generic hybrid adaptive sampling approach by combining exploration and exploitation criteria. Later Crombecq et al. [25] used a Monte Carlo-based approximation of Voronoi tessellation for exploration and LOLA for exploitation as the basis of a hybrid sequential design strategy. Van et al. [26] presented a hybrid adaptive sampling strategy based on a LOLA-Voronoi algorithm [14]. Zhou et al. [27] proposed a sequential multi-fidelity (SMF) metamodeling approach for the multi-fidelity surrogate model to determine where to allocate the LF and HF sample points. Jiang et al. [28] developed an adaptive sampling strategy for Kriging based on Delaunay triangulation and technique for order performance by similarity to ideal solution (TOPSIS). Zhang et al. [29] used the cross-validation error and interquartile range (IQR) to build an adaptive sampling method.

For an infilling strategy, it is important to achieve a trade-off between global exploration and local exploitation. Generally, global exploration helps to discover interesting regions that have not yet been detected and to improve the completeness of the adaptive sampling process for global metamodeling, while local exploitation helps to find interesting regions with larger prediction errors from which to sample more points [3]. This paper proposes a hybrid adaptive strategy called the Go-inspired hybrid infilling (Go-HI) strategy, which is inspired by the game of Go and based on the Kriging method. Go is an abstract strategy board game for two players, as shown in Fig. 1, in which each player attempts to surround more territory than the opponent. In general, Go players can choose among hundreds of alternative routes to decide where to place their pieces based on their experience. Evaluating each route requires several steps of backward calculations to obtain a rough estimate of the win rate. Finally, the route with the highest win rate is selected to determine the position of the current piece. As an example, Fig. 1(b) shows two routes, Routes 1 and 2, both of which are calculated three steps backward. Route 1 has a higher win rate, so the player ultimately chooses to place the current piece at point $P_1$. The Go-HI strategy combines three infilling strategies, i.e., the mean square error (MSE), PoI, and EI strategies, by means of a tree-like structure. The number of subtrees increases with increasing the tree depth. Finally, the decision value for each subtree is calculated using a CV-based criterion, and the current infilling sample is chosen based on the calculated decision values.
The remainder of the paper is organized as follows. Mathematical theories related to the Go-HI strategy are briefly described in Section 2. The overall framework of the Go-HI strategy is detailed, along with an example, in Section 3. Section 4 investigates the effects of key factors on the performance of the Go-HI strategy and compares different Go-HI strategies and the other four benchmark adaptive sampling strategies, i.e., CV-Voronoi [19], IQR [29], EIGF [17], and IMSE [16], by applying to six numerical test problems. Conclusions and future work are summarized in Section 5.

2. Mathematical theories

2.1 Kriging model

The Kriging model (KRG) [13] was originally developed for mining and geostatistical applications in which a spatial correlation function (SCF) is used for reference. A general KRG model can be expressed as shown in Eq. (1) below:

\[ y(x) = f(x) + z(x) \]  \hspace{1cm} (1)

where \( y(x) \) is the response function of interest, \( f(x) \) is a polynomial function serving as an approximate basic model, and \( z(x) \) is a stochastic process representing the uncertainty about the mean of \( y(x) \). The KRG model is an interpolative method passing through each sample point. The covariance matrix of \( z(x) \) is formulated as shown in Eq. (2):

\[ \text{cov}[z(x'), z(x'')] = \sigma^2 R[R(x', x'')] \] \hspace{1cm} (2)

where \( R \) is a symmetric correlation matrix; \( R(x', x'') \) is the correlation function between any two
samples, i.e., \( x^i \) and \( x'^i \); and \( n \) is the number of samples. The correlation function could be exponential, Gaussian, cubic, or others. In this paper, we use the Gaussian correlation function, given by:

\[
R(x^i, x'^i) = \exp \left[ -\sum_{k=1}^{n_s} \theta_k | x^i_k - x'^i_k |^p \right]
\]  

(3)

where \( n_s \) is the number of design variables, \( \theta \) and \( P \) are flexible parameters varying from variable to variable, and \( | x^i_k - x'^i_k | \) is the distance between samples \( x^i \) and \( x'^i \) in terms of the \( k^{th} \) variable. The prediction \( \hat{y}(x) \) at a test point \( x \) is given below:

\[
\hat{y}(x) = \hat{\beta} + \mathbf{r}(x) \mathbf{R}^{-1} (\mathbf{y} - f \hat{\beta})
\]  

(4)

where \( \mathbf{y} \) is a column vector of the sample responses, and \( f \) is a column vector of length \( n \) that is filled with ones when \( f(x) \) is a constant vector. \( \mathbf{r}(x) \) is the correlation vector between the test point \( x \) and the samples:

\[
\mathbf{r}(x) = \left[ R(x, x^1), R(x, x^2), \ldots, R(x, x^n) \right]^T
\]  

(5)

Estimates of the mean and variance (\( \hat{\beta} \) and \( \hat{\sigma}^2 \), respectively) of the samples are calculated as shown in Eqs. (6) and (7):

\[
\hat{\beta} = (f^T \mathbf{R}^{-1} f)^{-1} f^T \mathbf{R}^{-1} \mathbf{y}
\]  

(6)

\[
\hat{\sigma}^2 = \frac{(\mathbf{y} - f \hat{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - f \hat{\beta})}{n}
\]  

(7)

2.2 Maximizing the mean squared error (MSE)

MSE is determined by means of a Gaussian-process-based prediction as shown in Eq. (8):

\[
s^2(x) = \sigma^2 \left[ x - f^T \mathbf{R}^{-1} f + \frac{(1 - f^T \mathbf{R}^{-1} f)^2}{f^T \mathbf{R}^{-1} f} \right]
\]  

(8)

where \( s^2(x) \) is an MSE function w.r.t. \( x \) and \( \sigma \), which denotes the constant process variance of a Gaussian field. \( \mathbf{R} \) and \( \mathbf{r} \) are defined in Eqs. (3) and (5), respectively.

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
MSE indicates the degree of uncertainty in an approximate model. The smaller the MSE is, the lower the degree of uncertainty. When an approximate KRG model is used to interpolate samples, the MSE should be zero at the samples, representing that there is no uncertainty at these points. The MSE should be highest at the points that are the farthest from the samples. Hence, it is reasonable to add points in regions with the highest MSE. However, this MSE infilling strategy, which is a model-independent method, will miss the global and local optima and simply fill in gaps [31].

2.3 Maximizing probability of improvement (PoI)

When applying an infilling strategy, we wish to place the next infilling point at the value of \( x \) that will lead to an improvement on the best value observed so far. \( y_{\min} \) denotes the minimum among the responses at the sample points [27]. The PoI is the probability that \( Y(x) \) is less than \( y_{\min} \). Under the assumption of a normally distributed random variable, the probability of an improvement, i.e., \( I = y_{\min} - Y(x) \), can be calculated as shown in Eq. (9) below:

\[
P[I(x)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{0} \exp\left(-\frac{[I - \hat{y}(x)]^2}{2s^2}\right)dl
\]

where \( \hat{y}(x) \) is the prediction at the sample and \( s^2 \) is the uncertainty on the prediction as calculated using Eq. (8).

Equation (9) can be formulated using the Gaussian error function as follows:

\[
P[I(x)] = \frac{1}{2} \left[ 1 + \text{erf}\left( \frac{y_{\min} - \hat{y}(x)}{s\sqrt{2}} \right) \right]
\]

(10)

2.4 Maximizing expected improvement (EI)

Unlike the PoI strategy which focuses on the probability that the result will improve, the EI strategy is related to the amount of improvement expected to be obtained when an infilling point is added. Given an infilling point, the EI can be calculated as shown in Eq. (11), which is expressed in terms of the Gaussian error function.

\[
E[I(x)] = (y_{\min} - \hat{y}(x)) \left[ \frac{1}{2} + \text{erf}\left( \frac{y_{\min} - \hat{y}(x)}{\hat{s}\sqrt{2}} \right) \right] + \hat{s} \sqrt{2\pi} \exp\left[ -\frac{(y_{\min} - \hat{y}(x))^2}{2\hat{s}} \right]
\]

(11)

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
It is observed from Eqs. (10) and (11) that the expressions for the PoI and EI strategies each contain both a global term and a local term, indicating that they can balance global exploration and local exploitation to some extent.

2.5 Cross-validation (CV)

To avoid adding extra samples, the LOO CV criterion is adopted to assess the performance of a surrogate model based on the calculation described in Eq. (12) below:

\[
CV\text{error} = \frac{1}{n-1} \sum_{j=1}^{n-1} (y_j - \hat{y}_j)^2
\]

(12)

where \( CV\text{error} \) is the LOO CV error, \( y_j \) and \( \hat{y}_j \) denote the true and predicted responses, respectively, at the \( j^{th} \) sample as calculated based on the \( n-1 \) samples by excluding the \( j^{th} \) sample.

2.6 Performance criterion

In this paper, we use two criteria, i.e., the coefficient of determination \( (R^2) \) as a global evaluation metric shown in Eq. (13) and the normalized max absolute error (NMAE) as a local metric shown in Eq. (14), to assess the global and local performance of different infilling strategies, respectively.

\[
R^2 = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{y}_i - \hat{y}_j}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{MSE}{Variance}
\]

(13)

\[
NMAE = \max(|y - \hat{y}|) \cdot \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(y_i - \hat{y}_j)^2}{n_{test}}}
\]

(14)

where \( y_i \) denotes the response at the \( i^{th} \) testing sample, \( \hat{y}_i \) denotes the corresponding prediction, \( \bar{y} \) is the mean of the observed responses, and \( n_{test} \) is the number of testing samples. Generally, \( R^2 \) ranges from 0 to 1, and a larger \( R^2 \) or a smaller \( NMAE \) indicates a more accurate surrogate. If the trend of a surrogate model is opposite to that of the true model, \( R^2 \) could be negative. For each numerical problem, a total of 1000\( m \) test points (where \( m \) is the number of design variables) are used to calculate \( R^2 \) and \( NMAE \).
3 The Go-inspired hybrid infilling (Go-HI) strategy

3.1 Modeling using the Go-HI strategy

Figure 2 depicts the overall framework of the proposed Go-HI strategy. This proposed infilling strategy is a decision-making method based on a tree-like structure. In theory, the Go-HI strategy is able to combine an infinite number of individual infilling strategies by constructing a tree-like structure of unlimited depth to be used to select each new infilling point. The size of the tree-like structure is determined by the number of strategies and the depth of the tree. The number of branches of the tree-like structure is expressed as $n^{l-1}$, where $n$ and $l$ are the number of strategies and the tree depth (expressed as a number of levels), respectively. To introduce the Go-HI strategy concisely and effectively, only two infilling strategies (MSE and EI) are considered in the flowchart ($n = 2$), the tree depth is chosen to be 3 ($l = 3$), and the procedure for generating one infilling point is presented as an example. Hence, the tree-like structure contains $2^2$ branches in total. The proposed Go-HI strategy is mainly composed of four steps.

◆ Step 1: Model Construction Step

Construct an initial KRG surrogate model $SM_0$, which is mostly inaccurate, based on an initial sample set $S_0$.

◆ Step 2: Policy Step (Tree-Like Structure)

In this step, the strategy library consists of the MSE and EI strategies. The tree-like structure contains 4 subtrees ($n^{l-1} = 2^2$) and 7 nodes, represented by numbered black dots. We will describe one of the subtrees in detail; others are similar. The root node is Node 1, and its depth is given as Level 1. Subtree 1 is taken as an example to illustrate the process of policy step.

Nodes 1, 2, and 4 compose Subtree 1.

(1) Level 2: Node 2

The initial MSE infilling point $x_1^1$, which is a virtual infilling point, is generated by using
the MSE strategy based on the initial surrogate model $SM_0$ that was constructed in Step 1. Accordingly, the corresponding prediction $y^1_t$ at $x^1_t$ can be obtained from $SM_0$. The value of the predicted MSE of $SM_0$ at point $x^1_t$ is $mse^1_t$. The prediction at $x^1_t$ is indeterminate because of the existence of uncertainty in the surrogate model. It is assumed that each virtual infilling point corresponds to five virtual predictions, as detailed below.

Suppose there is a normal distribution $N(\mu_t, \sigma^2)$, and the probability density function ($pdf$) is shown in Eq. (15). Five virtual predictions that are evenly distributed within a range of $[y^1_t - 2\sqrt{mse^1_t}, y^1_t + 2\sqrt{mse^1_t}]$ can be denoted by $\tilde{y}^1_i = \{\tilde{y}^1_i \}_{i=1,2,...,5}$ which are sorted from the smallest to the largest. $\tilde{y}^1_i$ is equal to $y^1_t + 2(0.5i - 1.5)mse^1_t$. The vector element corresponding to $i=1$ has the smallest value, i.e., $\tilde{y}^1_{15} = y^1_t + 2mse^1_t$; the vector element corresponding to $i=5$ has the largest value, i.e., $\tilde{y}^1_{15} = y^1_t + 2mse^1_t$. Each virtual sample pair, i.e., $(x^1_t, \tilde{y}^1_i)$, is separately added to the initial sample set $S_0$ to form a new virtual sample cell $\hat{S}^1_i = \{\hat{y}^1_i \}_{i=1,2,...,5}$, where $\hat{y}^1_i$ consists of $S_0$ and the virtual sample pair $(x^1_t, \tilde{y}^1_i)$. Based on the new sample cell $\hat{S}^1_i$, five CVerrors, i.e., $CV^{1\mu}_i$ are calculated by Eq. (12). The corresponding value of pdf, i.e., $f^{1\mu}_i$, can be calculated by Eq. (15). The CVerror of this node, namely, $ACVerror^1_t$ is obtained by Eq. (16). Finally, the pair $(x^1_t, \tilde{y}^1_{15})$ is added to the sample set $S_0$ to form a new virtual sample set $S^1_t$, which is used to construct a new surrogate model $SM^1_t$ for Nodes 4 and 5.

(2) Level 3: Node 4

In Node 4, the MSE infilling point $x^2_t$ is generated by using the MSE infilling strategy based on the surrogate model $SM^1_t$ constructed in Node 2. Accordingly, the corresponding prediction $y^2_t$ at $x^2_t$ can be determined based on $SM^1_t$. The MSE of $SM^1_t$ at $x^2_t$ is $mse^2_t$. Because of the existence of uncertainty in the surrogate model, the prediction at $x^2_t$ is indeterminate. The virtual predictions at $x^2_t$ are determined in the same manner described
above.

Each virtual pair, i.e., \((x_i^2, \hat{y}_i^2)\), is separately added to the sample set \(S_1^2\) to form a new virtual sample cell \(\hat{S}_1^2 = \{\hat{S}_1^2\}_i (i=1,2,...,5)\), where \(\hat{S}_1^2\) consists of \(S_1^2\) and the pair \((x_i^2, \hat{y}_i^2)\). Based on the new sample cell \(\hat{S}_1^2\), five CV errors are calculated using Eq. (12); then, the mean of these five CV errors, denoted by \(ACVerror^2_i\), is obtained. Finally, the pair \((x_i^2, \hat{y}_i^2)\) is added to the sample set \(S_1^2\) to form a new virtual sample set \(S_1^3\), which is used to construct a new surrogate model \(SM_1^3\).

**Step 3: Value Step**

For each descendant (i.e., Nodes 2, 3, 4, 5, 6, and 7) of the tree-like structure, the normalized CV error (\(NACVerror\)) can be calculated as shown in Eq. (17). Based on \(NACVerror\), the nodal solution for each subtree is then calculated based on the criterion shown in Eq. (16).

\[
f_j^{(k-1)v} = \frac{1}{2\pi \sqrt{mse_j^{(k-1)v}}} \exp \left( -\frac{(y_j^{(k-1)} - \hat{y}_j^{(k-1)v})^2}{2mse_j^{(k-1)v}} \right)
\]  

(15)

\[
ACVerror_j^{k-1} = \sum_{i=1}^{K} f_j^{(k-1)v} CV_j^{(k-1)v} \left( i=1,2,...,5, k=2,...,l; j=1,...,n^{k-1} \right)
\]  

(16)

\[
NACVerror_j^{k-1} = \frac{ACVerror_j^{k-1}}{\sum_{j=1}^{K} ACVerror_j^{k-1}} \left( k=2,...,l; j=1,...,n^{k-1} \right)
\]  

(17)

\[
Nod-Sol_j = \prod_{k=2}^{K} NACVerror_j^{(k-1)} \left( k=2,...,l; j=1,...,J, J = n^{l-1} \right)
\]  

(18)

where \(n\) is the number of component infilling strategies, \(l\) is the tree of depth, and \(Nod-Sol_j\) is the nodal solution for the \(j\)-th subtree. As mentioned above, we take the Go-HI strategy with two components (\(n=2\)) and a tree depth of three (\(l=3\)) as an example to demonstrate.

It is assumed that the subtree with the minimum \(Nod-Sol\) corresponds to the optimal decision. By comparing the calculated \(Nod-Sol\) values, the smallest can be selected to determine the optimal decision. If \(Nod-Sol_1\) or \(Nod-Sol_2\) is the smallest, then Subtree 1 or Subtree 2, respectively, is the
optimal decision, and the initial MSE infilling point produced in Node 2 will be added to the initial sample set \( S_0 \); otherwise, the initial EI infilling point produced in Node 3 should be added.

◆ Step 4: End Step

The surrogate models are sequentially updated. Steps 1 to 3 are repeated until the prediction performance meets the specified requirements.

![Diagram](image)

**Notes:**
1) MSE denotes the generation of an infilling point using the MSE infilling strategy;
2) EI denotes the generation of an infilling point using the EI infilling strategy;
3) Process of updating Surrogates:

- Updated Response
- Surrogate Modeling

**Fig. 2** Modeling using the Go-HI strategy

### 3.2 Illustrative example of modeling using the Go-HI strategy

To further clarify how modeling is performed using the Go-HI strategy, a one-dimensional (1-D) numerical calibration problem, as shown in Eq. (19) below, is adopted to illustrate the modeling process. Figure 3 shows the flowchart of generating the first infilling point.

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
\[ y = (6x - 2)2\sin\left[\frac{2}{2(6x - 2)}\right], 0 < x < 1 \quad (19) \]

- Step 1: The initial surrogate model \( SM_0 \), represented by a black line in Fig. 4(a), is constructed from the initial sample set \( S_0 = \{x = 0, 0.3, 1; y = 3.0, -0.015; 15.8\} \), which includes the three samples indicated by red balls in the figure. The uncertainty \( mse_0 \) in the surrogate model \( SM_0 \), represented by the yellow shaded area, is calculated using Eq. (8).

- Step 2: Policy Step (Tree-Like Structure)

  Two infilling strategies, MSE and EI, are ensembled in the Go-HI strategy in this example, and the depth of the tree-like structure is three. The tree-like structure in this example consists of four subtrees, and Subtrees 1 and 3 are described in detail.

  - Subtree 1: As shown in Fig. 4(a), the initial MSE infilling point \( x^1_i = 0.78 \), located at the position indicated by the blue line, is generated by using the MSE infilling strategy. Five virtual predictions are chosen within a range of plus or minus the uncertainty, namely, \( \{\hat{y}_i^u = 3.57, 7.48, 11.38, 15.29, 19.20\} (i = 1, \ldots, 5) \), as shown by the blue squares. Each virtual sample pair \( (x^1_i, \hat{y}_i^u) \) is separately added to the initial sample set \( S_0 \) to form a new virtual sample cell \( \hat{S}_i^1 \) \( (i = 1, 2, \ldots, 5) \). Based on the new sample cell \( \hat{S}_i^1 \), five CVerrors are calculated using Eq. (12), and the average of these five CVerrors, \( ACVerror_i^1 \), is equal to 8.198.

    The virtual sample \( (x^1_i, \hat{y}_i^{15}) = (0.78, 19.20) \), marked with the larger blue square, is finally added to \( S_0 \) to form a new virtual sample set \( S_1^1 \), which is used to construct the virtual surrogate model \( SM_1^1 \) for the next stage, represented by the thicker blue line in Fig. 4(b).

    In Level 3, the MSE infilling point \( x^2_i = 0.23 \), represented by the purple line, is generated by using the MSE infilling strategy based on the virtual surrogate model \( SM_1^1 \). In a similar manner as described above, the five virtual predictions at \( x^2_i \) are determined to be \( \{\hat{y}_i^{2u} = -3.92, -1.63, 0.65, 2.94, 5.22\} (i = 1, \ldots, 5) \). A new virtual sample cell \( \hat{S}_i^2 \) is formed by adding
the corresponding virtual sample pairs to \( S_i^1 \). Based on the new sample cell \( \hat{S}_i^1 \), \( \text{ACVerror}^1_i = 11.546 \) is obtained.

- **Subtree 3:** As shown in Fig. 4(c), the initial EI infilling sample \( x_i^1 = 0.20 \), at the location indicated by the green line, is generated by using the EI infilling strategy. Five virtual predictions at \( x_i^1 = 0.20 \) are \( \{ \hat{y}_i^1 = -5.04, -1.98, 1.07, 4.13, 7.18 \} (i = 1,...,5) \), as shown by the green squares. Each virtual sample \( (x_i^1, \hat{y}_i^1) \) is separately added to the initial sample set \( S_0 \) to form a new sample cell \( \hat{S}_i^1 = \{ \hat{S}_i^1 \} (i = 1,2,...,5) \). Based on the new sample cell \( \hat{S}_i^1 \), five CVerrors are calculated using Eq. (12), and \( \text{ACVerror}^4_i = 12.261 \) is then obtained by averaging these five CVerrors.

The virtual sample \( (x_i^1, \hat{y}_{i5}^1) = (0.20, 7.18) \), marked with the larger green square, is finally added to \( S_0 \) to form a new sample set \( S_i^1 \), which is used to construct the virtual surrogate model \( SM_i^1 \) for the next stage, represented by the thicker green line in Fig. 4(c).

In Level 3, the MSE infilling point \( x_i^1 = 0.46 \), represented by the purple line in Fig. 4(c), is generated by using the MSE infilling strategy based on the virtual surrogate model \( SM_i^1 \). In a similar manner as described above, the five virtual predictions at \( x_i^1 \) are determined to be \( \{ \hat{y}_i^2 = -1.23, 0.24, 1.70, 3.16, 4.62 \} (i = 1,...,5) \). A new virtual sample cell \( \hat{S}_i^2 = \{ \hat{S}_i^2 \} \) that includes \( S_i^1 \) is formed, and \( \text{ACVerror}^3_i = 9.874 \) is obtained.

The processes for the other two subtrees are similar to those for Subtrees 1 and 3. To simplify the presentation of the paper, their descriptions are omitted.

- **Step 3:** The nodal solution \( \text{Nod-Sol} \) for each subtree is calculated using Eq. (17) based on the \( \text{NACVerror} \) value for each node, and the results are listed in Table 1. We assume that the subtree with the minimum nodal solution corresponds to the optimal decision, and its corresponding initial infilling point will be added to the initial sample set.

Subtree 2 has the minimum nodal solution, as indicated in bold-faced type in Table 1; therefore, its...
corresponding initial infilling sample \((x_i^1, y_i^1) = (0.78, -5.68)\) can be added to the initial sample set \(S_0\).

Thus, the first iteration of the selection of an infilling sample is completed. To demonstrate the performance of the Go-HI strategy, five iterations of the process from Step 1 to Step 3 are considered in this example.

It is seen from Fig. 3 that the tree-like structure has four subtrees. Nodes 1, 2, and 4 construct Subtree 1; Nodes 1, 2, and 5 construct Subtree 2; Nodes 1, 3, and 6 construct Subtree 3; and Nodes 1, 3, and 7 construct Subtree 4. The initial surrogate model in Node 1 is built from the initial sample set, i.e., \(X = [0; 0.3; 1]\). Then the MSE and EI strategies generate two new virtual samples, i.e., \(x = 0.78\) and \(x = 0.20\), which are candidates of the infilling point from the first iteration. According to the CV-based criterion, the decision value of Subtree 2 is the lowest, then we choose \(x = 0.78\) as the first infilling point.

Each infilling sample added through the Go-HI strategy is depicted in Fig. 5, and the surrogate models constructed from the updated sample sets are illustrated by blue dashed lines. From the Table 2, we can see that in Step 1, the predictive performance of the initial surrogate model is quite poor, with a negative \(R^2\). After one iteration, the Go-HI strategy results in a pretty nice outcome, with a \(R^2\) of 0.8. After five iterations, the accuracy of the surrogate model is close to 1. The EI strategy leads to the worst performance after four iterations. Go-HI strategy successively selects MSE, EI, EI, EI, and MSE strategies to generate five new samples as listed in Table 2.

### Table 1 Node values and nodal solutions of the first iteration

<table>
<thead>
<tr>
<th></th>
<th>Subtree 1</th>
<th>Subtree 2</th>
<th>Subtree 3</th>
<th>Subtree 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2- ACError(_j^1)</td>
<td>8.198</td>
<td>8.198</td>
<td>12.261</td>
<td>12.261</td>
</tr>
<tr>
<td>Level 2- NACError(_j^1)</td>
<td>0.401</td>
<td>0.401</td>
<td>0.599</td>
<td>0.599</td>
</tr>
<tr>
<td>Level 3- NACError(_j^2)</td>
<td>0.266</td>
<td>0.256</td>
<td>0.227</td>
<td>0.251</td>
</tr>
<tr>
<td>Nod-Sol</td>
<td>0.107</td>
<td><strong>0.103</strong></td>
<td>0.136</td>
<td>0.150</td>
</tr>
</tbody>
</table>

Corresponding author: Xueguan Song  
Email: sxg@dlut.edu.cn
Fig. 3 Flowchart generating of the first infilling point of the Forretal function by using the 3L-2S Go-HI strategy.

Fig. 4 Determination of virtual predictions for Subtrees 1 and 3.
Corresponding author: Xueguan Song  
Email: sxg@dlut.edu.cn

Fig. 5 Surrogate models generated with different infilling strategies after five iterations

<table>
<thead>
<tr>
<th>Iteration</th>
<th>MSE</th>
<th>EI</th>
<th>Go-HI</th>
<th>Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-2.51</td>
<td>-2.51</td>
<td>-2.51</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0.80</td>
<td>-2.49</td>
<td>0.80</td>
<td>MSE</td>
</tr>
<tr>
<td>2</td>
<td>0.82</td>
<td>-2.11</td>
<td>0.82</td>
<td>EI</td>
</tr>
<tr>
<td>3</td>
<td>0.91</td>
<td>-2.15</td>
<td>0.95</td>
<td>EI</td>
</tr>
<tr>
<td>4</td>
<td>0.98</td>
<td>-1.45</td>
<td>0.99</td>
<td>EI</td>
</tr>
<tr>
<td>5</td>
<td>0.99</td>
<td>0.94</td>
<td>0.99</td>
<td>MSE</td>
</tr>
</tbody>
</table>

4. Effects of key factors

The effects of key factors through six numerical test functions [16], i.e., the 1-D ONEVAR function (OV), the 2-D Branin function (BR), the 2-D Shubert function (SH), and the 3-D Hartmann function (HA).

4.1 Test functions

ONEVAR function (OV):

\[
f(x) = 3(1-x)^2 \exp(-x^2 - 1) - 10(0.2x - x^3) \exp(-x^5), x \in [-4,1]^d
\]

Branin function (BR):

\[
f(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 \right) + 10 \left(1 - \frac{1}{8\pi} \right) \cos (x_1) + 10, x \in [-5,10]^2
\]

Shubert function (SH):

\[
f(x) = \left(\sum_{i=1}^{d} i \cos \left((i+1)x_1 + i\right) \right) \left(\sum_{i=1}^{d} i \cos \left( (i+1)x_2 + i\right) \right), x \in [0,1]^d
\]

Corresponding author: Xueguan Song  
Email: sxg@dlut.edu.cn

17
Hartmann function (HA):

\[
f(x) = -\sum_{i=1}^{4} \alpha_i \exp \left( -\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij}\right)^2 \right), \quad x \in [0,1]^3
\]

where \( \alpha = [1.0 \ 2.0 \ 3.0 \ 3.2]^T \), \( A = [a_{ij}] = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix} \), and \( P = [p_{ij}] = 10^{-4} \begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix} \)

4.2 Experiments

As mentioned above, the proposed Go-HI strategy is affected by the component infilling strategies and the depth of the tree-like structure. In this paper, the MSE, EI, and PoI strategies compose the strategy library for the Go-HI strategy. In theory, we can select one, two, or three strategies from the strategy library to build different Go-HI strategies. There are three ways to select two strategies from the strategy library, i.e., three possible combinations of two strategies, namely, MSE/EI, MSE/PoI, and EI/PoI. We first use the BR test function, which is a 2-D function combining polynomial and cosine terms, to explore which combination of the two strategies is the best for the Go-HI strategy in Subsection 4.2.1. Next, the effect of the number of individual infilling strategies and the impact of the tree depth are investigated based on the OV, SH, and HA functions in Subsection 4.2.2. Herein, Go-HI strategies with different component strategies and tree depth are compared with the individual MSE, PoI, and EI strategies.

Table 3 lists the sampling configurations for the six aforementioned test functions. According to Ref. [30], 10n samples (where n is the dimensionality of the variables) are generally sufficient to construct an approximate surrogate model. In this paper, 3n samples (i.e., fewer than 10n) are selected to deliberately construct an invalid initial surrogate model in the first step. A total of 1,000n test samples are used for validation. The number of infilling samples varies depending on the nature of each function. The initial samples and test samples are randomly generated by the built-in MATLAB function \texttt{lhsdesign} in accordance with the maximin criterion. To eliminate the effect of the random sampling plan, all results are averaged over 20 DoE sets.

<table>
<thead>
<tr>
<th>Function</th>
<th>n</th>
<th>Initial samples</th>
<th>Infilling samples</th>
<th>Test samples</th>
</tr>
</thead>
</table>

Corresponding author: Xueguan Song  
Email: sxg@dlut.edu.cn
4.2.1 Effect of each combination of two infilling strategies for the Go-HI strategy

Figure 6 compares the different combinations of two infilling strategies for the Go-HI strategy based on the BR function. As mentioned above, there are three possible combinations of two infilling strategies, namely, MSE/EI, MSE/PoI, and EI/PoI. Figure 6 (a) presents the contour plot of the BR function. For the cases of two-, three-, and four-level trees, as shown in Figs. 6 (b), (c) and (d), respectively, the Go-HI strategies built with the EI/PoI combination always perform the worst, and the Go-HI strategies with the MSE/EI and MSE/PoI have similar performances.

The EI and PoI strategies mainly focus on local exploitation instead of global exploration, and they are vulnerable to becoming trapped in local optima. Hence, the Go-HI strategy based on these two strategies can also easily become stuck in a local optimum, resulting in a poor outcome. In addition, there exists a situation [23] in which the PoI strategy is likely to repeatedly generate infilling samples near known samples when there is no need to do so, causing a jam. Therefore, the performance of the Go-HI strategy based on the EI/PoI combination is also worse than that of the Go-HI strategy based on the MSE/EI and MSE/PoI combinations. Ultimately, the combination of the MSE and EI strategies is selected for the Go-HI strategy with two infilling strategies that is considered in the following sections.

<table>
<thead>
<tr>
<th>Function</th>
<th>Level</th>
<th>No. of Infilling Strategies</th>
<th>No. of Infilling Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>2</td>
<td>6</td>
<td>2,000</td>
</tr>
<tr>
<td>OV</td>
<td>1</td>
<td>3</td>
<td>1,000</td>
</tr>
<tr>
<td>SH</td>
<td>2</td>
<td>6</td>
<td>2,000</td>
</tr>
<tr>
<td>HA</td>
<td>3</td>
<td>9</td>
<td>3,000</td>
</tr>
</tbody>
</table>

4.2.2 Effect of key factors

Figure 7 compares the results of Go-HI strategies with different number of component infilling strategies and different depths of tree for the OV, SH, and HA functions. The legend “2L-2S Go-HI” means that a Go-HI strategy contains two component infilling strategies, i.e., MSE and EI, and the depth of tree in the strategy is two, similarly, the “3L-2S” and “4L-2S” Go-HI strategies; the legend “2L-3S Go-HI” indicates that a Go-HI strategy contains three component infilling strategies, i.e., MSE, EI, and PoI, and

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
the depth of tree in the strategy is three, similarly, the “3L-3S” and “4L-3S” Go-HI strategies.

Figure 7 (a) shows the mean of $R^2$, i.e., a global performance metric, for the OV, SH, and HA functions. It is obvious that the depth of tree and the number of component infilling strategies have more or less impacts on the global prediction performance of the Go-HI strategy. The effect of tree depth is explored by fixing the number of component infilling strategies. For the OV function, which contains two exponential terms, the performance of the 2L-3S Go-HI strategy is slightly better than that of the 3L-3S and 4L-3S Go-HI strategies at most iterations. The results of Go-HI strategies with two component infilling strategies are close to each other. For the SH function, which has many peaks and valleys over the design space, the 4L-3S Go-HI strategy performs better than the 2L-3S and 3L-3S Go-HI strategies, and the 4L-2S Go-HI strategy performs better than the 2L-2S and 3L-2S Go-HI strategies. In the case of the HA function, the performance of the 2L-3S Go-HI strategy is close with that of the 4L-3S strategy, both of which are better than the 3L-3S Go-HI strategy. The 4L-2S Go-HI strategy performs poor at the beginning and becomes better than 3L-2S and 2L-2S Go-HI strategies when increasing the number of iterations.

For a given depth of tree, the effect of the number of component infilling strategies can be illustrated intuitively. For the OV function, the Go-HI strategies with two component strategies performs always better than those with three component strategies except at the 7th iteration. In terms of the SH function, the 4L-3S Go-HI strategy has a similar performance with the 4L-2S Go-HI strategy. However, for the three depth of tree, the Go-HI strategy with tree components performs slightly worse than that with two components, as well as the Go-HI strategy with a tree depth of two. Finally, for the HA function, at low iterations, the Go-HI strategies with three components, i.e., 2L-3S, 3L-3S, and 4L-3S Go-HI strategies, behave slightly better than those corresponding Go-HI strategies two components, i.e., 2L-2S, 3L-2S, and 4L-2S Go-HI strategies. Obviously, the performances of Go-HI strategies with two components becomes better.

Figure 7 (b) shows the mean of NMAE, i.e., a local performance metric, for the OV, SH, and HA functions. When the number of component infilling strategies is fixed, in the case of the OV function, the 3L-2S Go-HI strategy performs much worse than the 2L-2S and 4L-2S Go-HI strategies, and the 2L-3S Go-HI strategy behaves much better than the 3L-3S and 4L-3S Go-HI strategies. For the SH functions, the performance of the 2L-2S Go-HI strategy is better than that of the 3L-2S and 4L-2S Go-HI strategies over
most iterations, and the outcome of the 3L-3S Go-HI strategy is better than those of the 2L-3S and 4L-3S Go-HI strategies. In the case of the HA function, the performance of the 3L-2S Go-HI strategy is slightly better than those of the 2L-2S and 4L-2S Go-HI strategies after 10 iterations, while the 4L-3S Go-HI strategy performs better than the 2L-3S and 3L-3S Go-HI strategies almost overall iterations. Unfortunately, to some extent, the value of NMAE climbs with iterations, that is, the local performance becomes worse and worse. The iteration continues to increase, the NMAE becomes smaller, and the local accuracy becomes better, as the result of the HA function shows.

For a fixed depth of tree, the performance of Go-HI strategy with different component infilling strategy can be compared and analyzed. For the OV function, the two- and three- levels Go-HI strategies with two components always behaves better than those with three components, while the four-levels Go-HI strategy with three components performs worse than that with two components. The results for the SH function are similar to those for the OV function, namely, the two-, three-, and four- levels Go-HI strategies with two components outperform those with tree components. In the case of the HA function, the 2L-2S Go-HI strategy behaves slightly better than the 2L-3S in most iterations; the 4L-2S Go-HI strategy behaves slightly worse than the 4L-3S Go-HI strategy almost overall iterations. Overall, fixing the tree depth, the Go-HI strategies with two components perform better than those with tree component in most cases.

![Graphs comparing Go-HI strategies with different tree depths and different components](image)

**Fig. 7 Comparing Go-HI strategies with different tree depths and different components**

Overall, for all Go-HI strategies, the global performances are improved with iterations, while the local
performances deteriorate within certain limits; the trends of global and local performances are consistent across all Go-HI strategies. It is concluded that the number of component infilling strategies exerts a large influence on the global and local performance of the Go-HI strategy; for a given tree depth, in most cases, the Go-HI strategy with two component infilling strategies performs better than the one with three, indicating that it is not generally beneficial to simply consider as many infilling strategies as possible. The depth of tree also has a strong effect on the local performance of the Go-HI strategy, and in most cases, the Go-HI strategy with a tree depth of four performs better. It is important to note that both the computation cost and level of uncertainty may increase with the number of tree levels in Go-HI. It is also statisticced the detail percentage of each component of Go-HI strategy with different components as listed in Table 4.

When the Go-HI strategy consists of two components, i.e., MSE and EI, the EI component makes a greater chontribution than MSE; for Go-HI strategy with three components, i.e., MSE, EI, and PoI, the PoI component contributes the most, followed by EI and then MSE.

<table>
<thead>
<tr>
<th>Function</th>
<th>Component</th>
<th>2L-2S</th>
<th>2L-3S</th>
<th>3L-2S</th>
<th>3L-3S</th>
<th>4L-2S</th>
<th>4L-3S</th>
</tr>
</thead>
<tbody>
<tr>
<td>OV</td>
<td>MSE</td>
<td>54%</td>
<td>52%</td>
<td>39%</td>
<td>24%</td>
<td>36%</td>
<td>24%</td>
</tr>
<tr>
<td></td>
<td>EI</td>
<td>46%</td>
<td>17%</td>
<td>61%</td>
<td>33%</td>
<td>64%</td>
<td>36%</td>
</tr>
<tr>
<td></td>
<td>PoI</td>
<td>-</td>
<td>31%</td>
<td>-</td>
<td>43%</td>
<td>-</td>
<td>39%</td>
</tr>
<tr>
<td>SH</td>
<td>MSE</td>
<td>34%</td>
<td>19%</td>
<td>19%</td>
<td>14%</td>
<td>27%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>EI</td>
<td>66%</td>
<td>26%</td>
<td>81%</td>
<td>28%</td>
<td>73%</td>
<td>28%</td>
</tr>
<tr>
<td></td>
<td>PoI</td>
<td>-</td>
<td>55%</td>
<td>-</td>
<td>58%</td>
<td>-</td>
<td>59%</td>
</tr>
<tr>
<td>HA</td>
<td>MSE</td>
<td>19%</td>
<td>5%</td>
<td>8%</td>
<td>6%</td>
<td>21%</td>
<td>4%</td>
</tr>
<tr>
<td></td>
<td>EI</td>
<td>81%</td>
<td>33%</td>
<td>92%</td>
<td>35%</td>
<td>79%</td>
<td>37%</td>
</tr>
<tr>
<td></td>
<td>PoI</td>
<td>-</td>
<td>61%</td>
<td>-</td>
<td>59%</td>
<td>-</td>
<td>59%</td>
</tr>
</tbody>
</table>

4.3 Comparisons of different infilling strategies

In this section, the Go-HI strategy with a tree depth of four and three component infilling strategies, i.e., the 4L-3S Go-HI strategy, is used to validate the performance of Go-HI strategy by comparing with the three component infilling strategies (i.e., MSE, EI, and PoI) and four adaptive sampling strategies (i.e., CV-Voronoi, IQR, EIGF, and IMSE). As mentioned in Ref. [31], an $R^2 > 0.8$ indicates a surrogate with good predictive accuracy. Thus in this paper, the iteration process stops when $R^2$ is greater than 0.8.

4.4.1 Numerical cases

The OV, SH, and HA functions are employed to investigate the performance of different adaptive
sampling strategies. Table 4 intuitively lists number of iterations when $R^2 > 0.8$ of different adaptive sampling strategies through three numerical functions and one engineering case.

Figures 8 shows the comparisons of the 4L-3S Go-HI strategy with three component infilling strategies, namely, MSE, EI, and PoI, and four baseline adaptive sampling strategies, i.e., CV-Voronoi, IQR, EIGF, and IMSE strategies, on $R^2$ and standard deviation (Std) of $R^2$. For the three functions, the Go-HI strategy with less number of iterations and lower Std of $R^2$, always performs much better than MSE, EI, and PoI in terms of the global performance and robustness. Compared with CV-Voronoi, IQR, EIGF, and IMSE, the analyses are detailed as follows. For the OV function, the Go-HI strategy performs the best in the global accuracy performance with the lowest number of iterations and better in robustness with lower Std of $R^2$; CV-Voronoi and IMSE perform the worst in predictive performance with more iterations but better in robustness; IQR and EIGF require less iterations to make appreciate surrogate models but behave worse in robustness. In the cases of SH function and HA functions, Go-HI has the best global performance and robustness, EIGF performs the worst in global predictive performance with the largest number of iterations, and IMSE has worse robustness performance with higher value of Std. Figure 9 shows the comparisons of the 4L-3S Go-HI strategy with three component infilling strategies and four benchmark strategies on NMAE and the Std of NMAE. The Go-HI strategy behaves relatively poor in local performance with higher NMAE; CV-Voronoi performs worse both in local performance and robustness; EIGF behaves better in local performance but worse in robustness.

Overall, the Go-HI strategy performs better than other component infilling strategies in terms of both global performance and robustness. Go-HI is able to balance global exploration and local exploitation, and take advantages of each component infilling strategy. The Go-HI strategy also performs better than the four baseline strategies on global performance in most cases, but performs a little worse on local performance. Compared with other infilling strategies, the Go-HI strategy that combines MSE, EI, and PoI strategies, is able to balance the global exploration and local exploitation. It is seen from Table 4 that Go-HI relies on more PoI and EI strategies to exploit the local region and less MSE strategies to explore the global region. The CV-based criterion in the Go-HI strategy can help jump out, and then turn into global exploration. Hence, Go-HI has a faster rate of improvement than those for other strategies, and Go-HI can help build an appropriate surrogate model in fewer iterations. Due to the small infilling samples added in the local region,
the local performance of the Go-HI strategy might be worse than that of other slower infilling strategies. Detailed comparison results are listed in Tables 5 and 6. Compared with the tree component infilling strategies (i.e., MSE, EI, PoI), the Go-HI strategy saves an average of 53%, 27%, and 26% computational cost for the OV, SH, and HA functions, respectively. Compared with the four benchmarks (i.e., CV-Voronoi, IQR, EIGF, and IMSE), the Go-HI strategy saves an average of 20%, 16%, and 70% computational cost for the OV, SH, and HA functions, respectively.

![Fig. 8](image1.png)
Fig. 8 Comparison of different adaptive sampling strategies on $R^2$ for OV, SH, and HA functions

![Fig. 9](image2.png)
Fig. 9 Comparison of different adaptive sampling strategies on NMAE for OV, SH, and HA functions

### 4.4.2 Practical problem

In this section, a pressure relief valve (PRV) as shown in Fig. 10 (a) is adopted to further investigate the performance of the developed Go-HI strategy. The flow enters the valve from the inlet, goes through the gap between the disc and the seat, and discharges from the outlet. To obtain the fluid force ($F$), steady simulations based on a 2-D CFD model with 26023 grids (Fig. 10 (b)) are employed. The standard $k-\varepsilon$ turbulent model is used, the medium is air with an initial temperature of 300K and a pressure of 0.3 MPa. One steady simulation takes about 10 min on a computer with a 3.4GHz processor & 8 cores CPU, and 16G RAM. We choose eight design variables as shown in Fig. 10 (a), ten random initial samples are used to construct the initial surrogate model, and another 40 randomly generated samples are used for validation. Then adaptive sampling strategies are used to generate new infilling points to improve the accuracy of the surrogate model. The design space of the eight variables is defined in Eq. (22).

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
Figure 11 shows the comparison of the 4L-3S Go-HI strategy with the three component infilling strategies and the four baseline adaptive sampling strategies. It is seen that the Go-HI strategy uses five iterations to achieve an $R^2$ greater than 0.8, and the rate of improvement is significantly faster than that of other strategies. The fluctuating trends of MSE, EI, and PoI indicate that infilling new samples may also lead to poor accuracy. The EIGF strategy converges the most slowly, and from the 3\textsuperscript{rd} iteration to the 14\textsuperscript{th} iteration, there is no significant improvement in accuracy. IQR performs better than EIGF but worse than CV-Voronoi. In addition to the global performance, when the accuracy is satisfied, the 4L-3S Go-HI strategy also shows the lowest NMAE, which is significantly smaller than other strategies. Followed by 4L-3S Go-HI, CV-Voronoi performs the second, EIGF performs the third, and IQR performs the worst. As compared in Tables 5 and 6, CV-Voronoi, IQR, EIGF, and IMSE require 11, 15, 19, and 5 iterations to achieve $R^2 > 0.8$, respectively; MSE, EI, and PoI need 6, 9, and 9 iterations to achieve, respectively. The Go-HI strategy successively selects MSE, EI, EI, PoI, and MSE strategies to generate five new samples. Overall, Go-HI outperforms the three component infilling strategies and the four baseline strategies in both global and local performance in this case, and saves an average of 60\%-150\% computational cost.

Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn
Fig. 11 Comparison of different adaptive sampling strategies on the $R^2$ and NMAE

Table 5 Comparison of Go-HI with three component infilling strategies on number of iterations

<table>
<thead>
<tr>
<th>Cases</th>
<th>Go-HI</th>
<th>Average Savings</th>
<th>Component Infilling Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSE</td>
</tr>
<tr>
<td>OV</td>
<td>5</td>
<td>53%</td>
<td>7</td>
</tr>
<tr>
<td>SH</td>
<td>20</td>
<td>27%</td>
<td>26</td>
</tr>
<tr>
<td>HA</td>
<td>14</td>
<td>26%</td>
<td>18</td>
</tr>
<tr>
<td>Valve</td>
<td>5</td>
<td>60%</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 6 Comparison of Go-HI with four baseline adaptive sampling strategies on number of iterations

<table>
<thead>
<tr>
<th>Cases</th>
<th>Go-HI</th>
<th>Average Savings</th>
<th>Baseline Adaptive Sampling Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CV-Voronoi</td>
</tr>
<tr>
<td>OV</td>
<td>5</td>
<td>20%</td>
<td>7</td>
</tr>
<tr>
<td>SH</td>
<td>20</td>
<td>16%</td>
<td>23</td>
</tr>
<tr>
<td>HA</td>
<td>14</td>
<td>70%</td>
<td>15</td>
</tr>
<tr>
<td>Valve</td>
<td>5</td>
<td>150%</td>
<td>11</td>
</tr>
</tbody>
</table>

5 Conclusions

This paper developed a novel decision-making infilling strategy that is inspired by the game of Go, named the Go-inspired hybrid infilling (Go-HI) strategy, with the aim of making full use of the advantages of different component infilling strategies. The Go-HI strategy consists of two major parts. In the first part, a tree-like structure consisting of several subtrees is established. The number of subtrees depends on the number of individual infilling strategies and the depth of the tree. In each node of the tree-like structure, a virtual sample is generated via one of the component strategies. In the second part, the decision value for each subtree is calculated in accordance with a cross-validation (CV)-based criterion.

Different combinations of the two component infilling strategies for the Go-HI strategy have been explored and compared. Given the three individual strategies considered in this paper, there are three possible combinations of the two infilling strategies, namely, MSE/EI, MSE/PoI, and EI/PoI. The results
show that the Go-HI strategy built with the MSE/EI combination performs the best.

The effect of different component infilling strategies and tree depth had been investigated through three numerical functions. For all Go-HI strategies, the global performances were improved with iterations, while the local performances deteriorate within certain limits. The number of component infilling strategies exerted a larger influence on the global and local performance of the Go-HI strategy than the tree depth; the Go-HI strategy with two component infilling strategies performed better than the ones with three, indicating that it is not generally beneficial to simply consider as many infilling strategies as possible. In addition, the depth of tree had a strong effect on the local performance of the Go-HI strategy, and in most cases, the Go-HI strategy with a tree depth of two performed better. In addition, when Go-HI strategy consists of two components, i.e., MSE and EI, the EI strategy accounts for a much larger proportion than MSE; when the Go-HI strategy contains three components, i.e., MSE, EI, and PoI, the PoI strategy contributes the most, followed by EI and finally MSE.

Comparisons of Go-HI strategy with the three component infilling strategies, i.e., MSE, EI, and PoI and the four baseline strategies, i.e., CV-Voronoi, IQR, EIGF, and IMSE, have also been conducted through three numerical functions and one engineering problem. Results showed that Go-HI always outperforms the three components and the benchmarks in global performance and robustness, but sometimes behaves worse than other strategies in local performance. Go-HI can balance global exploration and local exploitation and make full use of advantages of each component infilling strategy. Compared with the three components (i.e., MSE, EI, PoI), Go-HI can save an average of 53%, 27%, and 26% computational cost. Compared with the four benchmarks (i.e., CV-Voronoi, IQR, EIGF, and IMSE), Go-HI can save an average of 20%, 16%, 70%, and 108% computational cost for OV, SH, HA, and the relief valve case, respectively.

In this paper, the CV-based criterion was used to determine the candidate infilling strategies. However, the CVerror is good for filtering out inaccurate surrogates without sufficient samples [34], but may not identify the best one. Future work will improve the assessment criterion by modifying the CVerror and employing other metrics.
ACKNOWLEDGMENT

The research was supported by the National Natural Science Foundation of China (Grant Nos. 51505061 and U1608256). The authors also would like to thank Xiaomin Xiang, Kunpeng Li, and Chaoyong Zong for their efforts.

References


Corresponding author: Xueguan Song    Email: sxg@dlut.edu.cn


[29] Zhang, Y., Kim, N. H., & Haftka, R. T. (2019). General Surrogate Adaptive Sampling using Interquartile Range for Corresponding author: Xueguan Song  Email: sxg@dlut.edu.cn


