

Multi-start Space Reduction (MSSR) surrogate-based global optimization method

Huachao Dong^{1,2} · Baowei Song¹ · Zuomin Dong² · Peng Wang¹

Received: 6 September 2015 / Revised: 7 March 2016 / Accepted: 28 March 2016 / Published online: 13 April 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In this work, a new Multi-start Space Reduction (MSSR) and surrogate-based search algorithm is introduced for solving global optimization problems with computationally expensive black-box objective and/or expensive black-box constraint functions. In this new algorithm, the design space is classified into: the original design space or global space (GS), the reduced medium space (MS) that contains the promising region, and the local space (LS) that is a local area surrounding the present best solution in the search. During the search, a kriging-based multi-start optimization process is used for local optimization, sample selection and exploration. In this process, Latin hypercube sampling is used to acquire the starting points and sequential quadratic programming (SQP) is used for the local optimization. Based upon a newly introduced selection strategy, better sample points are obtained to supplement the kriging model, and the estimated mean square error of kriging is used to guide the search of the unknown areas. The multi-start search process is carried out alternately in GS, MS and LS until the global optimum is identified. The newly introduced MSSR algorithm was tested using various optimization benchmark problems, including fifteen bound constrained examples, two nonlinear constrained optimization problems, and four nonlinear constrained engineering applications. The test results revealed noticeable advantages of the new algorithm in dealing with computationally expensive

black-box problems. In comparison with two nature-inspired algorithms, three space exploration methods, and two recently introduced surrogate-based global optimization algorithms, MSSR showed improved search efficiency and robustness.

Keywords Multi-start optimization · Kriging model · Space reduction · Expensive black-box problems · Nonlinear constrained engineering applications

1 Introduction

With rapid advances of computer modeling techniques, numerical simulations and analyses became common engineering design tools. However, most of these simulations and analyses are computationally intensive with black-box characteristics (Edke and Chang 2011; Xiang et al. 2012; Shan and Wang 2010). For instance, the design of airfoil and aircraft wings requires expensive black-box computational fluid dynamics (CFD) simulation to calculate the lift and finite element analysis (FEA) to ensure structure strength and light weight (Eves et al. 2012). Crash simulation of vehicle bodies using FEA is another example (Harmati et al. 2010; Zhang et al. 2008). In both cases, the identification of the optimized design parameters of the wing shape/structure and vehicle body requires global optimization using the computationally expensive black-box simulations and analyses as objective and constraint functions. Design optimization traditionally uses a large number of objective/constraint function evaluations in the iterative searches to identify the solution, and the number of search iterations increases dramatically for a non-unimodal optimization problem that requires global optimization algorithms. More efficient and robust global search methods thus become essential.

✉ Huachao Dong
dragon4562@126.com

¹ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

² Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada

The classical genetic algorithm (GA) global optimization method usually needs thousands of evaluations of the objective and constraint functions, and each of these evaluations, if conducted using complex simulation and analysis, will take minutes or hours to complete, making these classical global optimization method unsuitable as a design tool. One effective way to reduce the number of the expensive function evaluations in global optimization is to employ the more efficient surrogate-based optimization (SBO) technology (Queipo et al. 2005; Forrester and Keane 2009). SBO takes advantage of the prediction capacity of surrogate models (SM), or metamodels, to avoid redundant evaluations of the expensive functions and to focus on the most promising region of the global optimum (Ong et al. 2003; Zhou et al. 2007; Jouhaud et al. 2007). Generally, the SBO search process consists of three main steps: a) select sample points using design of experiments (DOE); b) construct a SM; and c) resample promising points to refine the SM. To get a better initial SM with fewer sample points, a variety of DOE methods have been introduced, including grid sampling, Latin Hypercube Sampling (LHS) (Iman 2008), Optimized Latin Hypercube Sampling (OLHS) (Jin et al. 2005), etc. Among them, grid sampling produces uniformly distributed points in the design space; and LHS/OLHS usually yields sample locations with some randomness. SMs involve interpolation and regression to envisage the black-box function. Common SM methods include kriging, radial basis functions (RBF), polynomial response surface (PRS), etc. Kriging is a stochastic model that utilizes the best linear unbiased estimator to predict the function values at untested locations (Cressie 1990). Composed of many simple functions, RBF establishes an expression to emulate the complicated design space (Mullur and Messac 2005). PRS constructs a polynomial model using least square fitting to make predictions (Myers et al. 2009). Among the three, kriging and RBF have been found to be more appropriate in dealing with non-linear problems (Forrester and Keane 2009). At the start of the search, the initial SMs of the objective/constraint functions with fewer initial sample points would not likely be able to locate the black-box problem's regions of optimum, especially for multimodal or high-dimensional problems. How to refine the SM with an adequate number of sample points is the key of the search technique, while to acquire the promising location or local optimum from the SM with a continuous mathematical expression subsequently using a regular optimization solver is a relatively straightforward task. In addition, it is normally difficult to solve multimodal and high-dimensional optimization problems only by supplementing the optimal solutions of the SM, and development of a generic infill strategy for a variety of optimization problems with fewer expensive function evaluations thus becomes critical.

2 Related work

Many researchers have contributed to the continuous developments of surrogate-based global optimization methods. Alexandrov et al. (1998) introduced a trust region approach to manage the use of SMs in engineering optimization. Jones et al. (1998) presented a widely cited global optimization algorithm for expensive black-box problems, which is known as EGO. EGO constructs the SM by kriging and updates the SM by maximizing an expected improvement function. Wang et al. (2001) provided an adaptive response surface method (ARSM), which creates a quadratic approximation model for the expensive objective function in a reduced space. Gutmann (2001) introduced a global optimization method based on RBF to solve problems with expensive function evaluations. Jin et al. (2001) explored the accuracy of SMs and how it affects the sampling strategies. Wang et al. (2004) developed the mode-pursuing sampling strategy (MPS) to refine the quadratic response surface. In the same year, Wang and Simpson (2004) utilized a fuzzy clustering method to get a reduced search space, which can efficiently find the global optimum on non-linear constrained optimization problems. A stochastic RBF method for the global optimization of expensive functions was proposed by Regis and Shoemaker (2007a), who also improved the Gutmann-RBF method by varying the size of the subdomain in different iterations (Regis and Shoemaker 2007b). Younis and Dong (2010) developed a kind of space reduction method called space exploration and unimodal region elimination (SEUMRE), which establishes a unimodal region to speed up the search. SEUMRE has been successfully used on several black-box automotive design optimization applications (Younis et al. 2011). Gu et al. (2012) introduced the hybrid and adaptive meta-model-based (HAM) method to divide the design space into several subdomains with different weights. In each iteration of the search, sample points are obtained from these sub-regions based on performance determined weights. HAM performed well in a FEA crash simulation based vehicle body design optimization problem. Villanueva et al. (2012) defined different sub-regions for the original design space and explored them using multiple SMs. Long et al. (2015) combined a type of intelligent space exploration strategy with ARSM to provide reduced regions for global optimization. These researches showed that the space reduction approach presents to be an efficiency global optimization search scheme for computationally expensive problems. In this work, a new multi-start approach is combined with a special space reduction strategy to form a new surrogate-based global optimization method. This multi-start approach provides multiple starting locations and utilizes local optimization solver to perform parallel searches in different sub-regions. The global optimal solution is identified from all obtained local optima. The general idea has been used in several global optimization algorithms, such as the GLOBAL algorithm (Csendes 1988),

the Tabu cutting method (Lasdon et al. 2010), and the quasi-multistart framework (Regis and Shoemaker 2013). In this work, the multi-start approach is found to be ideal for the kriging-based optimization method. In the global optimization, the initial kriging-based prediction model will normally not be very precise, leading to solutions with some local optima. When the SM becomes more accurate after the updates, one of the estimated optima may approach the true global optimum while most of the others will either disappear or go closer to the nearby local optima. If all predicted local optima in the entire design space are added to refine the SM, the number of sample points will become considerably large and the search efficiency will be lower.

In this new kriging-based multi-start optimization process, the three spaces, GS, MS and LS, with different ranges are explored alternately. The sizes of MS and LS change adaptively based on the current best obtained value in optimizing the kriging-based SM; selecting the promising results; and exploring the unknown area. Exploration of the design space at three different space levels facilitates the handling of the multimodal objective function. Bound and nonlinear constrained benchmark global optimization problems are used to test the search efficiency and robustness of this new algorithm.

3 Multi-start space reduction approach

The proposed MSSR approach uses a kriging-based SM, a multi-start scheme, and alternating sampling over the GS, MS and LS spaces to carry out the global optimization search. The local optima obtained from the SM are stored in a database as “potential sample points”. In each iteration, several better sample points are selected from these “potential sample points” to refine the SM to allow it to have a better approximation in the close neighborhood of the global optimum. The detailed search algorithm is introduced in this section.

3.1 Kriging-based model

Kriging is an interpolation method widely used for prediction. The kriging predictor and its estimated mean square error (MSE) function have the following forms:

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \quad (1)$$

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

Given M sample points, \mathbf{R} is a correlation matrix with $M \times M$ elements and the element in the j -th column of the i -th row is $C(\Theta, \mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ that is a correlation function value. $\mathbf{r}(\mathbf{x})$ is an M -dimensional vector and $C(\Theta, \mathbf{x}, \mathbf{x}^{(i)})$ is the i -th element

of $\mathbf{r}(\mathbf{x})$, where \mathbf{x} is the untested location. The values of $\hat{\mu}$, $\hat{\sigma}^2$, Θ are estimated using the maximum likelihood estimation (MLE) method and its detailed derivation can be found from (Jones et al. 1998; Forrester and Keane 2009).

Since $\hat{f}(\mathbf{x})$ is the approximation of the true function $f(\mathbf{x})$, $\hat{f}(\mathbf{x})$ can be exploited to estimate the true optimum. $\hat{s}^2(\mathbf{x})$ reflects the uncertainty of the predictor $\hat{f}(\mathbf{x})$. When the tested location \mathbf{x} is far away from the known points, $\hat{s}^2(\mathbf{x})$ will have a large value, indicating a large prediction error. The magnitude of $\hat{s}^2(\mathbf{x})$ can thus be used to guide the exploration of the unknown region. The global optimization search process for the benchmark Banana function (Fig. 1) is used to show the kriging prediction scheme graphically. Fifteen sample points are used to construct the initial kriging model shown in Fig. 2 to capture the rough shape of the original function illustrated in Fig. 1. As the number of sample points increases, the predicted shape of the model will better approximate the true shape of the banana function in Fig. 1. The estimated MSE function of the kriging model is given in Fig. 3, and the local maxima of the MSE always appear in the unexplored area. As Forrester has suggested that the kriging predictor can be exploited to acquire promising solutions and the unknown design space can be explored by the estimated MSE of kriging. An optimization search process that can effectively utilize these two properties of the kriging model will be able to take the full advantage of the kriging method (Forrester and Keane 2009)

3.2 The proposed multi-start optimization process

The proposed multi-start optimization process for kriging-based model consists of three parts: local optimization using SM, selection of better potential sample points, and exploration of the unknown area.

To obtain the randomly selected starting points that also well cover the search space, the LHS is used.

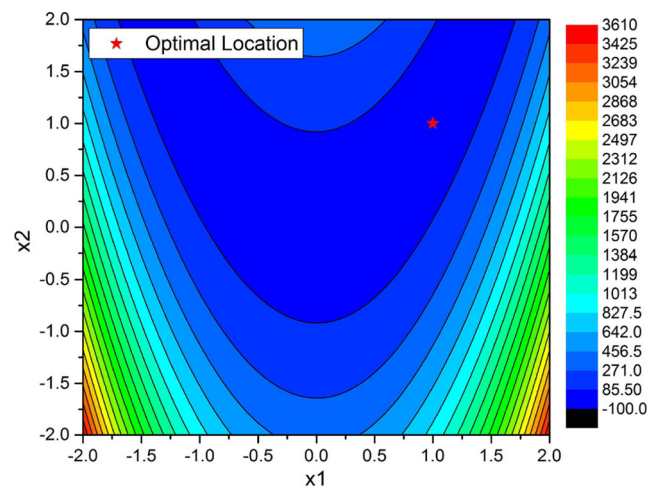


Fig. 1 Original Banana function

These different starting points will be used in each iteration during the iterative search process. The local optima from the kriging SM are identified using Sequential Quadratic Programming (SQP). All of the local optima obtained from the SM are stored in a database of “Potential Sample Points”. The SQP optimizers may sometimes converge to the same local optimum from different starting points, leading to repeated points in the database, and the local optimum may also be found at the existing sample points. To address these issues, new sample points need to have a defined dis-

tance from all obtained sample points. In other cases, there may not be any suitable local optimum in the defined space. For these occasions, the estimated MSE of kriging is maximized by the multi-start optimization process to explore the unknown area. As shown in details in the following pseudo codes a special sample point selection scheme has been introduced to identify more promising sample points from the acquired “potential sample points” list. These codes including optimization, selection and exploration on unknown area are summarized as follows.

Optimization:

Begin

Initialize Dimension n , Database “potential samples”, Design Space, Kriging Predictor, MSE;

Acquire m starting points by LHS; (Here, it is suggested that m can be defined in the range [20, 40] on two-dimensional problems, [6 n , 8 n] when the dimension of the problem is $2 < n < 10$, and [50, 70] on high-dimensional problems.)

for $i=1:m$

Employ SQP algorithm;

Optimize the Kriging Predictor from the i th starting point;

Store the local optimal solutions and its predicted values in the database “Potential Samples”;

end

“potential samples” is a matrix with $m \times (n + 1)$ elements;

end

(The design space is selected among GS MS and LS, which will change with the iteration going on. The kriging predictor and its estimated MSE can be obtained by DACE toolbox (Lophaven et al. 2002). The “fmincon” function of MATLAB can be employed to realize the SQP algorithm (The Mathworks 2015).)

Selection:**Begin**

Sort the predicted values in “potential samples” and get the maximum (Xps_{max} Yps_{max}) and minimum (Xps_{min} , Yps_{min}); (The sample and the predicted value in “potential samples” are expressed as (Xps , Yps))

Initialize parameters $k = 1$, $flag_repeat = 0$, $flag_stop = 0$, $e_error = 0.00001$ (If $n \geq 10$, $e_error = 0.0001$), $MAXK$; ($MAXK$ is a parameter that decides how many points can be sampled at most in one iteration, Here, $MAXK$ equals to 3 on two-dimensional problems and equals to 4 on higher-dimensional problems. For nonlinear constrained optimization problems, $MAXK$ equals to 3.)

Acquire the size of the expensive samples set S as m_size ;

While $k < MAXK$ and $flag_stop == 0$

for $i = 1 : m_size$

if square of the distance between Xps_{min} and the sample $S(i) \leq e_error$

$flag_repeat = 1$;

end

end

(Here, the new promising samples that go much close to the existing points will be flagged.)

if $flag_repeat == 0$

record the current sample Xps_{min} ; $k = k + 1$;

end

for $i = 1 : m$

if $|Yps(i) - Yps_{min}| \leq 0.0001$

$Yps(i) = Yps_{max} + 10$;

end

end

(At each iteration, just one local optimal solution from the kriging model is selected and the other same results are covered by a big value “ $Yps_{max} + 10$ ”. When the next iteration comes, the bigger values are ignored.)

```

Sort the predicted values  $Yps$  in “potential samples” again and update ( $Xpsmin$ ,  $Ypsmin$ );
If  $Ypsmin == Ypsmax + 10$ 
     $flag\_stop = 1$ 
end
 $flag\_repeat = 0$ ;
end
if  $k > 1$ 
    Store the selected samples and evaluate the true function values.
end
end

```

Explore Unknown Area:

```

Begin
if  $k == 1$ 
    Implement the above-mentioned Optimization method to get the local maxima of the MSE function.
    Get two new samples and evaluate the true function values.
end
end

```

(If the algorithm cannot find a satisfactory solution by the above-mentioned selection process, the estimated MSE can be maximized to acquire new samples which must be located in an unexplored area.)

Figure 4 shows the multi-start optimization process on the kriging model. Thirty starting points and two local optima were selected. The two selected solutions were in the valley of the Banana function, in which the best solution is located as shown in Fig. 1. Addition of these two sample points in this local area will make the kriging model more accurate locally.

3.3 Space reduction approach

A sample set, acquired using the DOE method is used to store the expensive data samples. According to the

magnitude of the function values of these expensive sample points, three spaces, GS, MS and LS are created for the multi-start optimization. Global Space or GS is the entire region of the original design space; Medium-sized Space or MS is based on the portion of design space of the current better samples; and Local Space or LS is the neighborhood area of the best current sample point. In the iterative search process, the sample set is supplemented with new samples, and updated using better samples. MS and LS will constantly change until the iteration stops. The definitions of the two spaces are as follows:

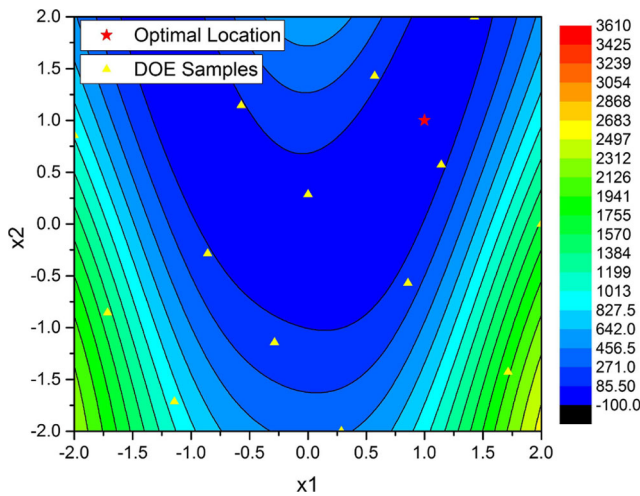


Fig. 2 Kriging prediction with 15 samples

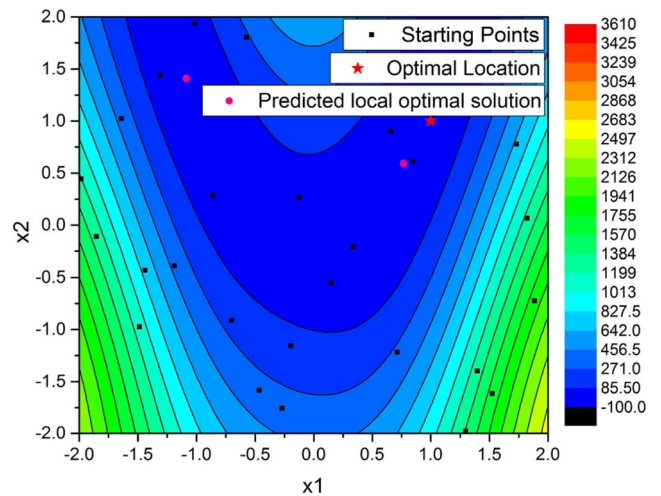


Fig. 4 Multi-start process on kriging

$$\begin{aligned}
 dis_i &= |\max(\mathcal{S}(1:k)_i) - \min(\mathcal{S}(1:k)_i)|, \quad i = 1, 2, \dots, n \\
 Lob_i &= \begin{cases} S_i^{best} - dis_i, & S_i^{best} - dis_i \geq \min(range_i) \\ \min(range_i), & S_i^{best} - dis_i < \min(range_i) \end{cases} \\
 Ub_i &= \begin{cases} S_i^{best} + dis_i, & S_i^{best} + dis_i \leq \max(range_i) \\ \max(range_i), & S_i^{best} + dis_i > \max(range_i) \end{cases} \\
 range_local_i &= [Lob_i, Ub_i]
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 Lob_i &= \min(\mathcal{S}(1:p)_i) \\
 Ub_i &= \max(\mathcal{S}(1:p)_i) \\
 range_medium_i &= [Lob_i, Ub_i], \quad i = 1, 2, \dots, n
 \end{aligned} \tag{4}$$

where, n is the dimension of a problem, $\mathcal{S}(1:k)_i$ is the i -th dimension of the first k best samples selected from the ranked sample set, $range_i$ is the i -th dimension of the original design range, and S_i^{best} is the i -th dimension of the current best sample. Formulas (3) and (4) define the Local and Medium-sized space, respectively. If the distance of Lob_i and Ub_i in Formulas (3) or (4) is smaller than $1e-5$, it is suggested that

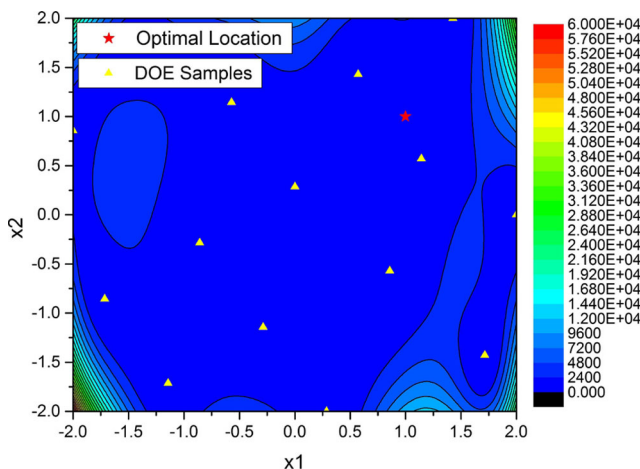


Fig. 3 Estimated MSE of kriging

$Lob_i = Lob_i - 0.025 \cdot (\max(range_i) - \min(range_i))$ and $Ub_i = Ub_i + 0.025 \cdot (\max(range_i) - \min(range_i))$. Meanwhile, the new range should also be the subset of the original design range. Both of the two spaces change their scopes based on the better samples acquired from the design space. Here, k and p are two user-defined parameters,

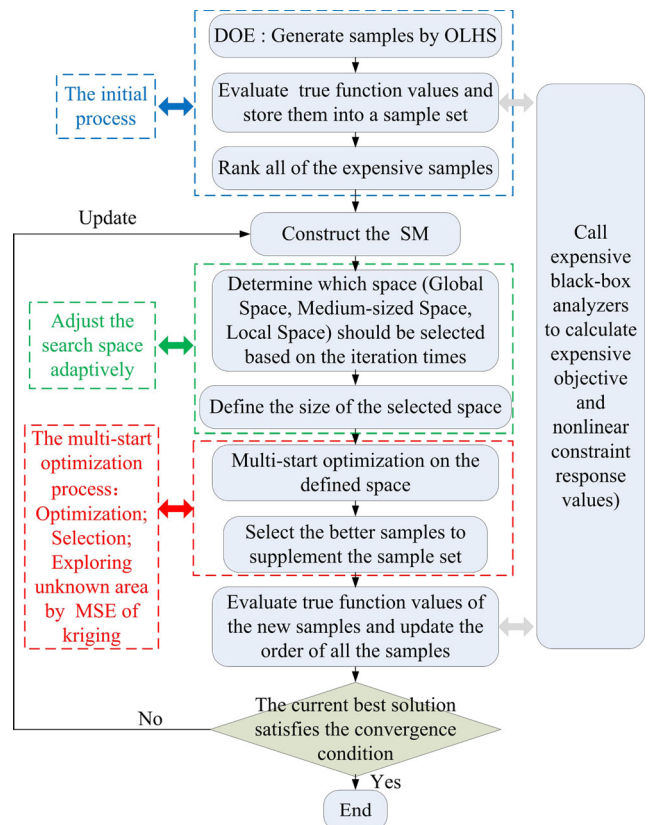


Fig. 5 Flowchart of the MSSR optimization process

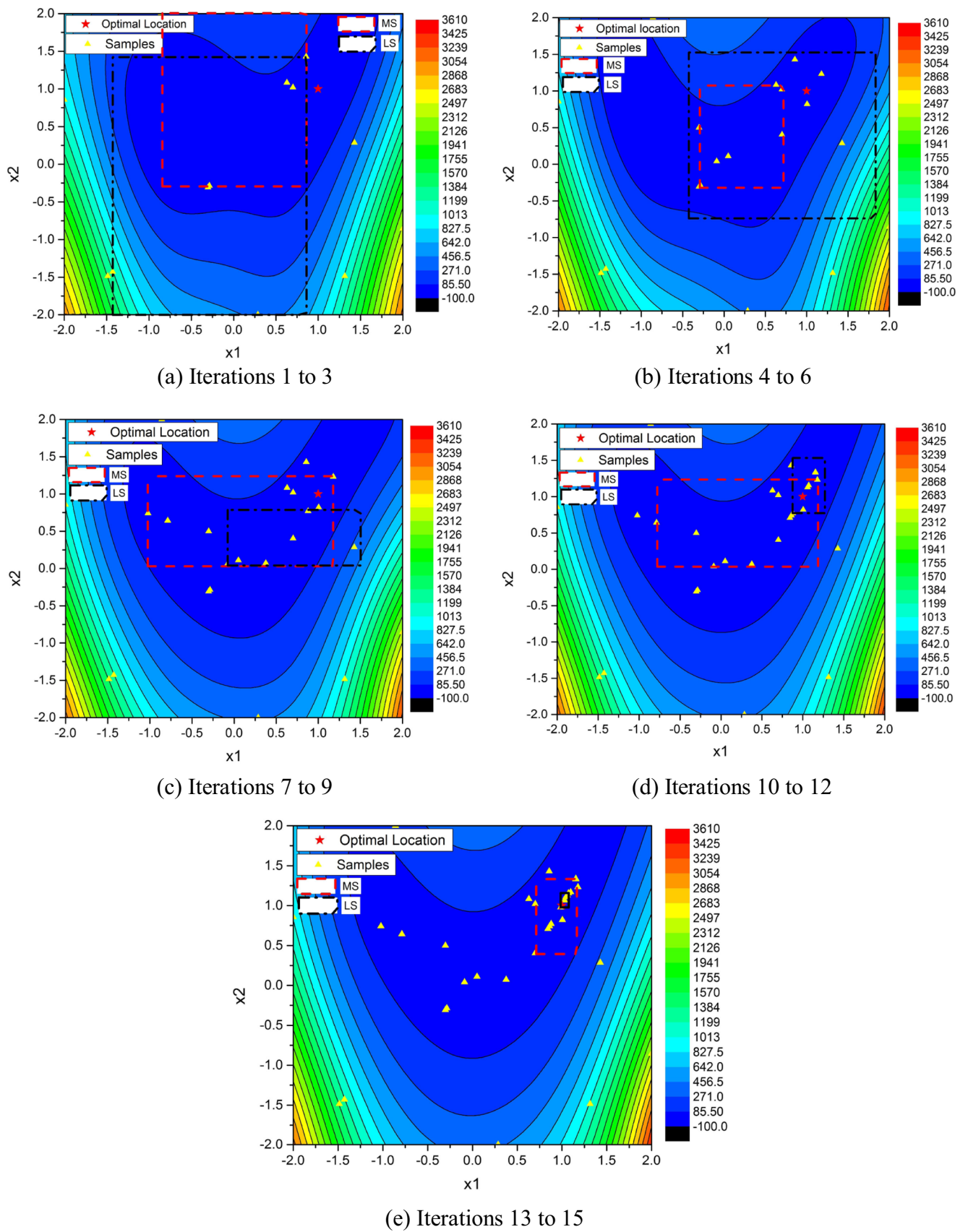


Fig. 6 MSSR optimization process on Benchmark Banana function

which represent the number of the better samples. We define k and p as follows:

$$k = \begin{cases} 3 & , n \leq 2 \text{ and } CS \leq 150 \\ \text{round}(CS/30) & , n \leq 2 \text{ and } CS > 150 \\ 5 & , n > 2 \text{ and } CS \leq 150 \\ \text{round}(CS/30) & , n > 2 \text{ and } CS > 150 \end{cases} \quad (5)$$

$$p = \begin{cases} \text{round}(CS/3) & , n \leq 2 \\ 3n & , n > 2 \text{ and } CS \leq 60 \\ \text{round}(CS/3) & , n > 2 \text{ and } CS > 60 \end{cases} \quad (6)$$

where, CS is the number of current sample points. k will be smaller than p with continuous iterations. According to (3) to (6), MS will lead to a reduced region that may include several promising solutions and LS will make the search focus on one of them quickly. In some cases, when LS turns into a tiny space or the search in LS, MS or GS repeats around a local optimum, there will be no appropriate locations to be selected as new samples. As discussed previously in Section 3.2, if new samples cannot be found after the optimization and selection, the estimated MSE of kriging is used to explore the unknown area. The ranges for getting the local maxima of MSE in local, medium and global search are defined as follows:

$$\begin{aligned} dis_i &= |max(range_i) - min(range_i)|, \quad i = 1, 2, \dots, n \\ Lob_i &= \begin{cases} S_i^{best} - 0.5 \cdot dis_i, & S_i^{best} - 0.5 \cdot dis_i \geq min(range_i) \\ min(range_i), & S_i^{best} - 0.5 \cdot dis_i < min(range_i) \end{cases} \\ Ub_i &= \begin{cases} S_i^{best} + 0.5 \cdot dis_i, & S_i^{best} + 0.5 \cdot dis_i \leq max(range_i) \\ max(range_i), & S_i^{best} + 0.5 \cdot dis_i > max(range_i) \end{cases} \\ range_mse_local_i &= [Lob_i, Ub_i] \\ range_mse_medium_i &= range_medium_i \\ range_mse_global_i &= range_i \end{aligned} \quad (7)$$

Intuitively, these defined ranges enclose the current best solution and they may move with the continuous iterative search. On the one hand, the algorithm combines GS, MS and LS to sufficiently exploit the kriging predictor and accelerate the convergence to the global optimum. On the other hand, the algorithm explores the unknown area to allow the current best solution to jump out of the local area.

3.4 The entire optimization process

The complete multi-start space reduction (MSSR) global optimization process is summarized into the following steps, and illustrated by the flowchart in Fig. 5.

The initial process

- (1) Design of experiment: apply OLHS (Jin et al. 2005) to generate DOE sample points over the entire design space.
- (2) Evaluate the expensive function using the DOE sample points and store the results in the sample set. (For non-linear constrained problems, expensive functions include objective and constraint functions.)
- (3) Rank all expensive samples based on their function values. (Here, if a sample point does not satisfy the constraints, the sample values should add a large penalty factor of 1E6.)

The search loop

- (4) Construct the kriging-based SM. (For nonlinear constrained problems, SMs of objective and constraint functions are built, respectively. Here, sample values use the true objective values without the additional penalty factor.)
- (5) Determine which space should be explored based on the present number of iterations. The global search, medium-sized search and local search will be implemented alternately in the process.
- (6) Define the size of the search space, according to the expensive sample set.
- (7) Use the chosen multi-start local optimization approach, SQP, to optimize the kriging-based SM in the defined space.
- (8) Store the obtained local optimal solutions in the database “potential sample points” and select better samples. If there is no better samples, select two new samples from the unknown area.
- (9) Evaluate the expensive function at the selected sample points and update the order of the expensive samples as in step (3).
- (10) If the current best sample value satisfies the stopping criteria, terminate the loop. Otherwise, update the SM and repeat the steps (4) to (9) until the global stopping criteria are satisfied.

The commonly used global stopping criteria are:

$$\begin{cases} \frac{|y_{best} - y_{optimal}|}{|y_{optimal}|} < 1\% & \text{if } y_{optimal} \neq 0 \\ y_{best} < 0.001 & \text{if } y_{optimal} = 0 \end{cases} \quad (8)$$

To better demonstrate the MSSR search process, generations and updates of sample points during global optimization on the Banana function are illustrated in Figs. 6a to e. The multi-start optimization process presented in

Table 1 Bound constrained benchmark problems for global optimization

Category	Func.	Number of dims.	Design space	Analytic global minimum
Low-dimensional problems ($n = 2-6$)	Banana	2	$[-2, 2]^2$	0.0000
	Peaks	2	$[-3, 3] \times [-4, 4]$	-6.5511
	GP	2	$[-2, 2]^2$	3.0000
	SC	2	$[-2, 2]^2$	-1.0320
	Shub	2	$[-10, 10]^2$	-186.7309
	GF	2	$[-2, 2]^2$	0.5233
	HM	2	$[-6, 6]^2$	0.0000
	Leon	2	$[-10, 10]^2$	0.0000
	Shekel	4	$[0, 10]^4$	-10.1532
	Levy	4	$[-10, 10]^4$	0.0000
	HN6	6	$[0, 1]^6$	-3.3220
High-dimensional problems ($n \geq 10$)	Trid6	6	$[-36, 36]^6$	-50.0000
	Sphere	10	$[-5.12, 5.12]^{10}$	0.0000
	Trid10	10	$[-100, 100]^{10}$	-210.0000
	F16	16	$[-1, 1]^{16}$	25.8750

Section 3.2 is used to explore the three defined spaces. Each figure shows three iterations of the searches in GS, MS and LS. At the start, the region of LS is larger than that of MS. As the iteration goes on and the number of expensive sample points increases, LS quickly shrinks to the local region around the global optimum. MS always provides a medium-sized region which includes the current best solution. Both MS and LS become smaller and smaller when more and more sample points are added. Intuitively, shrinking LS makes the search concentrate on more promising region and accelerate the convergence;

MS provides a promising region which may include several local optima; and GS ensures that the multi-start optimization process explores the entire design space. As shown in Figs. 6a and c, LS may not include the true global optimum at some time, but it will follow the moving best sample point and eventually move close to the global optimum. In this example, 15 iterations and 37 expensive sample points are needed to find the global optimum. During the search process, the general shape of the SM at different stages, as shown in Figs. 6a to e, gradually approached the shape of the true objective

Table 2 Preliminary comparison results on seven representative benchmark functions

Algorithms		Banana	GP	SC	Shub	Shekel	HN6	F16
HS	NFE	9122	512	310	450	10000	698	915
	Min	8.84e-4	3.0164	-1.0276	-185.6736	-2.6829	-3.3033	26.1207
DE	NFE	1390	830	450	3070	3730	3660	3690
	Min	4.05e-4	3.0075	-1.0299	-185.3988	-10.0930	-3.3085	26.1022
DIRECT	NFE	603	101	117	2883	103	213	6439
	Min	3.01e-4	3.0073	-1.0248	-185.5823	-10.0934	-3.2975	26.0884
MPS	NFE	145	134	35	545	680	783	3319
	Min	0.0358	3.0014	-1.0311	-186.7119	-5.0473	-3.3205	29.7177
EGO	NFE	216	167	35	227	250	54	200
	Min	9.67e-4	3.0323	-1.0297	-181.0324	-7.5345	-3.3152	27.4815
MS	NFE	61	124	25	117	289	121	161
	Min	2.51e-4	3.0065	-1.0299	-186.4286	-10.0863	-3.2973	26.1116
MSSR	NFE	41	82	22	115	197	83	138
	Min	3.45e-4	3.0049	-1.0303	-186.4203	-10.0829	-3.2967	26.1257

Table 3 Mean values of NFE and ranges of optimal values obtained by the three algorithms

Func.	MSSR		SEUMRE		HAM	
	NFE	Obtained value	NFE	Obtained value	NFE	Obtained value
Banana	42.8	[1.91e-5, 7.32e-4]	90.9	[4.75e-5, 6.37e-4]	68.3	[1.27e-5, 6.34e-4]
Peaks	28.1	[-6.5477, -6.5007]	42.7	[-6.5509, -6.4868]	>228.5	[-6.5510, -3.0498]
GP	87.1	[3.0001, 3.0273]	133.6	[3.0002, 3.0191]	122	[3.0001, 3.0227]
SC	22.5	[-1.0316, -1.0274]	48.8	[-1.0307, -1.0241]	33.9	[-1.0316, -1.0259]
Shub	122.9	[-186.7259, -184.9656]	>329.5	[-186.4404, -117.0721]	168.4	[-186.7209, -185.9839]
GF	34.2	[0.5233, 0.5277]	>208.4	[0.5259, 0.5350]	94.1	[0.5238, 0.5283]
HM	40.3	[7.79e-5, 7.56e-4]	>266.8	[1.04e-5, 0.0028]	120	[1.01e-4, 9.08e-4]
Leon	181.7	[8.88e-5, 9.68e-4]	>253.9	[1.12e-4, 0.3207]	239.4	[1.21e-4, 9.58e-4]
Shekel	207.1	[-10.1486, -10.0716]	>471.7	[-10.0546, -2.6303]	>458.1	[-10.1472, -2.6166]
Levy	218.5	[3.96e-4, 8.04e-4]	>358.1	[6.63e-4, 0.1103]	>341.7	[2.96e-5, 2.26e-2]
HN6	84.8	[-3.3119, -3.2890]	>282.5	[-3.3009, -3.1046]	93.5	[-3.3194, -3.2967]
Trid6	92.1	[-49.9021, -49.5544]	>500	[-47.5255, -7.9626]	127.5	[-49.9614, -49.6379]
Sphere	115.4	[4.57e-4, 9.98e-4]	>500	[1.8147, 17.2568]	>288.3	[4.20e-4, 0.1847]
Trid10	142.4	[-208.9614, -208.0416]	>500	[-83.0087, 990.0295]	>500	[-166.6914, -48.9592]
F16	137.7	[26.1053, 26.1307]	>500	[27.5243, 29.5178]	>249.8	[26.0410, 26.6333]

function as shown in Fig. 1, and the SM became sufficiently precise to the objective function in the local region around the global optimum.

4 Test cases and results

To verify the capability and to demonstrate the advantage of the new MSSR algorithm, a number of commonly-used, constrained global optimization benchmark

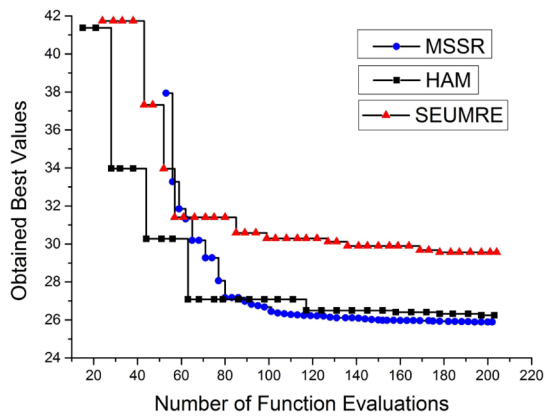
problems with 2 to 16 design variables were used as test examples. These included eight two-dimensional problems (Banana, Peaks, GP, SC, Shub, GF, HM, Leon), two four-dimensional problems (Shekel and Levy), two six-dimensional problems (HN6 and Trid6), two ten-dimensional problems (Sphere and Trid10), and one sixteen-dimensional problem F16 (Wang et al. 2004; Younis and Dong 2010; Gu et al. 2012) with variable bound constraints. These test problems have their own unique characteristics, and in combination they can better

Table 4 Specific statistical results of NFE obtained by MSSR, SEUMRE and HAM

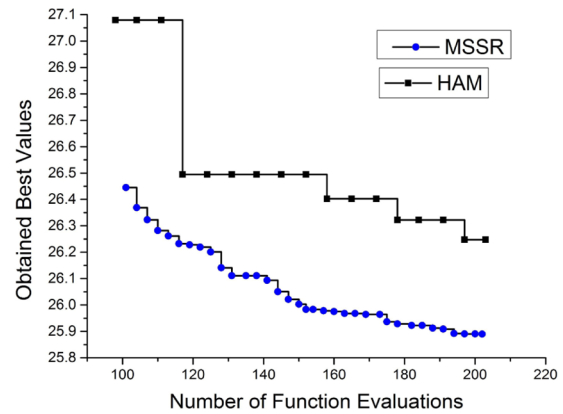
Func.	MSSR			SEUMRE			HAM		
	Min	Max	Median	Min	Max	Median	Min	Max	Median
Banana	24	66	41	72	114	86	45	104	62
Peaks	18	50	24	37	44	44	34	>500(4)	73
GP	51	141	82	79	359	93	82	195	117
SC	18	27	22	44	58	49	26	52	29
Shub	24	215	115	68	>500(3)	377	86	315	160
GF	15	64	29	65	>500(2)	100	46	281	76
HM	22	95	32	65	>500(4)	157	46	288	66
Leon	67	408	146	142	>500(2)	194	102	433	233
Shekel	68	415	197	217	>500(9)	>500	269	>500(8)	>500
Levy	89	376	181	119	>500(6)	>500	104	>500(5)	>370
HN6	52	117	83	125	>500(4)	149	87	108	91
Trid6	63	146	85	>500	>500(10)	>500	106	144	130
Sphere	94	145	117	>500	>500(10)	>500	180	>500(3)	198
Trid10	125	162	139	>500	>500(10)	>500	>500	>500(10)	>500
F16	103	168	138	>500	>500(10)	>500	136	>500(2)	201

represent many different situations in design optimization. The detailed forms of these functions are given in Table 1. For optimization problems with nonlinear constraints, two mathematical functions and four benchmark design optimization cases were used in the testing. All of these test cases are given in Appendix. Ten runs on each of these

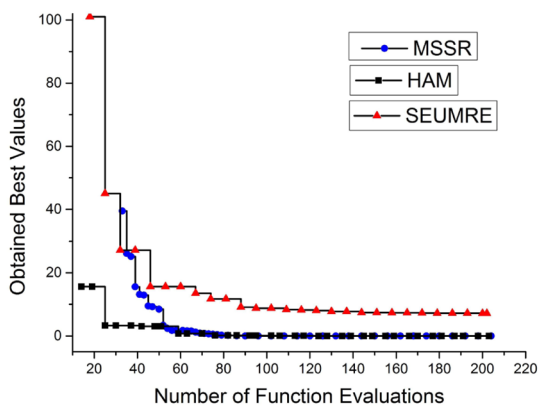
benchmark problems have been made using the new MSSR search program. The obtained statistical results were compared with the results from the other recently introduced space reduction search methods for global optimization to compare their search efficiency and robustness.



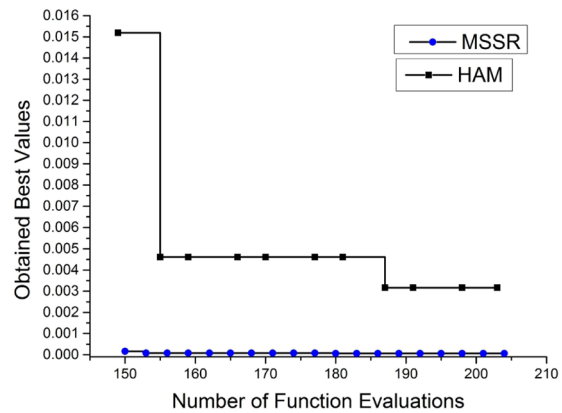
(a) Iterative results on F16



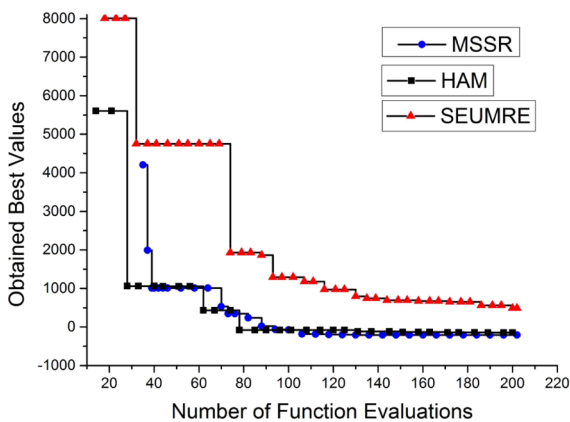
(b) Clearer iterative results on F16



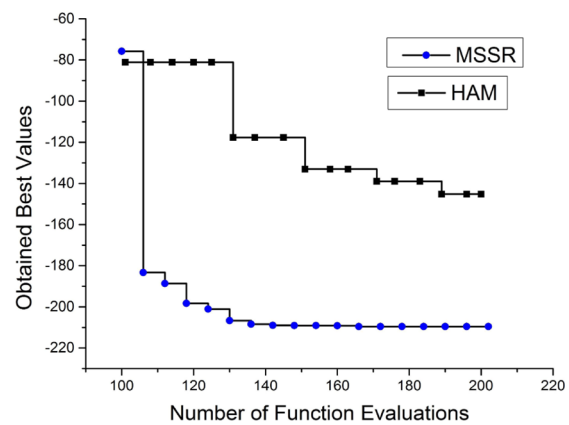
(c) Iterative results on Sphere



(d) Clearer iterative results on Sphere



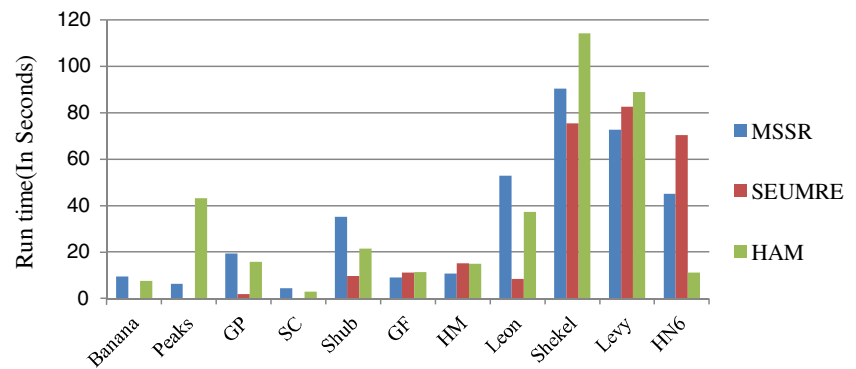
(e) Iterative results on Trid10



(f) Clearer iterative results on Trid10

Fig. 7 Iterative results on high-dimensional problems

Fig. 8 Execution time of MSSR, SEUMRE and HAM on benchmark functions



4.1 Bound constrained optimization problems

Firstly Harmony Search (HS, Yang 2010) and Differential Evolution (DE, Storn and Price 1997) algorithms, as examples of nature-inspired global optimization methods, were selected as references of high computation cost for the solution of optimization problem with expensive black-box functions. Secondly an effective space reduction algorithm DIRECT (Björkman and Holmström 1999) and a widely cited surrogate-based space exploration method MPS (Wang et al. 2004) were used as comparison. In addition, the EGO algorithm that uses the kriging model for expensive functions was compared to show the advantage of the algorithm, and the Mueller’s surrogate model toolbox (Mueller 2012) was used to carry out the “expected improvement” strategy in the EGO algorithm. Furthermore, the presented Multi-start (MS) optimization algorithm without using the “space reduction” strategy was also tested as another comparison. At last, the MSSR algorithm has been tested using the same benchmark problems and $3n+2$ DOE sample points have been generated to construct the initial SM during the tests.

Seven representative functions from Table 1 were used as test cases, and the seven different algorithms for comparison have run these tests for 10 times to get the statistical results. The collected median values of the number of function evaluations (NFE) and the obtained minimum values from the optimization are given in Table 2. The seven algorithms tried to get the values that satisfy the condition of Formula (8).

Since the EGO method requires much higher CPU time than the other algorithms when the dimension of the problem increases, a maximum allowable NFE (250 for low-dimensional problem and 200 for high-dimensional problem) was used during its tests. The test results in Table 2 have indicated:

- HS and DE consistently required higher NFE than the other algorithms;
- DIRECT performed well on most cases except the Banana, Shub and F16 functions;
- EGO and MPS could easily approach the global optimum for simpler cases like GP and SC, but they generally needed higher NFE for complex cases like Shub, Shekel and F16; and
- The proposed MS and MSSR algorithms showed better performance on all test cases, and MSSR used fewer NFE than the MS. This showed that the “space reduction” strategy did improve the multi-start optimization algorithm.

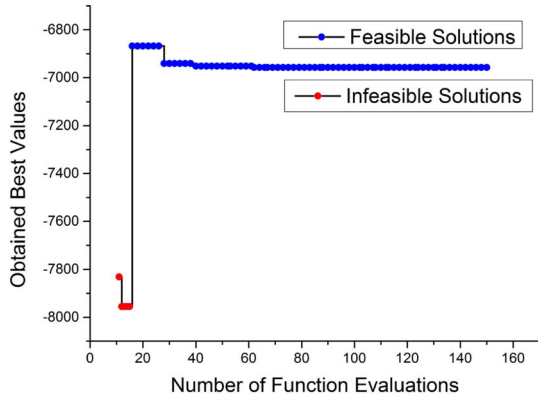
In summary, non-surrogate based search methods generally require considerably more NFE since these algorithms directly call the original objective function during the search, and the surrogate-based search methods use predictive models to guide the exploration of the design space, effectively decreasing the NFE. These preliminary comparisons with the nature-inspired global optimization methods and existing SBO methods showed the advantages of the new MSSR method on optimization problems with intensive computation due to the lower NFE required.

Table 5 Global optimal results obtained by MSSR on nonlinear constrained problems

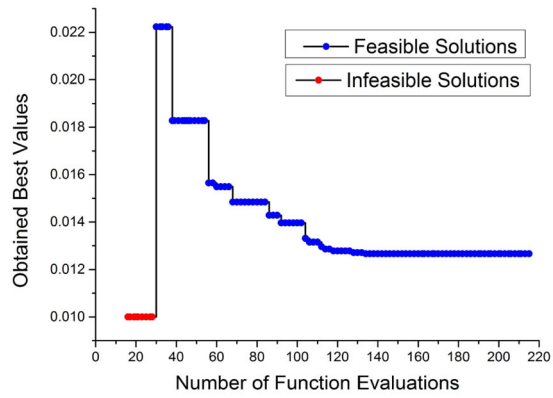
Problems	Design variables							$f(x)$
	x1	x2	x3	x4	x5	x6	x7	
G6	14.097149	0.847352						-6956.8719
TSD	0.0516827	0.3565636	11.2980133					0.0126652
WBD	0.2056902	3.4683028	9.0445203	0.2056904				1.7256
PVD	0.778187	0.384658	40.320586	199.986548				5885.3653
Him	78.000000	33.000000	27.072136	45.000000	44.967954			-31025.3139
SRD	3.500177	0.700000	17.000000	7.332558	7.715387	3.350284	5.286657	2994.8487

To further compare the search efficiency and robustness of the algorithms, MSSR and other two recently introduced surrogate-based global optimization algorithms (HAM and SEUMRE) were tested using the 15 functions listed in Table 1. SEUMRE utilizes grid sampling to divide the design space into several unimodal regions. Once the key unimodal region is obtained, a kriging model is built using sampled data. The optimization and reconstruction of the kriging model are mainly carried out in this key region. However, the initial

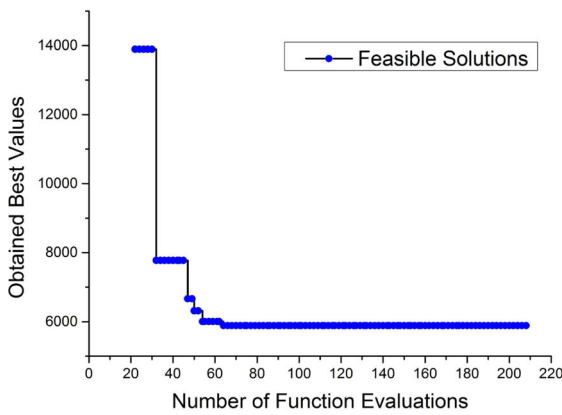
interpretation of the experiment data may miss the location of the true global optimum. The SEUMRE algorithm may thus be trapped around a local optimum for multimodal functions and/or high-dimensional problems. Although SEUMRE is highly efficient in search convergence, it is not stable for complex global optimization problems. HAM divides the original design space into several sub-spaces, applies three types of SMs, Kriging, RBF and PRS, to start the search, and acquires supplementary sample points using the better performing SM to



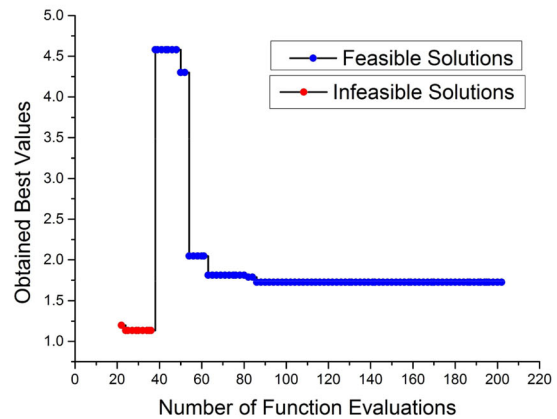
(a) Iterative results on G6



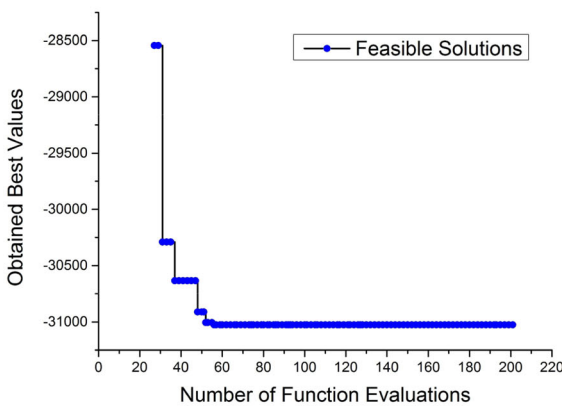
(b) Iterative results on TSD



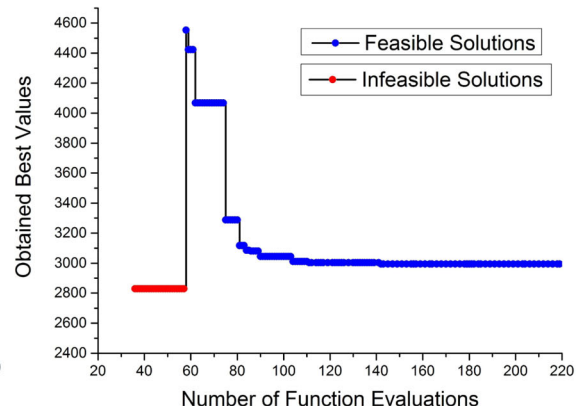
(c) Iterative results on PVD



(d) Iterative results on WBD



(e) Iterative results on Him



(f) Iterative results on SRD

Fig. 9 Iterative results obtained by MSSR on constrained optimization problems

adapt to the objective function on hand. The unique strategy effectively combines the predictive capacities of multiple SMs. However, once these SMs focus on the same region, the algorithm will likely converge to the local optimum and can hardly explore other promising areas. In this comparing test, the same termination criteria of Formula (8) and user-defined parameters in (Gu et al. 2012; Younis and Dong 2010) were used.

Since the grid sampling method may find the global optimum of the GP and Banana functions by chance before the iteration process of SEUMRE begins, the DOE range of the algorithm was changed to 95 % of the original range on the two problems. To deal with the randomness of these methods, ten experiment tests were made for each case. The mean number of NFEs and the range of the obtained best values from the searches are shown in Table 3. The statistical results of NFE with minimum NFE, maximum NFE and the median value are given in Table 4. The NFE values with the > sign indicate that at least one of the tests could not satisfy the stopping criteria within 500 function evaluations, and the numbers in the brackets represent how many failed searches it had. These results showed that the new MSSR method has successfully identified the global optimum in all test cases within 500 function evaluations and required the least number of NFE. The SEUMRE algorithm performed well on the Banana, Peaks, GP and SC functions, but it had difficulties in solving the multimodal and high-dimensional problems. Specifically, SEUMRE succeeded one time on Shekel, four times on Levy and six times on HN6 functions, but it failed all the ten runs on the Trid6, Sphere, Trid10 and F16 functions. The best value that the SEUMRE has obtained on F16 function is 27.524, which is much larger than the results from the MSSR and HAM methods.

HAM is an effective search method that performed well on Banana, GP, SC, Shub, GF, HM, Leon, HN6 and Trid6 functions, but it showed poor performance on Shekel and Trid10. For high-dimensional problems Sphere and F16, HAM could get satisfactory solutions most of time.

A comparison on calculated objective function values and required NFEs of the three methods for high-dimensional problems is shown in Figs. 7a–f. In order to improve the readability, two adjacent dots keep a small interval that is roughly greater than 2 units of NFE. Results from the entire search process within 200 function evaluations are shown in Figs. 7a, c and e, while closer views on the results of HAM and MSSR from 100 to 200 NFE are given in Figs. 7b, d and f. The new MSSR algorithm has been found to be able to converge to the optimum much quicker than HAM and SEUMRE, and meet the given stopping criteria within 200 objective function evaluations. The computation time needed for the three methods to find the optimum using a PC with Core i7-4720HQ CPU (2.60 GHZ) and 16GB memory is given in Fig. 8. The MSSR and HAM methods used more time on

two-dimensional problems since MSSR needed to call the SQP solver many times and HAM needed to construct three SMs in each iteration. All three methods require more computation time for higher-dimensional and multimodal problems.

As the most important indicator for an algorithm's search efficiency and computation time need for computationally expensive black-box problems, the results of required NFEs for different test problems put the computation performance of the three methods into perspective. The HAM method showed good performance most of the time, but it may sometimes be trapped into local optima. The SEUMRE method performed better on low-dimensional problems, but it could not function well on multimodal and high-dimensional optimization problems. The new MSSR method needed the least number of function evaluations and appeared to be most robust, presenting to be the most promising black-box global optimization technique.

4.2 Nonlinear constrained engineering applications

The new MSSR search methods have also been tested using six other nonlinear constrained optimization benchmark problems. These comprised the G6 function that came from the constrained optimization problems used by Hedar (2004), Egea (2008), Younis and Dong (2010); the widely used Himmelblau's nonlinear problems (Him, Himmelblau 1972) (Gen and Cheng 1997); and four other structural design applications, including the Tension/Compression Spring Design (TSD), Welded Beam Design (WBD), Pressure Vessel Design (PVD), and Speed Reducer Design (SRD) (Coello 2002; Ao and Chi 2010; Garg 2014). All of these six test problems have their objectives and constraints in computation expensive black-box form. These test cases (G6, TSD, WBD, PVD, Him, SRD) have dimensions range from 2 to 7 with 2, 4, 7, 4, 6 and 11 constraints, respectively. The test results are

Table 6 Summary of results obtained by MSSR on G6, TSD and Him

Exp.	G6		TSD		Him	
	NFE	Opt. value	NFE	Opt. value	NFE	Opt. value
No.1	62	-6957.3896	81	0.0126664	60	-31025.5575
No.2	79	-6958.4628	213	0.0126817	61	-31025.2482
No.3	19	-6955.8152	139	0.0126817	48	-31025.5270
No.4	44	-6958.2769	97	0.0126654	93	-31025.0141
No.5	20	-6958.2769	97	0.0126653	100	-31021.3633
No.6	53	-6957.8394	140	0.0126655	69	-31023.6350
No.7	39	-6958.0899	114	0.0126670	57	-31023.9933
No.8	40	-6961.2597	66	0.0126698	63	-31025.5595
No.9	29	-6955.2008	109	0.0126654	51	-31025.5557
No.10	63	-6955.2106	108	0.0126652	51	-31023.2053

given in Table 5 and the iterative processes are illustrated in Fig. 9. The new MSSR method could find the global optimum within 200 evaluations of the original objective and constraint functions on all the six cases. According to the results from previously published performance tests, the obtained optima are sufficiently accurate and the recorded NFEs are much lower (Coello 2002; Mezura-Montes et al. 2007; Mezura-Montes and Coello 2008; Ao and Chi 2010; Regis 2011, Akay and Karaboga 2012; Garg 2014). Furthermore, the MSSR program has been repeatedly applied on these cases ten times to assess its robustness with the results shown in Tables 6 and 7. In most cases, the MSSR method could obtain an accurate solution with a small number of function evaluations, showing good computation efficiency as well as robustness on nonlinear constrained optimization problems.

5 Conclusions

A new multi-start space reduction (MSSR) algorithm for solving computationally expensive black-box global optimization problems has been introduced. In this new search method, sampling spaces at three different levels have been introduced. The entire design space is regarded as the Global Space (GS) and remains unchanged during the search. The other two varying spaces are Medium Space (MS) and Local Space (LS). As two reduced regions which contain the promising solutions, the locations and boundaries of MS and LS change automatically during the search due to the SM update mechanism of the algorithm. Each of the three spaces has its own role in the search for global optimum. The GS supports consideration of the entire design space and ensures that the true global optimum will not be easily missed; the MS directs the search to a promising region that contains several possible best solutions

at the specific stage of the search; and the LS accelerates the search around a local optimum. A multi-start optimization strategy is introduced to carry out search in these three introduced spaces, using LHS to provide multiple starting points and SQP solver on the SM with starting points in the defined space. The new MSSR method utilizes the kriging model in the search of the global optimal solution to increase search efficiency and reduce the number of expensive function evaluation. In each of the iterative search loops, a new selection scheme is used to obtain several promising sample points. This selection scheme ensures that the kriging-based SM is sufficiently exploited, and the unknown area of the SM can be gradually explored. The estimated MSE of the kriging-based SM is used as an important tool to explore the unknown area of the design space. The new algorithm has been tested using a large number of optimization benchmark problems, including 15 bound constrained problems, 2 nonlinear constrained problems, and 4 structural engineering applications. All benchmark tests and comparisons with results from other optimization methods have showed superior performance and robustness of the new MSSR method in dealing with computationally expensive black-box optimization problems.

In our continuous research, the new MSSR method will be applied to large structural optimization and complex multidisciplinary optimization problems for further testing and improvements. Its applications to higher-dimensional black-box global optimization problems in various engineering fields will also be studied.

Acknowledgments Supports from the China Scholarship Council (Grant No. 201406290111), Natural Sciences and Engineering Research Council of Canada, and National Natural Science Foundation of China (Grant No. 51375389) are gratefully acknowledged. The authors are also grateful to members of the research group for the implementation of some existing global optimization algorithms and benchmark test cases.

Table 7 Summary of results obtained by MSSR on WBD, PVD and SRD

Exp.	WBD		PVD		SRD	
	NFE	Opt. value	NFE	Opt. value	NFE	Opt. value
No.1	110	1.7253	88	5885.4051	131	2994.8493
No.2	133	1.7253	87	5885.3782	164	2996.4051
No.3	99	1.7249	75	5885.3427	189	2995.5840
No.4	162	1.7253	125	5885.3979	134	2994.4745
No.5	167	1.7560	107	5885.3576	102	2997.4988
No.6	201	1.7535	97	5885.3658	102	2997.4988
No.7	113	1.7256	91	5885.3778	111	2994.6535
No.8	100	1.7256	112	5885.4247	96	2997.0597
No.9	153	1.7256	98	5885.3408	69	2995.4729
No.10	105	1.7254	73	5885.3993	71	2997.3088

Appendix. List of benchmark test problems

(1) Banana function, $n=2$:

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad -2 \leq x_i \leq 2, \quad i = 1, 2$$

(2) Peaks function, $n=2$:

$$f(\mathbf{x}) = 3(1 - x_1)^2 e^{-x_1^2 - (x_2 + 1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2 - x_2^2} - \frac{1}{3} e^{-(x_1 + 1)^2 - x_2^2} \quad -3 \leq x_1 \leq 3, \quad -4 \leq x_2 \leq 4$$

(3) Goldstein and Price function (GP), $n=2$:

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad -6 \leq x_i \leq 6, \quad i = 1, 2$$

$$f(\mathbf{x}) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \\ \times \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right] \\ - 2 \leq x_i \leq 2, \quad i = 1, 2$$

(8) Leon function, $n=2$:

$$f(\mathbf{x}) = 100(x_2 - x_1^3)^2 + (x_1 - 1)^2 \quad -10 \leq x_i \leq 10, \quad i = 1, 2$$

(4) Six-hump Camel-back function (SC), $n=2$:

$$f(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \quad -2 \leq x_i \leq 2, \quad i = 1, 2$$

(5) Shubert function (Shub), $n=2$:

$$f(\mathbf{x}) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right) \\ -10 \leq x_i \leq 10, \quad i = 1, 2$$

(9) Shekel function, $n=4$:

$$f(\mathbf{x}) = - \sum_{i=1}^5 \left(c_i + \sum_{j=1}^4 (x_j - a_{ji})^2 \right)^{-1} \\ a = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{bmatrix}$$

$$c = [0.1 \quad 0.2 \quad 0.2 \quad 0.4 \quad 0.4 \quad 0.6 \quad 0.3 \quad 0.7 \quad 0.5 \quad 0.5] \\ 0 \leq x_i \leq 10, \quad i = 1, 2, 3, 4$$

(6) Generalized Polynomial function (GF), $n=2$:

$$f(\mathbf{x}) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 + (2.625 - x_1(1 - x_2^3))^2 \\ -2 \leq x_i \leq 2, \quad i = 1, 2$$

(10) Levy function, $n=4$:(7) Himmelblau function (HM), $n=2$:

$$f(\mathbf{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} \left[(y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1)) \right] + (y_n - 1)^2 (1 + 10 \sin^2(2\pi y_n)) \\ y_i = 1 + \frac{x_i - 1}{4}, \quad i = 1, \dots, n. \quad -10 \leq x_i \leq 10, \quad i = 1, \dots, n.$$

(11) Hartman6 function (HN6), $n=6$:

$$f(\mathbf{x}) = - \sum_{i=1}^4 \alpha_i \exp \left[- \sum_{j=1}^6 B_{ij} (x_j - Q_{ij})^2 \right]$$

$$\alpha = [1, 1.2, 3, 3.2]^T,$$

$$B = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$$

$$Q = 10^{-4} \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \quad 0 \leq x_i \leq 1, \quad i = 1, 2, \dots, 6$$

(12) Trid function, $n=6, 10$:

$$f(\mathbf{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1} \quad -n^2 \leq x_i \leq n^2, \quad i = 1, \dots, n.$$

(13) Sphere function, $n=10$:

$$f(\mathbf{x}) = \sum_{i=1}^{10} x_i^2 \quad -5.12 \leq x_i \leq 5.12, \quad i = 1, 2, \dots, 10.$$

(14) 16 Variable function (F16), $n = 16$:

$$f(\mathbf{x}) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1)$$

$$a_{ij(\text{row } 1-8)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$a_{ij(\text{row } 9-16)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad -1 \leq x_i \leq 1, \quad i = 1, 2, \dots, 16.$$

(15) G6 function (G6), $n = 2$:

$$\begin{aligned} \min f(\mathbf{x}) &= (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{s.t. } g_1(\mathbf{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0, \\ g_2(\mathbf{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0. \\ 13 &\leq x_1 \leq 100; \quad 0 \leq x_2 \leq 100. \end{aligned}$$

(16) Tension/Compression Spring Design (TSD), $n = 3$:

This problem minimizes the weight of a spring and meanwhile is subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and side constraints.

$$\begin{aligned} \min f(\mathbf{x}) &= x_1^2 x_2(x_3 + 2) \\ \text{s.t. } g_1(\mathbf{x}) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0, \\ g_2(\mathbf{x}) &= \frac{4x_2^2 - x_1 x_2}{12566 x_1^3 (x_2 - x_1)} + \frac{1}{5108 x_1^2} - 1 \leq 0, \\ g_3(\mathbf{x}) &= 1 - \frac{140.45 x_1}{x_3 x_2^2} \leq 0, \\ g_4(\mathbf{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0. \\ 0.05 &\leq x_1 \leq 2; \quad 0.25 \leq x_2 \leq 1.3; \quad 2 \leq x_3 \leq 15 \end{aligned}$$

(17) Welded Beam Design (TSD), $n = 4$:

This problem is designed for minimum cost and meanwhile is subject to constraints on shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam and side constraints.

$$\begin{aligned} \min f(\mathbf{x}) &= 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \\ \text{s.t. } g_1(\mathbf{x}) &= \tau(\mathbf{x}) - \tau_{\max} \leq 0, \\ g_2(\mathbf{x}) &= \sigma(\mathbf{x}) - \sigma_{\max} \leq 0, \\ g_3(\mathbf{x}) &= x_1 - x_4 \leq 0, \\ g_4(\mathbf{x}) &= 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \leq 0, \\ g_5(\mathbf{x}) &= 0.125 - x_1 \leq 0, \\ g_6(\mathbf{x}) &= \delta(\mathbf{x}) - \delta_{\max} \leq 0, \\ g_7(\mathbf{x}) &= P - P_c(\mathbf{x}) \leq 0, \end{aligned}$$

where

$$\begin{aligned} \tau(\mathbf{x}) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1 x_2}, \quad \tau'' = \frac{MR}{J}, \\ M &= P \left(L + \frac{x_2}{2} \right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2}, \quad \sigma(\mathbf{x}) = \frac{6PL}{x_4 x_3^2}, \\ \delta(\mathbf{x}) &= \frac{4PL^3}{E x_3^3 x_4}, \quad J = 2 \left\{ \sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\}, \\ P_c(\mathbf{x}) &= \frac{4.013E \sqrt{x_3^2 x_4^6 / 36}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right), \quad G = 12 \times 10^6 \text{ psi}, \\ P &= 6000 \text{ lb}, \quad L = 14 \text{ in}, \quad \delta_{\max} = 0.25 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \\ \tau_{\max} &= 13600 \text{ psi}, \quad \sigma_{\max} = 30000 \text{ psi}. \\ 0.1 &\leq x_1, x_4 \leq 2; \quad 0.1 \leq x_2, x_3 \leq 10 \end{aligned}$$

(18) Pressure Vessel Design (PVD), $n=4$:

This problem is to minimize the total cost of a cylindrical vessel, including the cost of material, forming and welding. Four design variables are thickness of the pressure vessel, thickness of the head, inner radius of the vessel and length of the vessel, respectively.

$$\begin{aligned} \min \quad & f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0, \\ & g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leq 0, \\ & g_3(\mathbf{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ & g_4(\mathbf{x}) = x_4 - 240 \leq 0. \\ & 1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625; \quad 10 \leq x_3, x_4 \leq 200 \end{aligned}$$

(19) Himmelblau's nonlinear optimization problem (Him), $n=5$:

This problem has five design variables, six nonlinear inequality constraints and ten boundary conditions.

$$\begin{aligned} \min \quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5, \\ & g_2(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2, \\ & g_3(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4, \\ & 0 \leq g_1(\mathbf{x}) \leq 92, \quad 90 \leq g_2(\mathbf{x}) \leq 110, \quad 20 \leq g_3(\mathbf{x}) \leq 25, \\ & 78 \leq x_1 \leq 102; \quad 33 \leq x_2 \leq 45; \quad 27 \leq x_3, x_4, x_5 \leq 45. \end{aligned}$$

(20) Speed Reducer Design (SRD), $n=7$:

This problem is to minimize the total weight of the speed reducer. It has eleven constraints which include the limits on the bending stress of the gear teeth, surface stress and transverse deflections of shafts.

$$\begin{aligned} \min \quad & f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \\ & + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \quad g_6(\mathbf{x}) = \frac{[(745x_5/(x_2x_3))^2 + 157.5 \times 10^6]^{1/2}}{85x_7^3} - 1 \leq 0, \\ & g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3} - 1 \leq 0, \quad g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \leq 0, \quad g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \leq 0, \\ & g_3(\mathbf{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0, \quad g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \leq 0, \quad g_{10}(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\ & g_4(\mathbf{x}) = \frac{1.93x_3^3}{x_2x_3x_7^4} - 1 \leq 0, \quad g_{11}(\mathbf{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0, \\ & g_5(\mathbf{x}) = \frac{[(745x_4/(x_2x_3))^2 + 16.9 \times 10^6]^{1/2}}{110x_6^3} - 1 \leq 0, \\ & 2.6 \leq x_1 \leq 3.6; \quad 0.7 \leq x_2 \leq 0.8; \quad 17 \leq x_3 \leq 28; \quad 7.3 \leq x_4, x_5 \leq 8.3; \\ & 2.9 \leq x_6 \leq 3.9; \quad 5.0 \leq x_7 \leq 5.5. \end{aligned}$$

References

- Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J Intell Manuf* 23(4):1001–1014
- Alexandrov NM, Dennis JEJ, Lewis RM, Torczon V (1998) A trust-region framework for managing the use of approximation models in optimization. *Struct Optim* 15(1):16–23
- Ao YY, Chi HQ (2010) An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design. *Eng* 2(01):65
- Björkman M, Holmström K (1999) Global optimization using DIRECT algorithm in Matlab. *Adv Model Optim* 1(2):17–37
- Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 191(11):1245–1287
- Cressie N (1990) The origins of kriging. *Math Geol* 22(3):239–252
- Csendes T (1988) Nonlinear parameter estimation by global optimization - efficiency and reliability. *Acta Cybern* 8(4):361–372
- Edke MS, Chang KH (2011) Shape optimization for 2-D mixed-mode fracture using Extended FEM (XFEM) and Level Set Method (LSM). *Struct Multidiscip Optim* 44(2):165–181
- Egea JA (2008) New heuristics for global optimization of complex bioprocesses. Ph.D. thesis. Universidade de Vigo, Spain
- Eves J, Toropov VV, Thompson HM, Kapur N, Fan J, Copley D, Mincher A (2012) Design optimization of supersonic jet pumps using high fidelity flow analysis. *Struct Multidisc Optim* 45(5):739–745
- Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1):50–79
- Garg H (2014) Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J Ind Manag Optim* 10(3):777–794
- Gen M, Cheng R (1997) Genetic algorithms & engineering design. Wiley, New York
- Gu J, Li GY, Dong Z (2012) Hybrid and adaptive meta-model-based global optimization. *Eng Optim* 44(1):87–104
- Gutmann HM (2001) A radial basis function method for global optimization. *J Glob Optim* 19(3):201–227

- Harmati I, Rövid A, Várlaki P (2010) Approximation of force and energy in vehicle crash using LPV type description. *WSEAS Trans Syst* 9(7):734–743
- Hedar A (2004) Studies on metaheuristics for continuous global optimization problems. Ph.D. thesis. Kyoto University, Kyoto, Japan
- Himmelblau DM (1972) Applied nonlinear programming. McGraw-Hill Book Company, New York
- Iman RL (2008) Latin hypercube sampling. Encyclopedia of quantitative risk analysis and assessment. John Wiley, New York
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13
- Jin R, Chen W, Sudjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. *J Stat Plan Infer* 134(1):268–287
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492
- Jouhaud JC, Sagaut P, Montagnac M, Laurenceau J (2007) A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil. *Comput Fluids* 36(3):520–529
- Lasdon L, Duarte A, Glover F, Laguna M, Marti R (2010) Adaptive memory programming for constrained global optimization. *Comput Oper Res* 37(8):1500–1509
- Long T, Wu D, Guo X, Wang GG, Liu L (2015) Efficient adaptive response surface method using intelligent space exploration strategy. *Struct Multidiscip Optim* 51(6):1335–1362
- Lophaven SN, Nielsen HB, Søndergaard J (2002) DACE-A Matlab Kriging toolbox. version 2.0
- Mezura-Montes E, Coello CAC, Velázquez-Reyes J, Muñoz-Dávila L (2007) Multiple trial vectors in differential evolution for engineering design. *Eng Optim* 39(5):567–589
- Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int J Gen Syst* 37(4):443–473
- Mueller J (2012) User guide for modularized surrogate model toolbox. Department of Mathematics, Tampere University of Technology, Tampere, Finland
- Mullur AA, Messac A (2005) Extended radial basis functions: more flexible and effective metamodeling. *AIAA J* 43(6):1306–1315
- Myers RH, Montgomery DC, Anderson-Cook CM (2009) Response surface methodology: process and product optimization using designed experiments. John Wiley & Sons, New York
- Ong YS, Nair PB, Keane AJ (2003) Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J* 41(4):687–696
- Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. *Prog Aerosp Sci* 41:1–28
- Regis RG (2011) Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Comput Oper Res* 38(5):837–853
- Regis RG, Shoemaker CA (2007a) A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J Comput* 19(4):497–509
- Regis RG, Shoemaker CA (2007b) Improved strategies for radial basis function methods for global optimization. *J Glob Optim* 37(1):113–135
- Regis RG, Shoemaker CA (2013) A quasi-multistart framework for global optimization of expensive functions using response surface models. *J Glob Optim* 56(4):1719–1753
- Shan S, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct Multidiscip Optim* 41(2):219–241
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
- The Mathworks, Inc.: Matlab Optimization Toolbox: User's Guide, http://www.mathworks.com/help/pdf_doc/gads/gads_tb.pdf, 2015
- Villanueva D, Le Riche R, Picard G, Haftka RT (2012) Surrogate-based agents for constrained optimization. In: The 14th AIAA Non-Deterministic Approaches Conference, Honolulu, AIAA-2012-1935
- Wang GG, Simpson T (2004) Fuzzy clustering based hierarchical metamodeling for design space reduction and optimization. *Eng Optim* 36(3):313–335
- Wang GG, Dong Z, Aitchison P (2001) Adaptive response surface method – a global optimization scheme for approximation-based design problems. *Eng Optim* 33(6):707–734
- Wang L, Shan S, Wang GG (2004) Mode-pursuing sampling method for global optimization on expensive black-box functions. *Eng Optim* 36(4):419–438
- Xiang Y, Arora JS, Chung HJ, Kwon HJ, Rahmatalla S, Bhatt R, Abdel-Malek K (2012) Predictive simulation of human walking transitions using an optimization formulation. *Struct Multidiscip Optim* 45(5):759–772
- Yang XS (2010) Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons
- Younis A, Dong Z (2010) Metamodeling and search using space exploration and unimodal region elimination for design optimization. *Eng Optim* 42(6):517–533
- Younis A, Karakoc K, Dong Z, Park E, Suleman A (2011) Application of SEUMRE global optimization algorithm in automotive magnetorheological brake design. *Struct Multidiscip Optim* 44(6):761–772
- Zhang XY, Jin XL, Qi WG, Guo YZ (2008) Vehicle crash accident reconstruction based on the analysis 3D deformation of the auto-body. *Adv Eng Softw* 39(6):459–465
- Zhou Z, Ong YS, Lim MH, Lee BS (2007) Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Comput* 11(10):957–971