Hybrid surrogate-based constrained optimization with a new constraint-handling method ——Supplementary Material

This is the supplementary material to the paper entitled "Hybrid surrogate-based constrained optimization with a new constraint-handling method ", submitted to IEEE Transactions on Cybernetics. This supplementary material includes three parts. One is the description of the procedures of the kriging-based and RBF-based optimizations, and another is the results of the numerical experiment. This material also illustrates the optimization results of the twenty three constrained test problems considered in this work. Fifteen problems among them are inequality constrained problems, and the others are equality constrained problems. In each figure, it illustrates the progress of the objective value and constraint violation of the best solution found by COBRA, BCGO, Extended ConstrLMSRBF, COBRA-EH and our method in every iteration. In Section III, a method to adaptively adjust the feasible threshold is presented.

I. PROCEDURES OF THE KRIGING-BASED OPTIMIZATION AND RBF-BASED OPTIMIZATION

The framework and main process of the HSBCO are described in the main manuscript, and this section presents the details of kriging-based optimization in the first phase and RBF-based optimization in the second phase of the proposed method.

A. Procedure of the kriging-based constrained efficient global optimization algorithm

The kriging-based constrained efficient global optimization algorithm is specified as follows:

Algorithm 2: Kriging-based constrained efficient global optimization

Name: $KrigingOptimization(\mathbb{D}, p_{best}, r_{best}, \tilde{C}^{\max}, lb, ub, K, N_{KriRBF})$

Input: Initial training data set $\mathbb{D} = \{X, Y\}$, the best solution p_{best} in $\mathbb{D}$, the maximum number of function evaluations K, threshold N_{KriRBF}, the lower and upper bounds of design variables lb and ub, maximum constraint violation vector $\tilde{C}^{\max}$ and the minimum constraint violation r_{best}.

Output: new data set $\mathbb{D} = \{X, Y\}$, the best solution found, $p_{best}(\mathbf{x}_{best}, y_{best})$.

1. Initialize parameters α, ρ, ε
2. Set feasibility threshold $r_{threshold}$
3. Define a counter n_{KriRBF}

While $size(X,1) < K$

4. Construct the Kriging model $\hat{f}_{RBF}(\mathbf{x})$ of the objective function $f(\mathbf{x})$ and RBF models $\hat{g}_{RBF}(\mathbf{x})$ and $\hat{h}_{RBF}(\mathbf{x})$ of the constraints
5. Sample a point by maximizing the modified infill criterion, i.e., $\mathbf{x}^{opt} = \arg\max_{\mathbf{x}\in D} EI_{\mathrm{mod}}(\mathbf{x})$
6. Calculate the minimum Euclidean distance $d_{\min}$

   While $d_{\min} < \varepsilon$

7. Update the parameter ρ : $\rho \leftarrow \alpha \cdot \rho$
8. Repeat $\mathbf{x}^{opt} = \arg\max_{\mathbf{x}\in D} EI_{\mathrm{mod}}(\mathbf{x})$

   End While

9. Evaluate the objective function: $y^{opt} = f(\mathbf{x}^{opt})$
10. Evaluate the true constraint violation: $\hat{r}^{opt} = ConstraintViolation(\mathbf{x}^{opt}, \tilde{C}^{\max})$

If $y^{opt} < y_{best}$ when $\hat{r}^{opt} \le r_{best}$

11. Update the best solution: $p_{best}(\mathbf{x}_{best}, y_{best}) \leftarrow p^{opt}(\mathbf{x}^{opt}, y^{opt})$, and $r_{best} \leftarrow r^{opt}$

End If

12. Update training dataset: $X \leftarrow X \cup \mathbf{x}^{opt}$, $Y \leftarrow Y \cup y^{opt}$

   If $n_{KriRBF} > N_{KriRBF}$

13. break

   End if

End While

The routine of the constraint violation with true constraints $\hat{r} = ConstraintViolation(\mathbf{x}, \tilde{C}^{\max})$ is similar to the routine with the surrogates of the constraints $\hat{r}_{sur} = SurrogateConstraintViolation(\mathbf{x}, \tilde{C}^{\max}, \hat{g}_{RBF}(\mathbf{x}), \hat{h}_{RBF}(\mathbf{x}))$. Both routines can be described as follow:

Algorithm 3: **Evaluation of Constraint Violation**

Name: $\hat{r} = ConstraintViolation(\mathbf{x}, \tilde{C}^{\max})$, or $\hat{r}_{sur} = SurrogateConstraintViolation(\mathbf{x}, \tilde{C}^{\max}, \hat{g}_{RBF}(\mathbf{x}), \hat{h}_{RBF}(\mathbf{x}))$

Inputs: design point $\mathbf{x}$, and maximum constraint violation $\tilde{C}^{\max}$, (and the surrogates of the constraints $\hat{g}_{RBF}(\mathbf{x})$ and $\hat{h}_{RBF}(\mathbf{x})$)

Output: normalized union Euclidean distance of the constraint violations $\hat{r}$

1. Evaluate the constraints: $C = Constraint(\mathbf{x})$ or $\hat{r}_{sur} = SurrogateConstraintViolation(\mathbf{x}, \tilde{C}^{\max}, \hat{g}_{RBF}(\mathbf{x}), \hat{h}_{RBF}(\mathbf{x}))$ where $C = [C_1, \cdots, C_{n_{constr}}]$, n_{constr} is the number of constraints

   If $C_i \le 0$, $i = 1, \cdots, n_{constr}$

2. Set $C_i \le 0$

   Else

   if $C_i \ge \tilde{C}_i^{\max}$, $i = 1, \cdots, n_{constr}$

3. Set $C_i = \tilde{C}_i^{\max}$

   End If

   End If

4. Return $\hat{r} = \sqrt{\frac{C_1}{\tilde{C}_1^{\max}} + \cdots + \frac{C_{n_{constr}}}{\tilde{C}_{n_{constr}}^{\max}}}$

B. Procedure of the RBF-based constrained global optimization

The procedure of the RBF-based constrained global optimization is specified as follows:

Algorithm 4: RBF-based constrained global optimization

Name: $RBFOptimization(\mathbb{D}, p_{best}(\mathbf{x}_{best}, y_{best}), r_{best}, lb, ub, K, N_{RBF})$

Inputs: training data set $\mathbb{D} = \{X, Y\}$, the best solution $p_{best}(\mathbf{x}_{best}, y_{best})$ in $\mathbb{D}$,the maximum number of function evaluations K, threshold N_{RBF}, the lower and upper bounds of design variables lb and ub, and the minimum constraint violation r_{best}.

Output: global optimal solution $p^{opt}(\mathbf{x}^{opt}, y^{opt})$

1. Find the maximum constraint $\tilde{C}^{\max} = [\tilde{C}_1^{\max}, \cdots, \tilde{C}_{n_{constr}}^{\max}]$ in $\mathbb{D}$.
2. Set the variance ρ of the Gaussian penalty function (19)
3. Set feasibility threshold $r_{threshold}$
4. Set $\Theta = [\theta_1, \cdots, \theta_m]$
5. Set the initial value υ_{ini} and minimum value $\upsilon_{\min}$ of parameter υ
6. Set the adjustment parameter β
7. Set the number of random solutions m
8. Define the counter n_{RBF} of consecutive iterations before switching.
9. Define a flag for global and local search: global search if $flag = 0$, and local search if $flag = 1$.
10. Define the last best solution: $p_{lastbest}(\mathbf{x}_{lastbest}, y_{lastbest}, r_{lastbest})$

    While $size(X) < K$

11. Construct the RBF model $\hat{f}_{RBF}(\mathbf{x})$ and $\hat{g}_{RBF}(\mathbf{x})$ and $\hat{h}_{RBF}(\mathbf{x})$ of the objective and the constraints

    If $flag = 0$

12. Search for global optimal solution candidate $\mathbf{x}^{opt}$ by solving (26): $\mathbf{x}^{opt} = \arg\min_{\mathbf{x}\in[lb,ub]} \hat{f}_{RBF}(\mathbf{x}) + (1 - P(\mathbf{x})) \cdot \Gamma$

13. Evaluate the approximate constraint violation:

14. $\hat{r}_{sur}^{opt} = SurrogateConstraintViolation(\mathbf{x}^{opt}, \tilde{C}^{\max}, \hat{g}_{RBF}(\mathbf{x}), \hat{h}_{RBF}(\mathbf{x}))$

If $\hat{r}_{sur}^{opt} < r_{threshold}$

15. Evaluate the true objective value $y^{opt} = f(\mathbf{x}^{opt})$

16. Evaluate the true constraint violation $\hat{r}(\mathbf{x}^{opt}) = ConstraintViolation(\mathbf{x}^{opt}, \tilde{C}^{\max})$

If $y^{opt} < y_{best} \,\&\, \hat{r}(\mathbf{x}^{opt}) = 0$

17. Update the current best solution: $\mathbf{x}_{best} \leftarrow \mathbf{x}^{opt}$, $y_{best} \leftarrow y^{opt}$, $r_{best} \leftarrow \hat{r}(\mathbf{x}^{opt})$

End If

Else

18. Find a feasible solution candidate by solving (24): $\mathbf{x}^{opt} = \arg\min_{\mathbf{x}\in[lb,ub]} \hat{r}_{sur}(\mathbf{x})$ with

19. $\hat{r}_{sur} = SurrogateConstraintViolation(\mathbf{x}, \tilde{C}^{\max}, \hat{g}_{RBF}(\mathbf{x}), \hat{h}_{RBF}(\mathbf{x}))$

20. Evaluate the true objective value $y^{opt} = f(\mathbf{x}^{opt})$

21. Evaluate the true constraint violation $\hat{r}(\mathbf{x}^{opt}) = ConstraintViolation(\mathbf{x}^{opt}, \tilde{C}^{\max})$

If $y^{opt} < y_{best} \,\&\, \hat{r}(\mathbf{x}^{opt}) = 0$

22. Update the current best solution: $\mathbf{x}_{best} \leftarrow \mathbf{x}^{opt}$, $y_{best} \leftarrow y^{opt}$, $r_{best} \leftarrow \hat{r}(\mathbf{x}^{opt})$

End If

End If

Else

23. Determine the radius of neighborhood: $R = \upsilon \cdot (ub - lb)$

24. Determine the lower and upper bounds: $LB_i = \max[x_{best,i} - R_i, lb_i]$, $UB_i = \min[x_{best,i} + R_i, ub_i]$

25. Generate random solution set $X_{rand} = \{\mathbf{x}_{rand,i}, i = 1, \cdots, m\}$

26. Compute the distances between the solutions in X_{rand}, and determine the minimum distance of the solution $\mathbf{x}_{rand,i}$, i.e., $d_{\mathbf{x}_{rand,i}} = \min\{\|\mathbf{x}_{rand,i} - \mathbf{x}_{rand,j}\|, j = 1, \cdots, m, \, and \, i \neq j\}$.

27. Collect the candidate solutions with $d_{\mathbf{x}_{rand,i}} \geq \eta_k$: $X_{rand}^{valid_1} = \{\mathbf{x}_{rand,i} \mid d_{\mathbf{x}_{rand,i}} \geq \eta_k, i = 1, \cdots, m\}$

28. Estimate the constraint violations of the random solution set: $\hat{r}_{sur}^{rand} = \{\hat{r}_{sur}^{rand,i}, i = 1, \cdots, m\}$

29. Collect the candidate solutions with the minimum estimated constraint violation: $X_{rand}^{valid_2} = \{\mathbf{x}_{rand,i} \mid \hat{r}_{sur}^{rand,i} = \min(\hat{r}_{sur}^{rand}), i = 1, \cdots, size(X_{rand}^{valid_1})\}$

30. Estimate the objective values of the solutions in $X_{rand}^{valid_2}$: $Y_{rand}^{valid_2} = \{\hat{f}_{RBF}(\mathbf{x}_{rand,i}), i = 1, \cdots, size(X_{rand}^{valid_2})\}$

31. Select a solution $\mathbf{x}^{opt}$ with the minimum estimated objective value

32. Evaluate the objective value of solution $\mathbf{x}^{opt}$: $y^{opt} = f(\mathbf{x}^{opt})$.

33. Evaluate the true constraint violation $\hat{r}(\mathbf{x}^{opt}) = ConstraintViolation(\mathbf{x}^{opt}, \tilde{C}^{\max})$

If $y^{opt} < y_{best} \,\&\, r(\mathbf{x}^{opt}) = 0$

34. Update the current best solution: $\mathbf{x}_{best} \leftarrow \mathbf{x}^{opt}$, $y_{best} \leftarrow y^{opt}$, $r_{best} \leftarrow \hat{r}(\mathbf{x}^{opt})$

Else

35. Update the parameter υ: $\upsilon \leftarrow \upsilon / \beta$.

End If

If $\upsilon < \upsilon_{\min}$

36. $\upsilon := \upsilon_{ini}$

End if

End if

If $p^{opt}(\mathbf{x}^{opt}, y^{opt})$ is worse than $p_{best}(\mathbf{x}_{best}, y_{best})$

37. $n_{RBF} \leftarrow n_{RBF} + 1$

Else

38. $n_{RBF} = 0$

End if

If $n_{RBF} \geq N_{RBF}$

If $flag = 0$

39. $flag = 1$

Else

40. $flag = 0$

End if

41. $n_{RBF} = 0$

End if

42. Update dataset $X : X \leftarrow X \bigcup \mathbf{x}^{opt}$, $Y \leftarrow Y \bigcup y^{opt}$

End while

return $p^{opt}(\mathbf{x}^{opt}, y^{opt})$

II. Properties of Test Problems and Results of the numerical experiment

A. Properties of test problem

This work considers 23 test problems. The information about G1~G24 can be found in reference [63], SE, TSD and PVD refer to [45], and RB (Rosenbrock function) and MB (Mishra's Bird function) refer to [64] and [65], respectively. Their number of design variables and constraint conditions are described in Table S1.

Table S1. Properties of the test problems

Problems	d	N_{inequ}	N_{equ}	Known best
G1	13	9	0	-15
G2	20	2	0	-0.80361910412559
G3	10	0	1	-1.00050010001000
G4	5	6	0	-30665.539
G5	4	2	3	5126.4967140071
G6	2	2	0	-6961.81387558015
G7	10	8	0	24.30620906818
G8	2	2	0	-0.0958250414180359
G9	7	4	0	680.630057374402
G10	8	6	0	7049.24802052867
G11	2	0	1	0.7499
G12	3	1	0	unknown
G13	5	0	3	0.053941514041898
G15	3	0	2	961.715022289961
G21	7	1	5	193.724510070035
G23	9	2	4	-400.055099999999584
G24	2	2	0	-5.50801327159536
SE	2	1	0	-1.1743
TSD	3	4	0	0.0126652327883
PVD	4	4	0	5888.66
SYP	2	0	1	-0.08752
RB	2	2	0	0
MB	2	1	0	-106.7645367

The details of the known best points of the test problems are given as follows [Res 1, Res 2, Res 3]:

G1:

f(1,1,1,1,1,1,1,1,1,3,3,3,1)=-15;

Constraints:

g1=0, g2=0, g3=0, g4=-5, g5=-5, g6=-5, g7=0, g8=0, g9=0

G2:

f(3.16246061572185,3.12833142812967,3.09479212988791,3.06145059523469,3.02792915885555,2.99382606701730,2.95866871765285,2.92184227312450,0.49482511456933,0.48835711005490,0.48231642711865,0.47664475092742,0.47129550835493,0.46623099264167,0.46142004984199,0.45683664767217,0.45245876903267,0.44826762241853,0.44424700958760,0.44038285956317)= -0.803619104;

Constraints:

g1=-1.28785870856518e-14, g2=-120.067416152593

G3:

f(1, 0.707106781, 0.577350269, 0.5, 0.447213595, 0.40824829, 0.377964473, 0.353553391, 0.333333333, 0.316227766)= -1.0005001;

Constraint:

g1=1.92896825270731

G4:

f(78, 33, 29.9952560256815985, 45, 36.7758129057882073)= -30665.539;

Constraints:

g1=0, g2=-92, g3=-11.1594996910731, g4=-8.84050030892686, g5=-5.00000000000000, g6=-3.55271367880050e-15

G5:

f(679.945148297028709, 1026.06697600004691,0.118876369094410433, -0.396233485215178266)= 5126.496714;

Constraints:

g1=-0.0348901456904114, g2=-1.06510985430959, g3=9.99999999748980e-05, g4=9.99999999748980e-05, g5=9.99999999748980e-05

G6:

f(14.09500000000000064, 0.8429607892154795668)= -6961.81387558015;

Constraints:

g1=0, g2=0

G7:

f(2.17199634142692,2.3636830416034,8.77392573913157,5.09598443745173,0.990654756560493,1.43057392853463,1.32164415364306,9.82872576524495,8.2800915887356,8.3759266477347)= 24.30620907;

Constraints:

g1=5.68434188608080e-14, g2=-1.17239551400417e-13, g3=3.90798504668055e-14, g5=-6.02540239924565e-12, g6=0, g7=-2.84217094304040e-14, g8=-6.14850368960364, g9=-50.0239617318381

G8:

f(1.22797135260752599, 4.24537336612274885)= -0.095825041;

Constraints:

g1=-1.73745972329799, g2=-0.167763263805117

G9:

f(2.33049935147405174, 1.95137236847114592,-0.477541399510615805, 4.36572624923625874, -0.624486959100388983, 1.03813099410962173, 1.5942266780671519)= 680.6300574;

Constraints:

g1=-1.37667655053519e-14, g2=-252.561716343466, g3=-144.878178454615, g4=-2.48689957516035e-14

G10:

f(579.306685017979589,1359.97067807935605,5109.97065743133317,182.01769963061534,295.601173702746792,217.982300369384632,286.41652592786852,395.601173702746735)= 7049.248021;

Constraints:

g1=0, g2=0, g3=-5.55111512312578e-16, g4=-1.45519152283669e-11, g5=-2.91038304567337e-11, g6=-1.16415321826935e-10

G11:

f(0.707106781,0.5)= 0.7499;

Constraint:

g1=2.63818022983031e-10

G12:

The global optimum of this problem depends on the parameters a_1, a_2, a_3 of the constraint. The detail of G12 is described by the following equation:

$$
\begin{aligned}
&y(\mathbf{x}) = -\frac{100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2}{100} \\
&s.t.\, g(\mathbf{x}) = (x_1-a_1)^2+(x_2-a_2)^2-(x_3-a_3)^2-0.0625 \le 0
\end{aligned} \tag{1}
$$

where $0 \le x_i \le 10, i=1,2,3$, and $a_1, a_2, a_3 = 1,2,\cdots,9$. If suitable values for the parameters a_1, a_2, a_3 are set, then the feasible global optimum will be located at $\mathbf{x}^* = (5,5,5)$, and the corresponding objective value is -1. But it is not easy to find a set of suitable parameters to make the feasible global optimum be located at $\mathbf{x}^* = (5,5,5)$. In this experiment, the three parameters were set as 1, 2 and 3, respectively. In this case, the solution $\mathbf{x}^* = (5,5,5)$ becomes infeasible, and the true global optimum is unknown.

G13:

f(-1.71714224003, 1.59572124049468,1.8272502406271,-0.763659881912867,-0.76365986736498)= 0.053941514;

Constraint:

g1=9.99999999944379e-05, g2=0.000100000000003320, g3=9.99999999988788e-05

G15:

f(3.51212812611795133, 0.216987510429556135, 3.55217854929179921)= 961.7150223;

Constraints:

g1=9.99999999997669e-05, g2=9.99999999891088e-05

G21:

f(193.724510070034967, 5.56944131553368433e-27, 17.3191887294084914, 100.047897801386839, 6.68445185362377892, 5.99168428444264833, 6.21451648886070451)= 193.7245101;

Constraints:

g1=-2.84217094304040e-14, g2=9.99999974737875e-05, g3=621.749410562572, g4=9.99999999997669e-05 g5=9.99999999997669e-05, g6=9.99999999997669e-05

G23 [Res 1]:

f(0.00510000000000259465, 99.9947000000000514, 9.01920162996045897e-18, 99.9999000000000535, 0.000100000000027086086, 2.75700683389584542e-14, 99.9999999999999574, 2000.0100000100000100008)= -400.0551;

Unfortunately, the known best solution presented in [Res 1] is incomplete. It seems something of this solution is missed.

G24:

f(2.329520197477623, 3.17849307411774)= -5.508013272;

Constraints:

g1=-4.88498130835069e-14, g2=1.70530256582424e-13

SE:

f(2.745,2.3523)= -1.1743;

Constraint:

g1=-9.18301275909399e-07

TSD [Res 3]:

f(0.051689156131, 0.356720026419, 11.288831695483)= 0.0126652327883;

Constraints:

g1=-1.44297906956581e-11, g2=1.04636299624872e-11, g3=-4.05379014638206, g4=-0.727727211633333

PVD:

f(0.7792, 0.3852, 40.3713, 199.3308)= 5888.66;

Constraints:

g1=-3.39099999999704e-05,g2=-5.77979999999978e-05,g3=-250.855683541624,g4=-40.6692000000000

SYP:

f(-0.1818 0.9959)= -0.0839;

Constraint:

g1=7.96199999999470e-05

RB:

$f(1,1)=0$;

Constraints:

g1=0, g2=0

MB:

$f(-3.1302468, -1.5821422)= -106.7645367$.

Constraint:

g1=-9.82227103006892

[Res 1]. *Efrén Mezura-Montes, Omar Cetina-Domínguez. Empirical analysis of a modified Artificial Bee Colony for constrained numerical optimization. Applied Mathematics and Computation, 218: 10943–10973, 2012*

[Res 2]. *Huachao Dong, Baowei Song, Zuomin Dong, Peng Wang. SCGOSR: Surrogate-based constrained global optimization using space reduction. Applied Soft Computing, 65:462–477, 2018.*

[Res 3]. *Garg, Harish . "Solving structural engineering design optimization problems using an artificial bee colony algorithm." Journal of Industrial & Management Optimization 10(3):777-794, 2014.*

B. Experimental result

To demonstrate the optimization of the algorithms on the test problems, we recorded the best solution and constraint violation at every iteration, and plot them in Fig. 1S- 23S, respectively. In addition, for better comparison, we put the medians of the global optima of the test problems found by the algorithms in Fig. 24S, and we provide the details of the globally optimal points found by HSBCO are given as follows:

G1:

f(1, 0.999999994304638, 0.999999999996492,0.999999915571090,1,1,1,1, 0.999999998271348, 2.99999926680943, 2.99999982204242, 2.99999983578305, 0.999999838761289)= -14.9999983110287;

Constraints:

g1=-9.22538873737722e-07, g2=-8.97414535572239e-07, g3=-3.53572268352309e-07, g4=-5.00000073319057, g5=-5.00000013239468, g6=-5.00000016418889, g7=-5.64332749863894e-07, g8=-1.77957579783339e-07, g9=-1.62488297927865e-07

G2:

f(5.63752582077430, 4.03645879887191, 3.18993356912130, 5.09311083951308, 0.350567605767289, 0.338763147954773, 0.360809002514289, 2.95679981928567, 0.333755013615159, 2.46075313379958, 0.371754852747970, 0.379198683431602, 0.364587891503094, 0.339938870687046, 0.346300557431858, 2.81732166966283, 0.362451717408195, 2.83909387078731, 3.02757410901102, 0.367116153826775)= -0.496271286926841;

Constraints:

g1=5.39858158177253e-11, g2=-114.026184872285

G3:

f(0.316675891666452, 0.323450169992633, 0.315155811977024, 0.314375136760537, 0.315523475344349, 0.315697895522475, 0.316019869868962, 0.314979486135265)= -0.999386766045074;

Constraint:

g1=-1.75391234868982e-09

G4:

f(78 33.0017284824460, 29.9947647563331, 44.9995162428572, 36.7752590447642)= -30665.7326772422;

Constraints:

g1=0.000310349209470928, g2=-92.0003103492095, g3=-11.1588371382763, g4=-8.84116286172369, g5=-5.00028104802540, g6=0.000281048025399855

G5:

f(678.583001464931, 1023.50982204729, 0.118594026273886, -0.395180624150917)= 5110.03947835775;

Constraint:

g1=-0.0362253495751970, g2=-1.06377465042480, g3=0.583907320677213, g4=0.988550083381142, g5=1.80470127325884

G6:

f(14.0947986217041, 0.842520231744258)= -6962.30906039213;

Constraints:

g1=7.19387571734842e-09, g2=0.000402749397977686

G7:

f(2.17149137039167, 2.36467636107817, 8.77360447173358, 5.09358346997454, 0.991063545553153, 1.43108229243357, 1.32080838200125, 9.82811975801799)= 24.3062552629381;

Constraints:

g1=-3.68843160458709e-08, g2=-1.62693957861393e-07, g3=-6.46673363746686e-08, g4=-4.10619270780899e-05, g5=-4.27275686831763e-07, g6=-2.72214233376644e-07, g7=-6.15013820771799, g8=-50.0399541840863

G8:

f(1.22824464439397, 3.74363758706965)= -0.105454995571428;

Constraints:

g1=-1.23505268058718, g2=-0.162522957630499

G9:

f(1.64657488804173, 1.80221452117813, 1.49215253874562, 4.64265036141734, 0.444110584411462, -0.718845461169403, 0.549925472876088)= 724.114577016294;

Constraints:

g1=-9.73137659343593e-09, g2=-238.603600454318, g3=-156.179771395440, g4=-1.95825755611168e-09

G10:

f(642.674039582746， 1171.25427315339， 5249.14259696899， 187.044527969872 ， 290.037517260712， 212.952364915637， 296.964320348172， 390.037386715484)= 7063.07090970512;

Constraints:

g1=-7.76778622824104e-06, g2=-0.000106725902470406, g2=-1.30545228105294e-06, g4=-54.5977954125556, g5=-2.78991366695846, g6=-7.36759816063568

G11:

f(0.707874422153400, 0.501393088283924)= 0.7499;

Constraint:

g1=0.000306890744913635

G12:

f(1.15626023803962, 2.11726644805889, 8.84401509448681)= -0.621390598641953;

Constraint:

g1=-2.74194941751738e-08

G13:

f(-1.63297973368204， 1.49696289613582， 1.47433648687155， 1.68832982426517 ， 0.261445006470463)= 0.00227723794848746;

Constraints:

g1=-1.13544055935222e-07, g2=8.11953571044910e-09, g3=2.17913651567869e-08;

G15:

f(3.48824613988198， 0.208057215198934， 3.60991612714015)= 961.396038039433;

Constraints:

g1=0.242643382184454, g2=0.0881830218220117

G21:

f(358.454917376894， 28.5541105892787， 5.48641859939573， 233.224447945320 ， 6.50243055411658， 6.27898055523257， 5.45361732309785，)= 358.454917376894;

Constraints:

g1=0.238034448491021, g2=6.66629502472301, g3=2.34869531427103, g4=2.29315028326127e-05, g5=-3.81166188434534e-05, g6=-0.000216408653480826

G23:

f(1.06290379409880， 99.7488476235035， 0.510742477100147， 100.603040721553 ， 0.942498561436840， 0.444567802863501， 98.0111793903414， 198.685086437295 ， 0.0101616558640558)= -401.842326939658;

Constraints:

g1=-0.00948111869120388, g2=0.00224076993742361, g3=0.302031781051042, g4=0.00189212208053280, g5=0.0128117185268074, g6=0.0708663254003454

G24:

f(2.32952019936084,3.17849304334237)= -5.50801324270322;

Constraints:

g1=-4.61511349136856e-08, g2=-2.19243645460665e-08

SE:

f(2.74570109029209, 2.35300192087764)= -1.17424853993213;

Constraint:

g1=-8.77157235734670e-08

TSD:

f(0.0514855447894149, 0.352478211145434, 11.5163763485856)= 0.0126288317277333;

Constraints:

g1=1.68594880215478e-05, g2=0.000248284963390466, g3=-3.97008720928026, g4=-0.754982974232437

PVD:

f(0.778168641495407,0.384649162822226,40.3196187254776,199.999999998047)= 5885.33277523461;

Constraints:

g1=-9.36895006020677e-11, g2=-1.81169745872012e-10, g3=-8.80607403814793e-05,g4=-40.0000000019527

SYP:

f(0.170188299085617, -0.995943461621032)= -0.0875195844475520;

Constraint:

g1=0.000855606967901212

RB:

f(0.999998975338284, 1.00000001866185)= 4.28705811377444e-10;

Constraints:

g1=-1.86618525077620e-08, g2=-1.00599986385674e-06

MB:

f(-3.13003884489657, -1.58211306101684)= -106.764530703486.

Constraint:

g1=-9.82129415073257;

Notes: the word “iteration” used in this paper represents the number of the function evaluations after initial sampling.

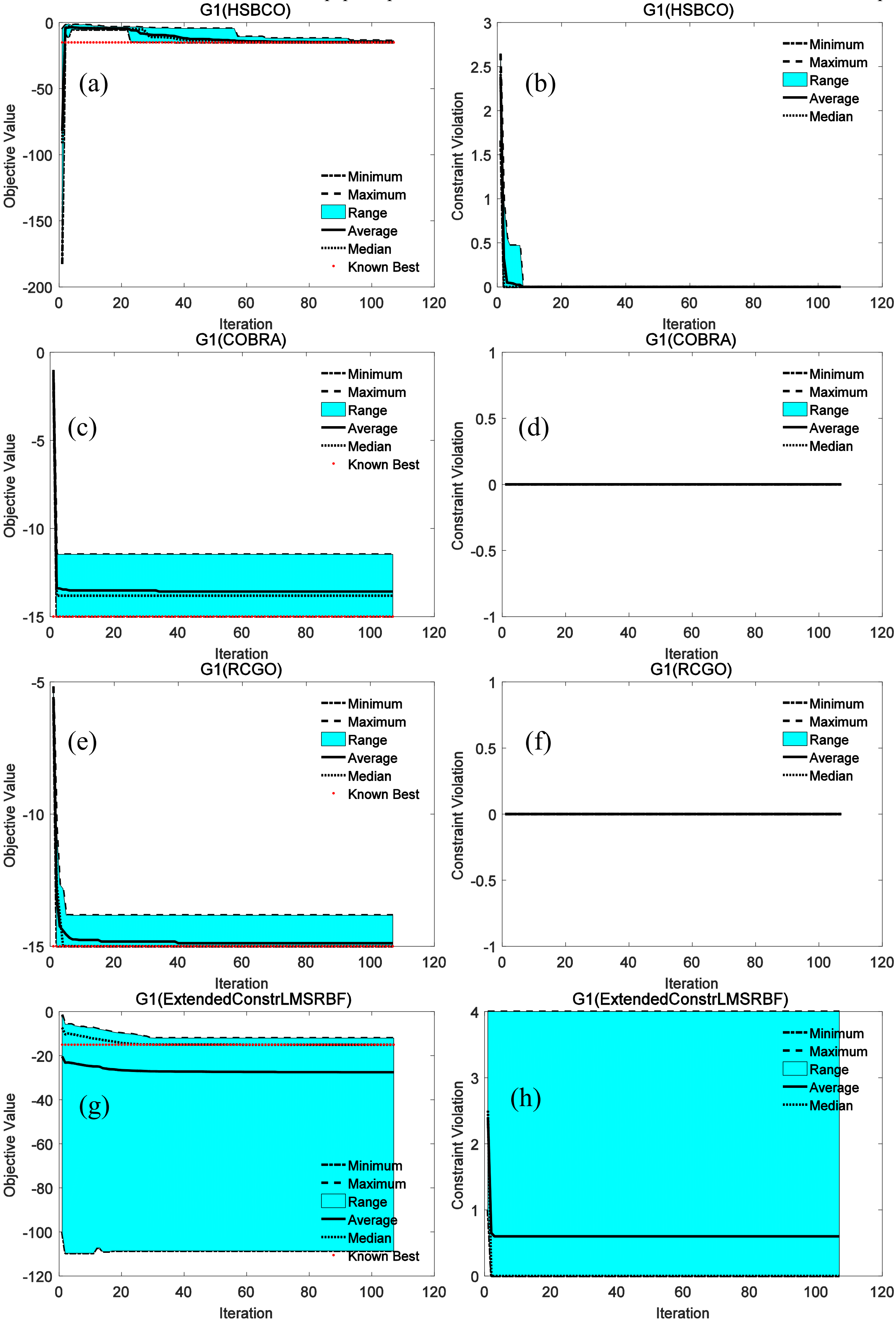

Fig. 1S. Convergence processes of objective value and constraint violation of problem G1 in 20 trials with 107 iterations (function evaluations).

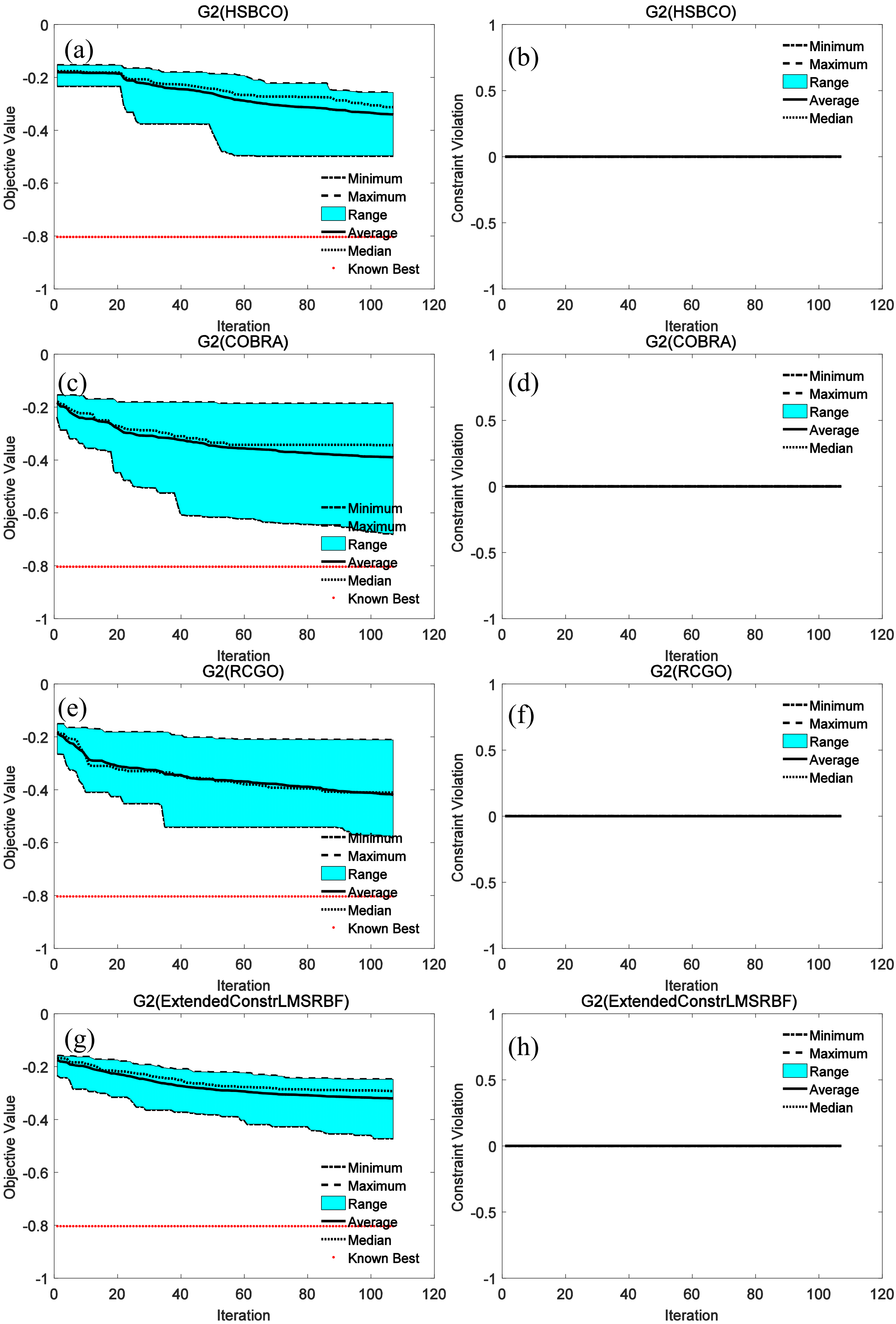

Fig. 2S. Convergence processes of objective value and constraint violation of problem G2 in 20 trials with 107 iterations (function evaluations).

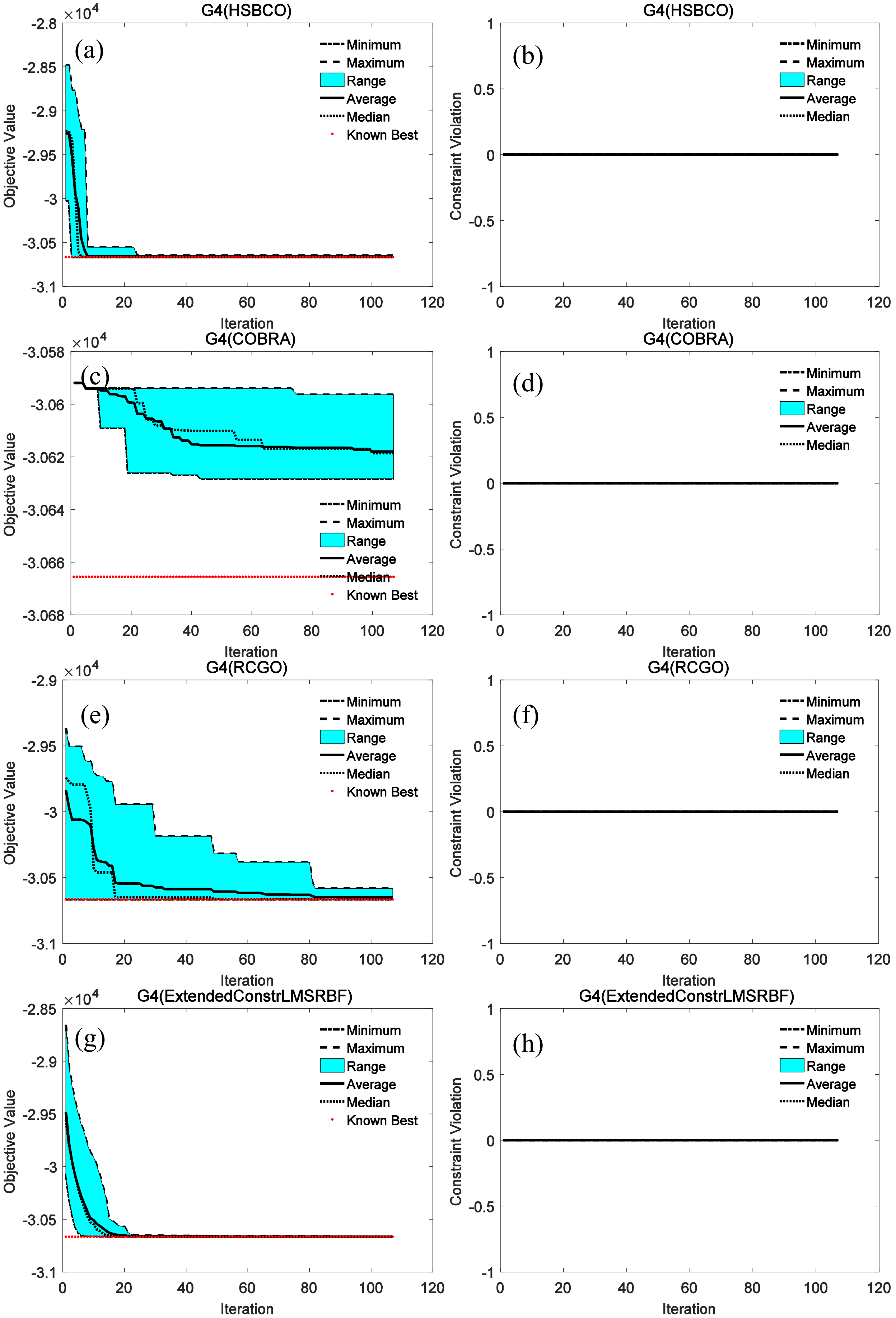

Fig. 3S. Convergence processes of objective value and constraint violation of problem G4 in 20 trials with 107 iterations (function evaluations).

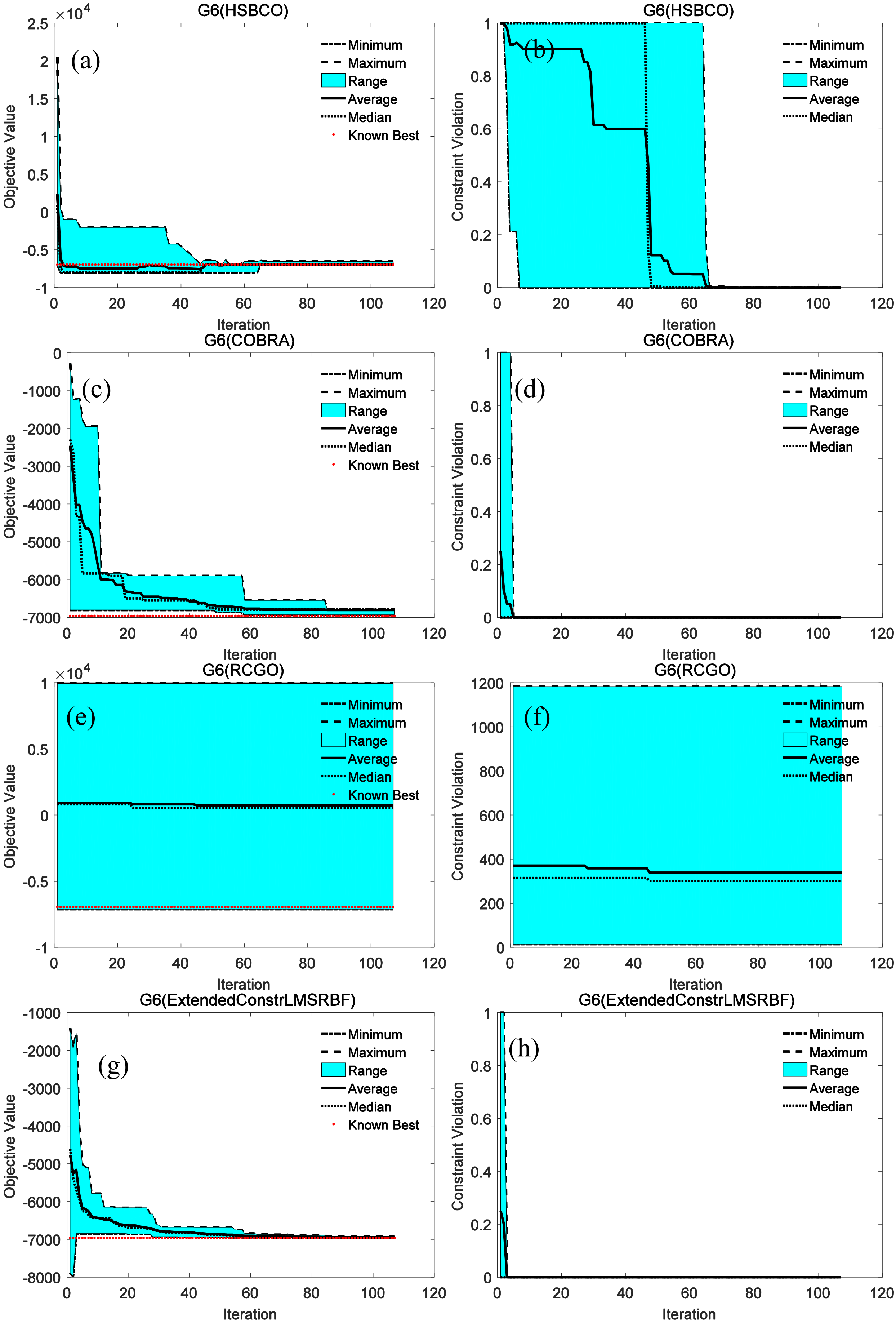

Fig. 4S. Convergence processes of objective value and constraint violation of problem G6 in 20 trials with 107 iterations (function evaluations).

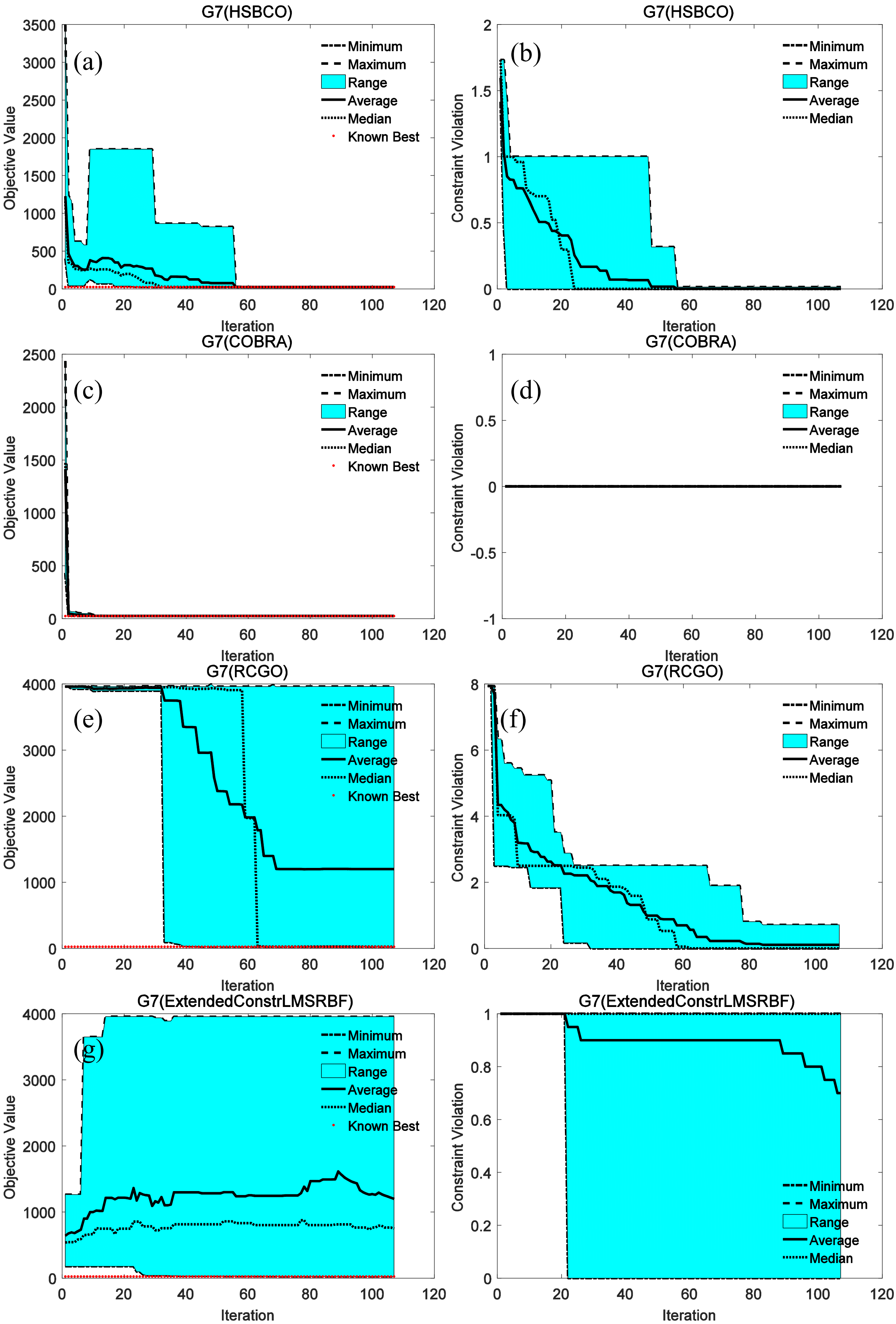

Fig. 5S. Convergence processes of objective value and constraint violation of problem G7 in 20 trials with 107 iterations (function evaluations).

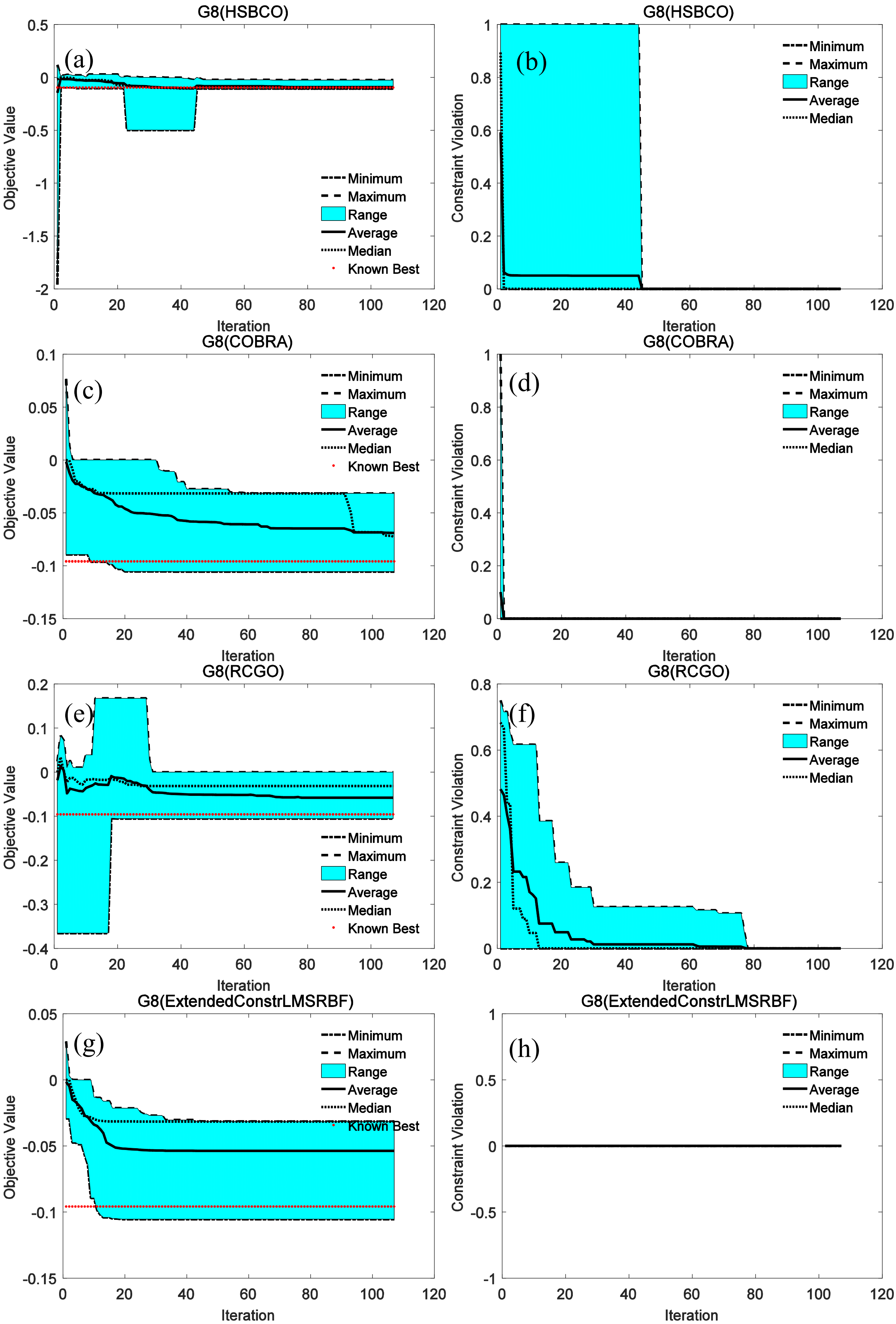

Fig. 6S. Convergence processes of objective value and constraint violation of problem G8 in 20 trials with 107 iterations (function evaluations).

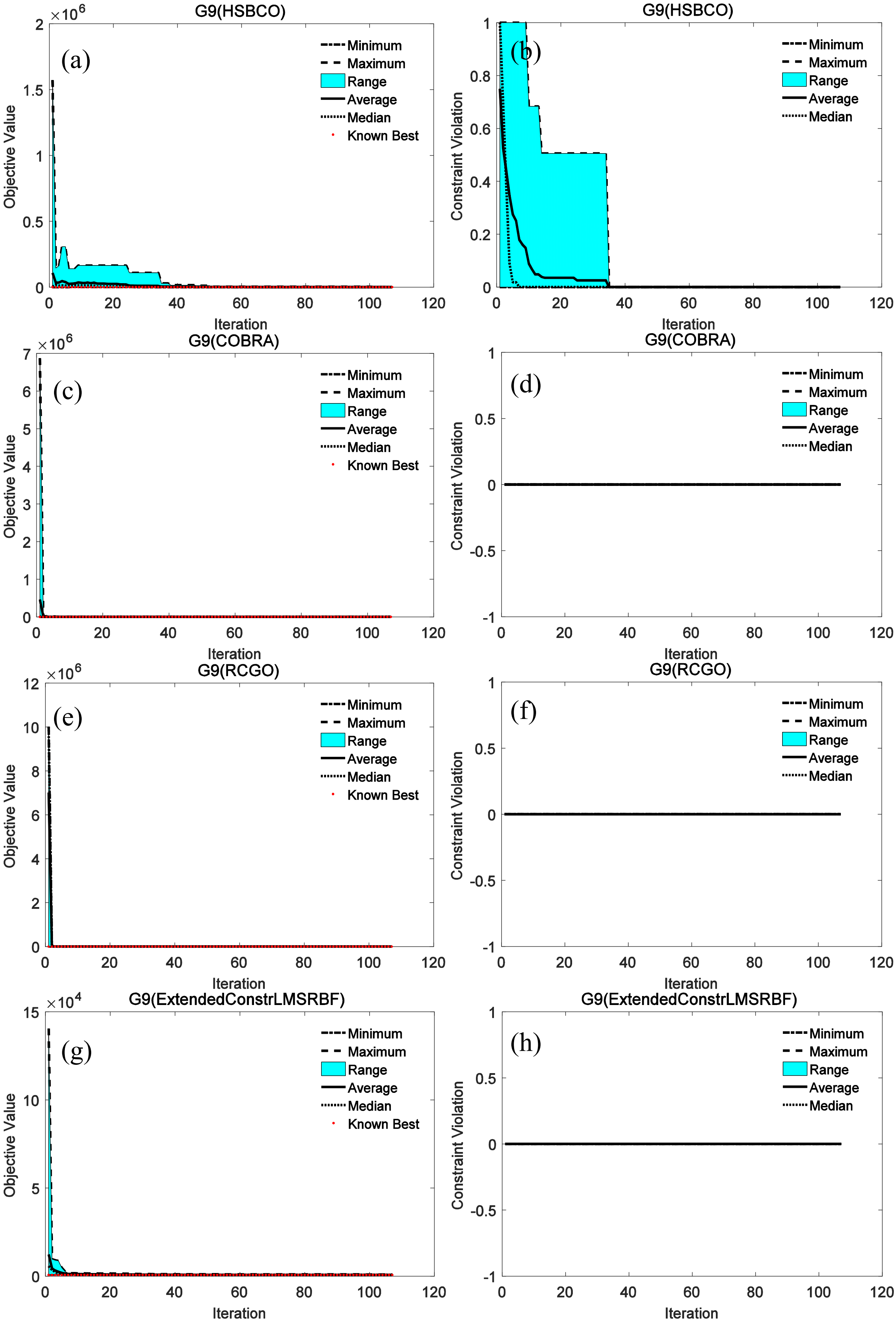

Fig. 7S. Convergence processes of objective value and constraint violation of problem G9 in 20 trials with 107 iterations (function evaluations).

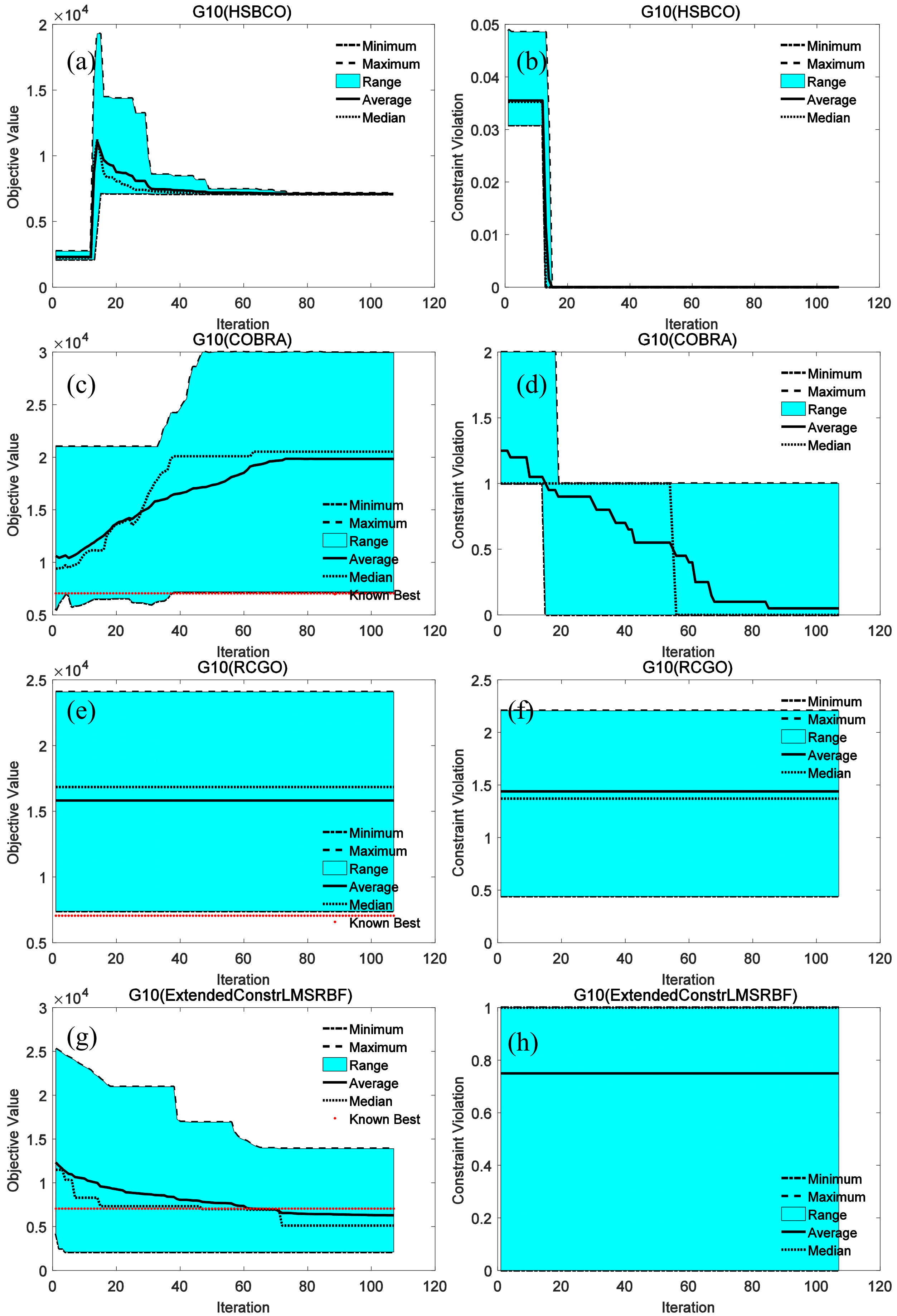

Fig. 8S. Convergence processes of objective value and constraint violation of problem G10 in 20 trials with 107 iterations (function evaluations).

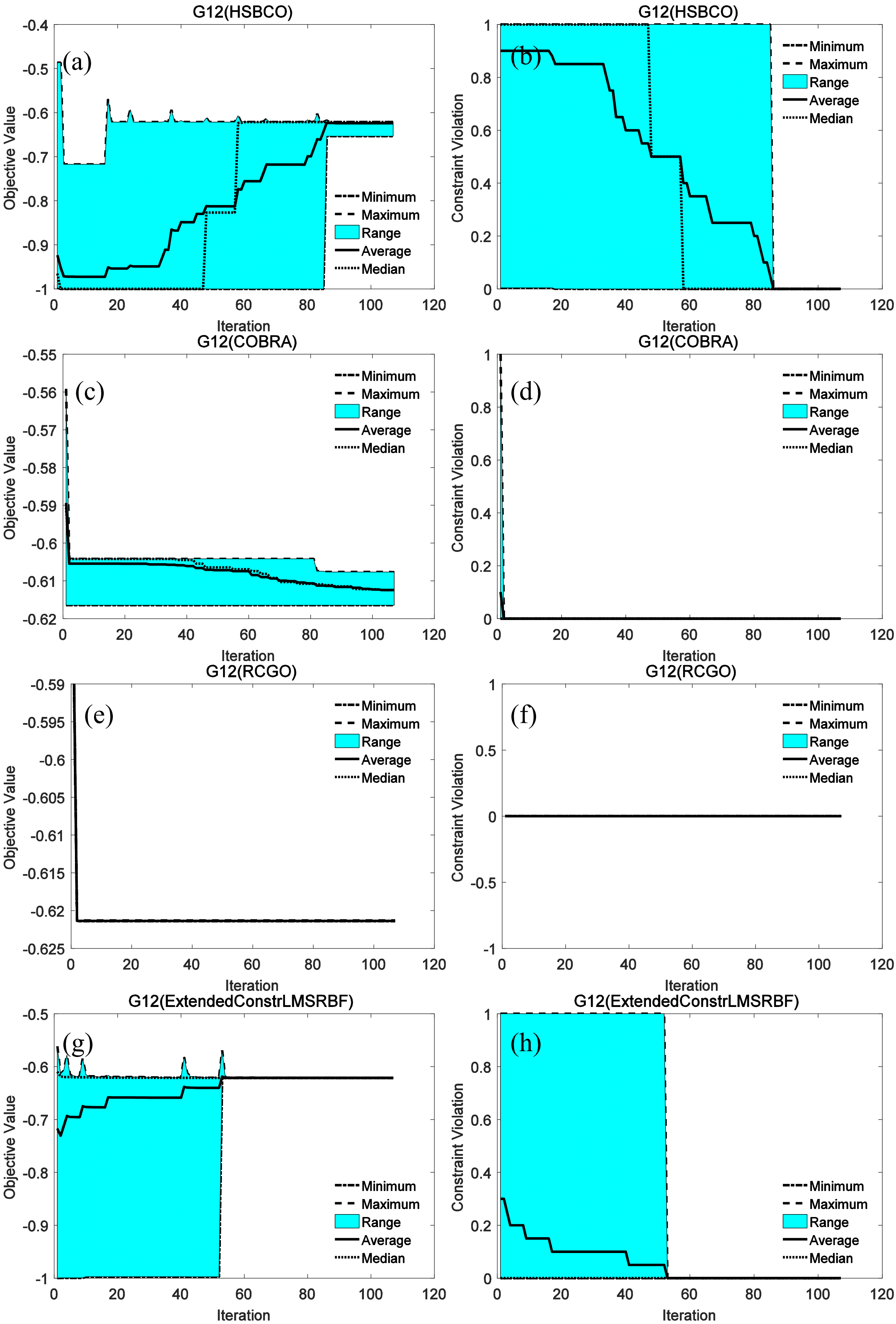

Fig. 9S. Convergence processes of objective value and constraint violation of problem G12 in 20 trials with 107 iterations (function evaluations).

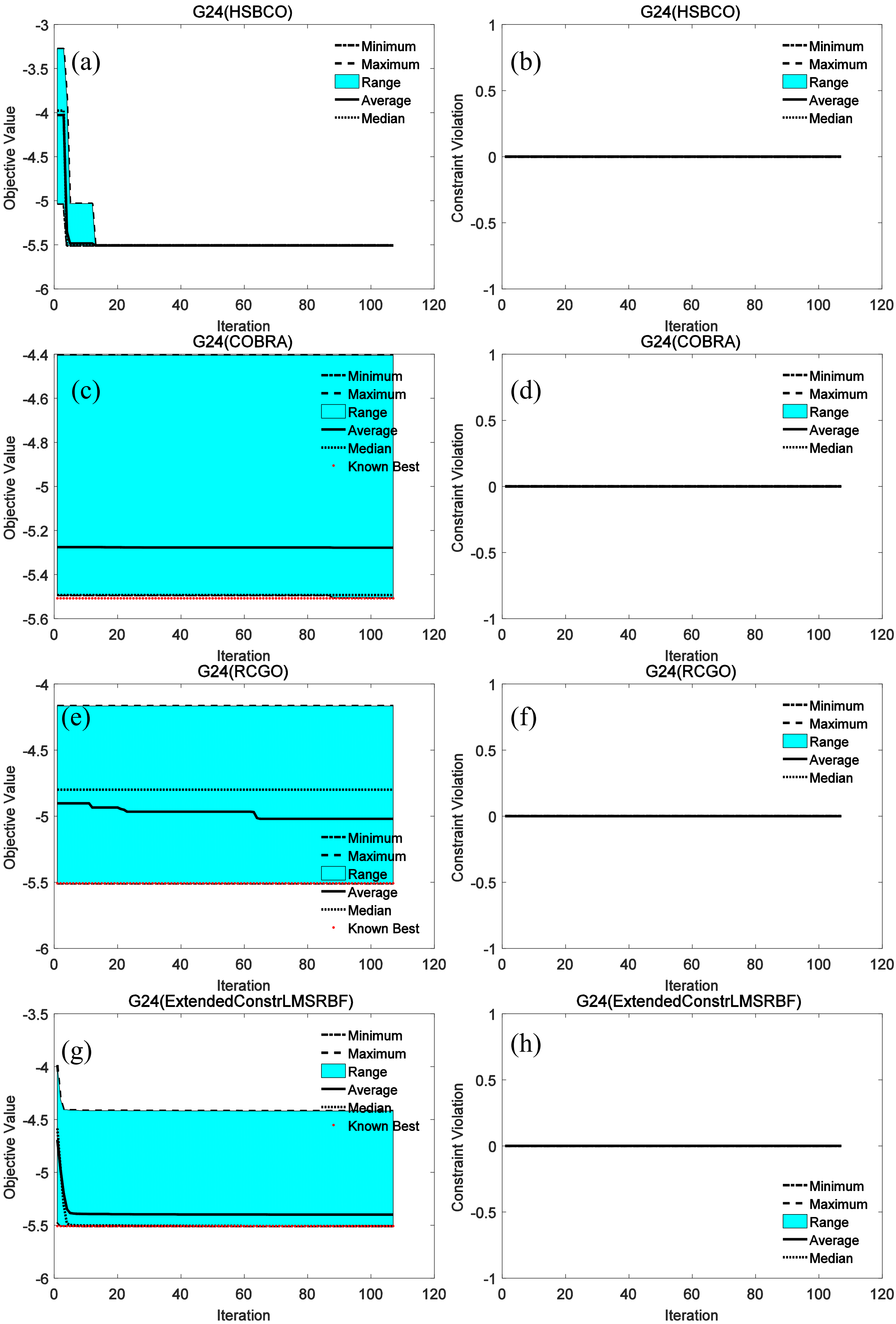

Fig. 10S. Convergence processes of objective value and constraint violation of problem G24 in 20 trials with 107 iterations (function evaluations).

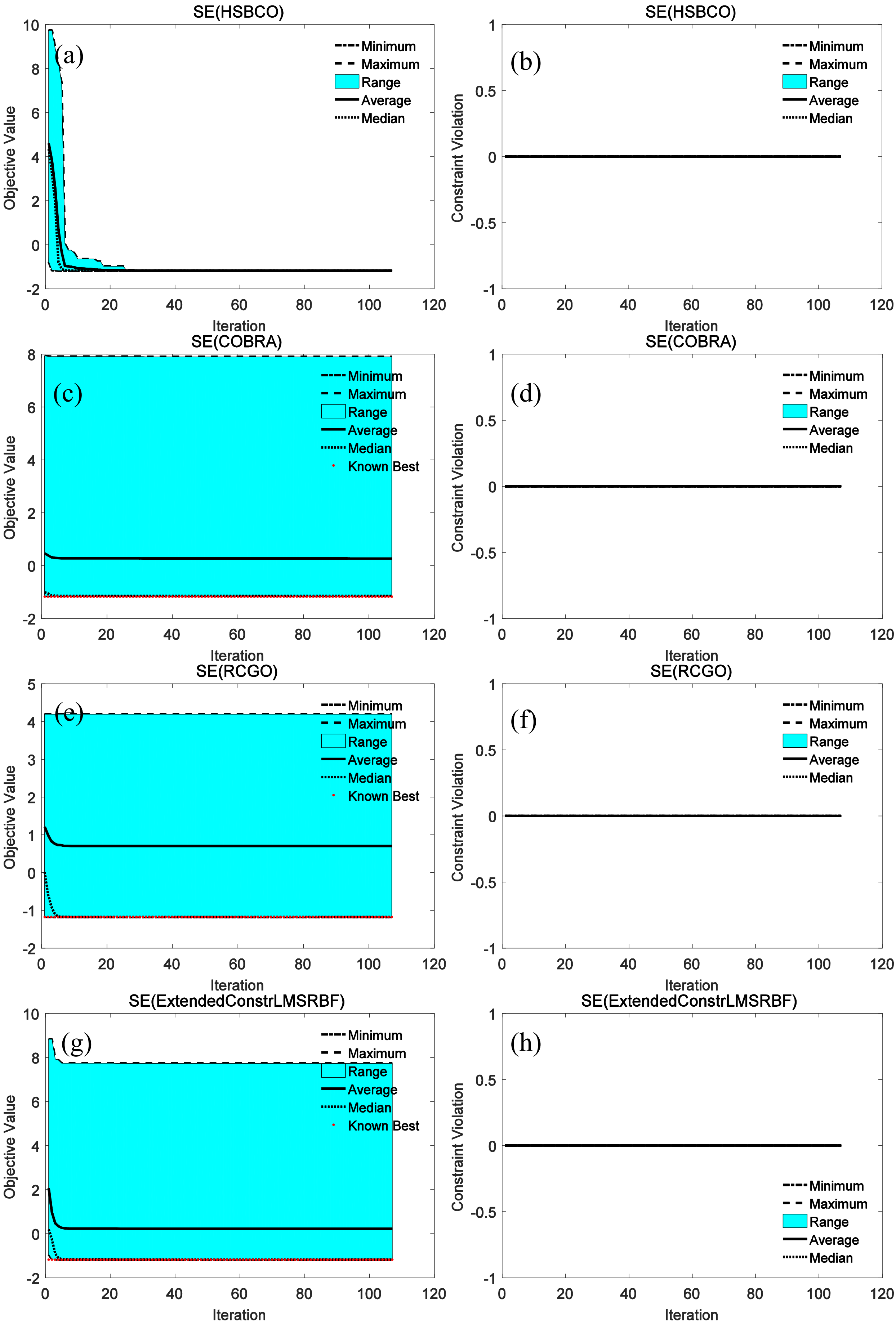

Fig. 11S. Convergence processes of objective value and constraint violation of problem SE in 20 trials with 107 iterations (function evaluations).

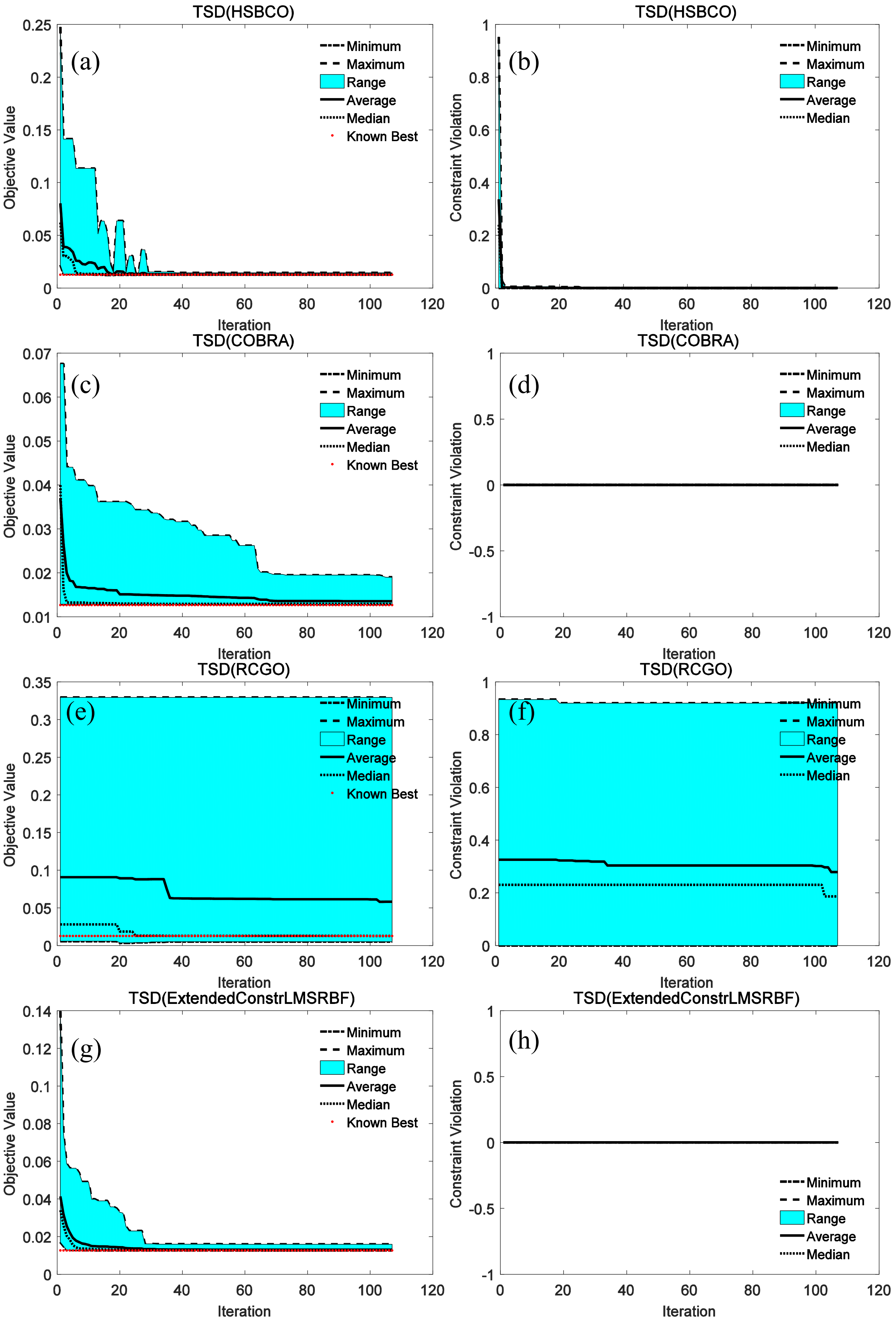

Fig. 12S. Convergence processes of objective value and constraint violation of problem TSD in 20 trials with 107 iterations (function evaluations).

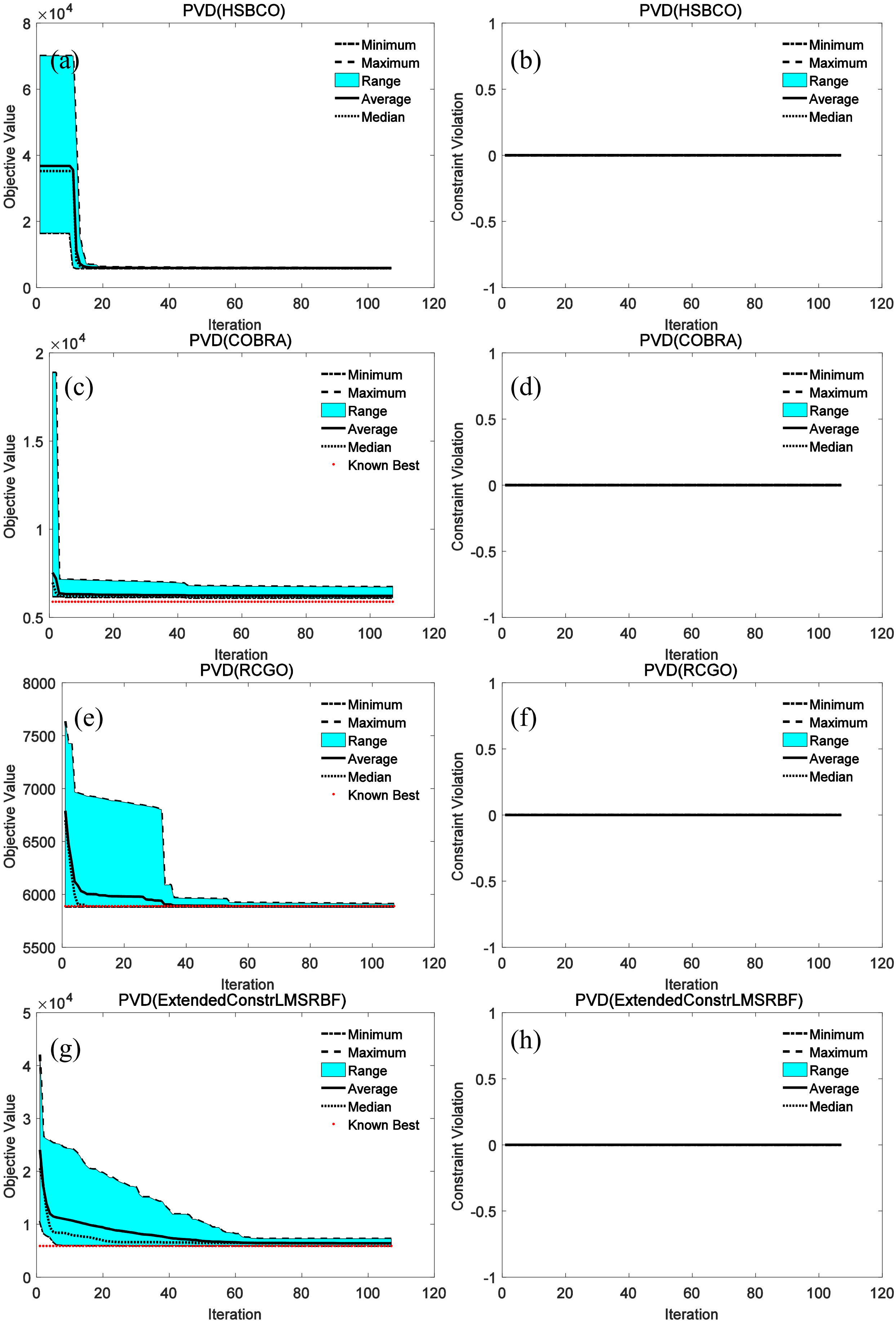

Fig. 13S. Convergence processes of objective value and constraint violation of problem PVD in 20 trials with 107 iterations (function evaluations).

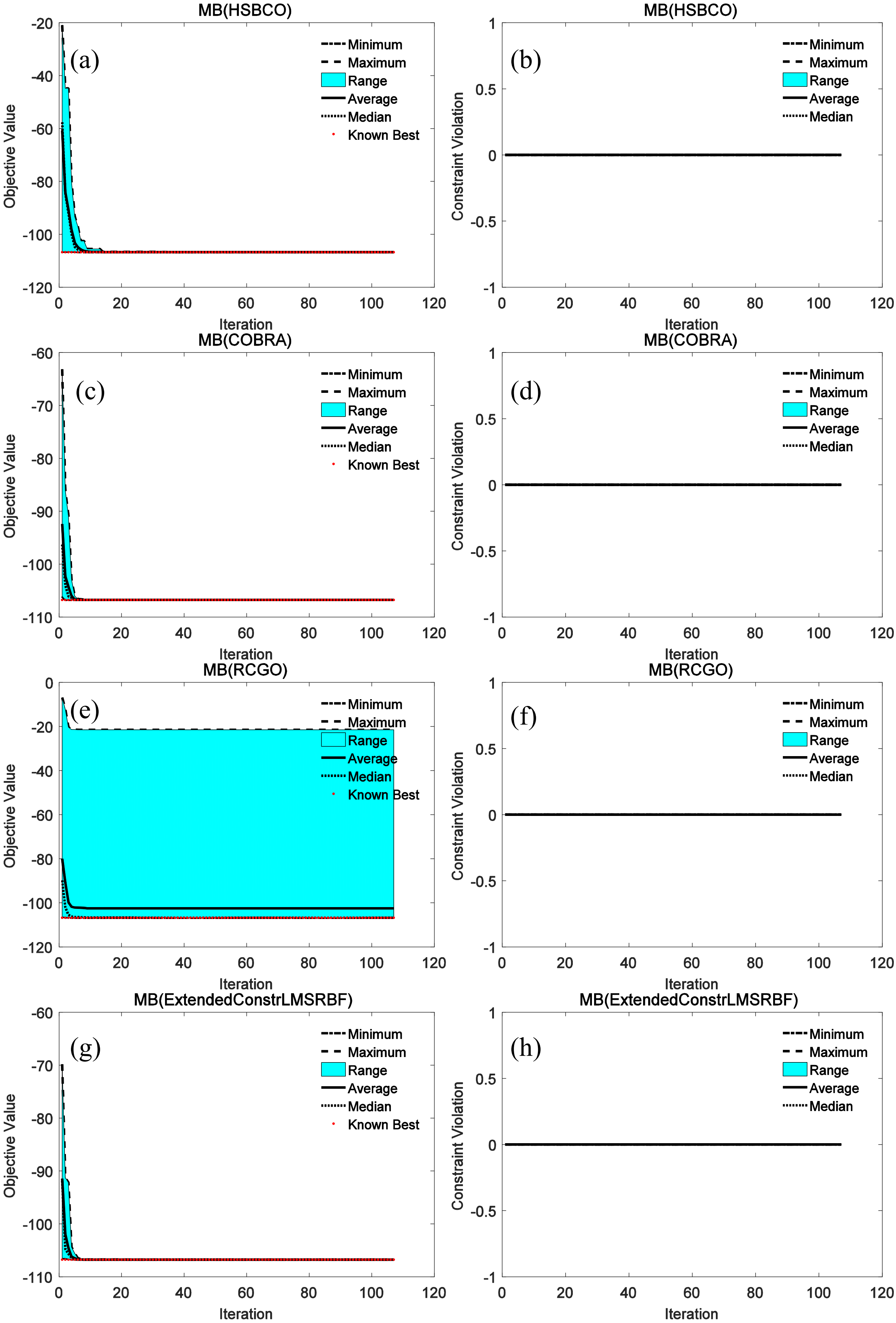

Fig. 14S. Convergence processes of objective value and constraint violation of problem MB in 20 trials with 107 iterations (function evaluations).

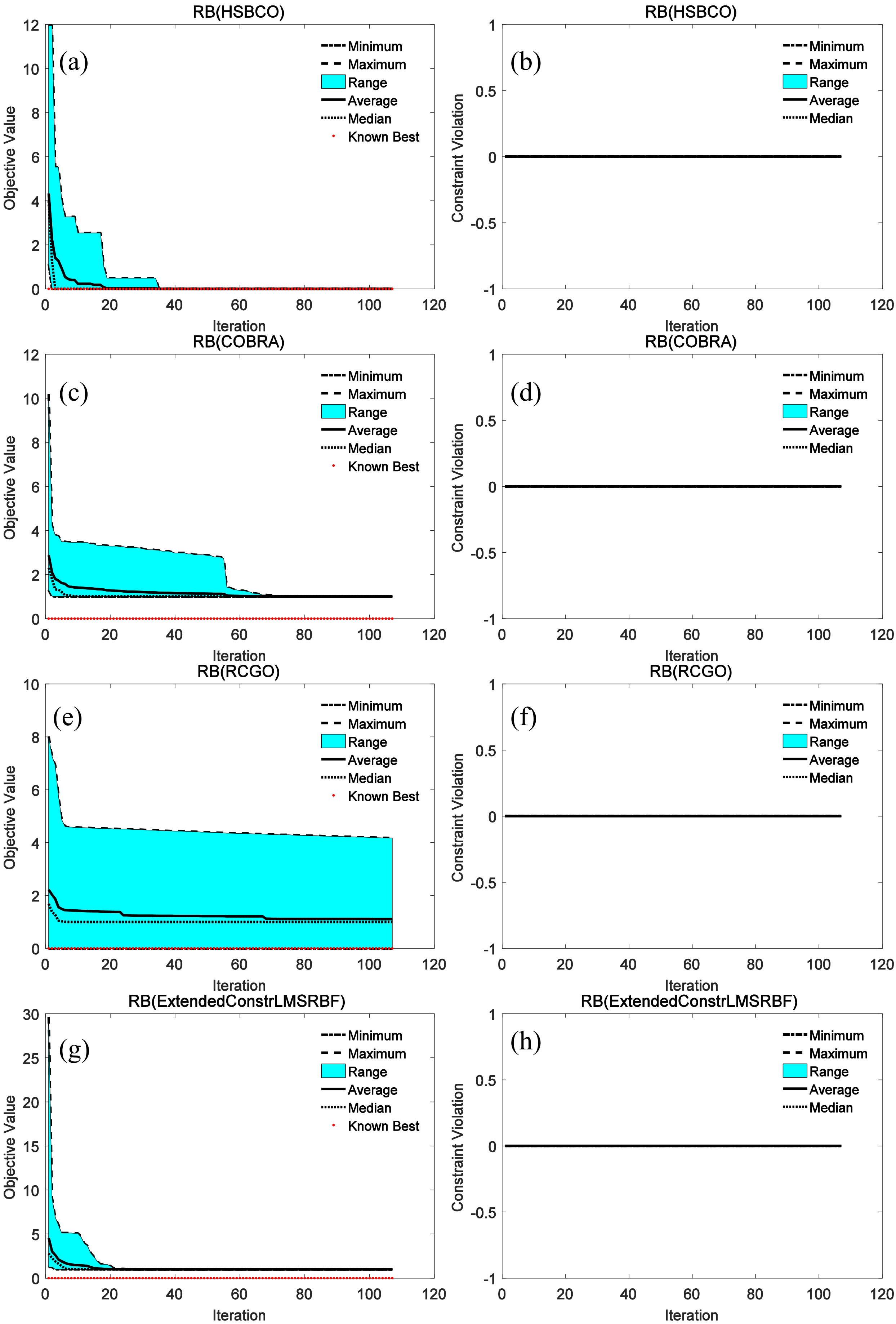

Fig.15S. Convergence processes of objective value and constraint violation of problem RB in 20 trials with 107 iterations (function evaluations).

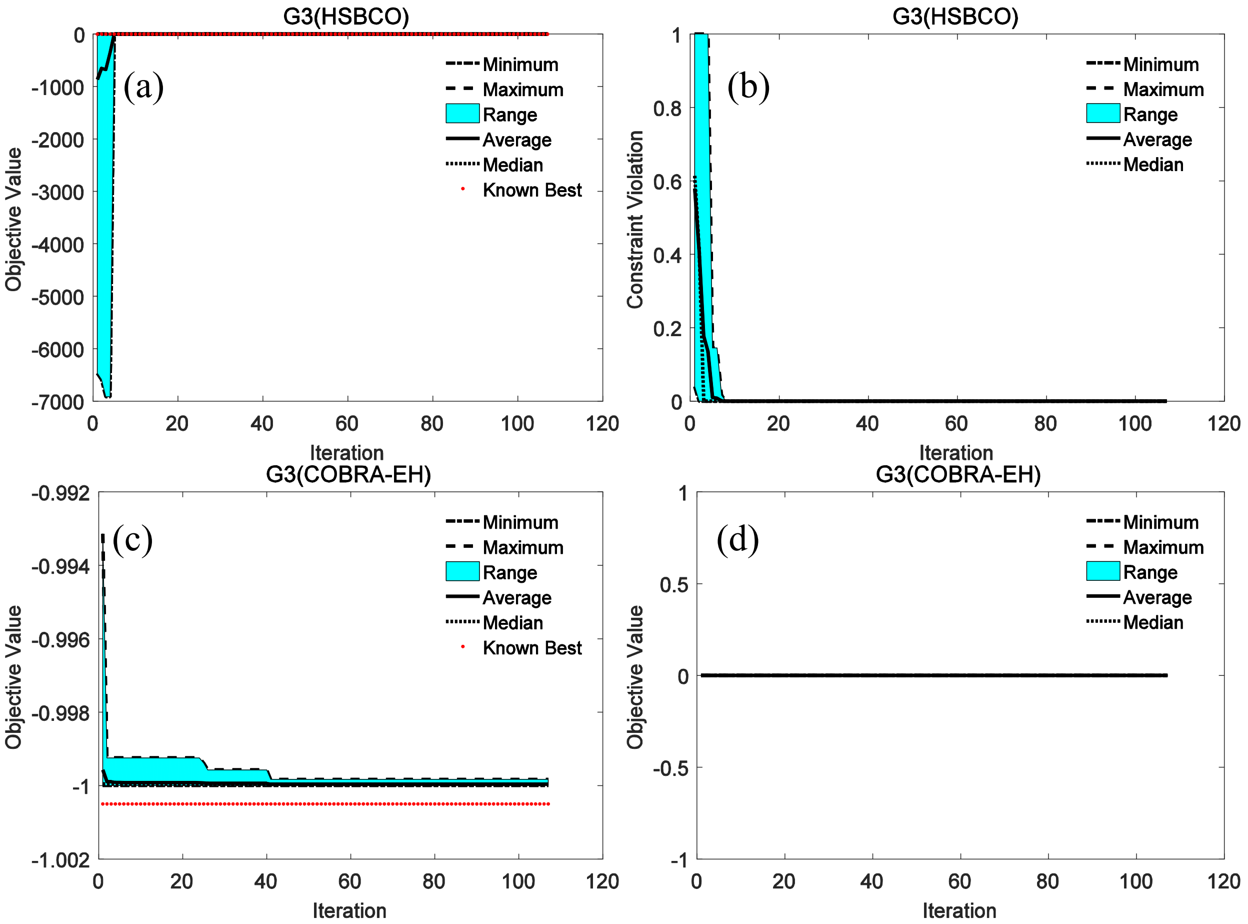

Fig. 16S. Convergence processes of objective value and constraint violation of problem G3 in 20 trials with 107 iterations(function evaluations).

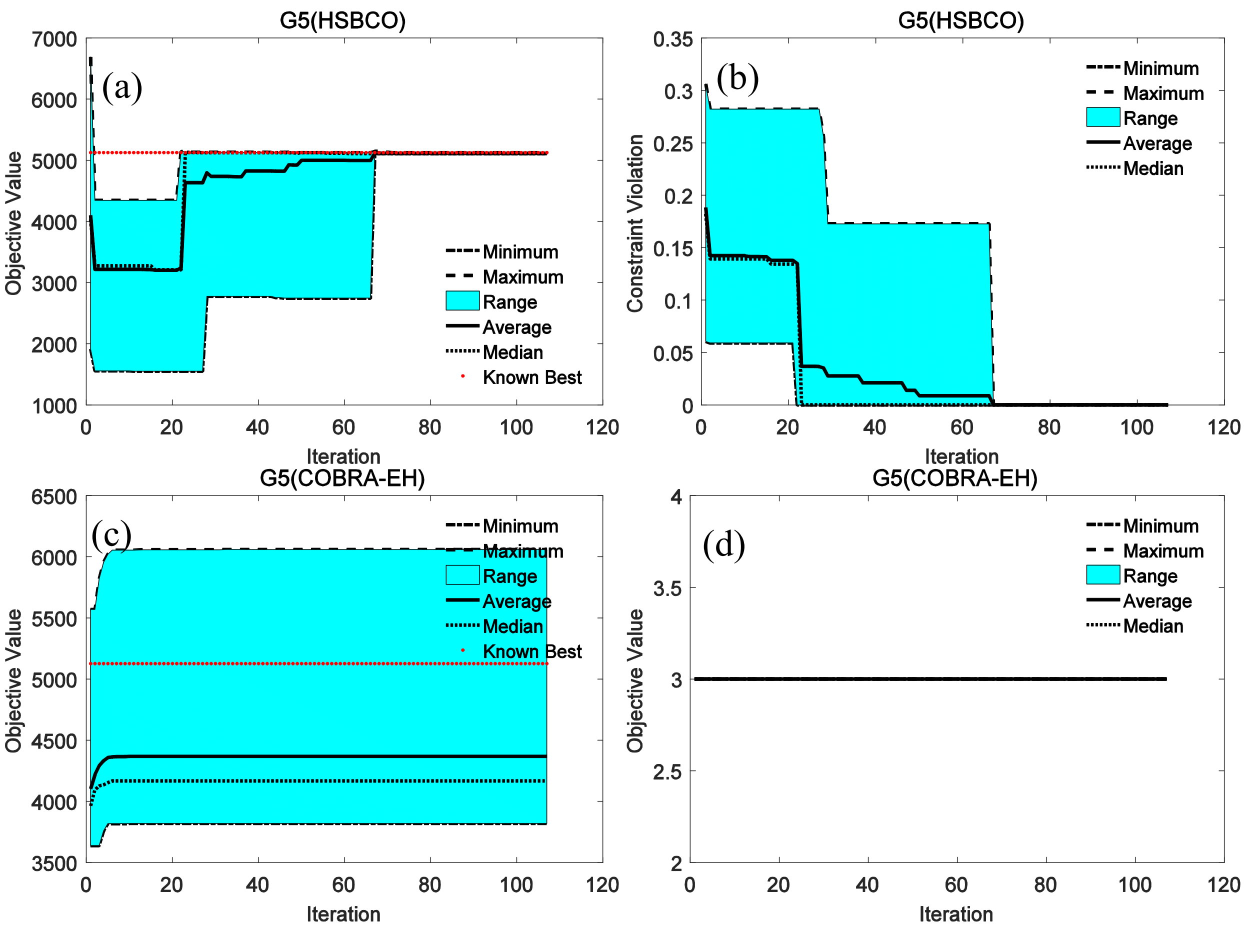

Fig. 17S. Convergence processes of objective value and constraint violation of problem G5 in 20 trials with 107 iterations (function evaluations).

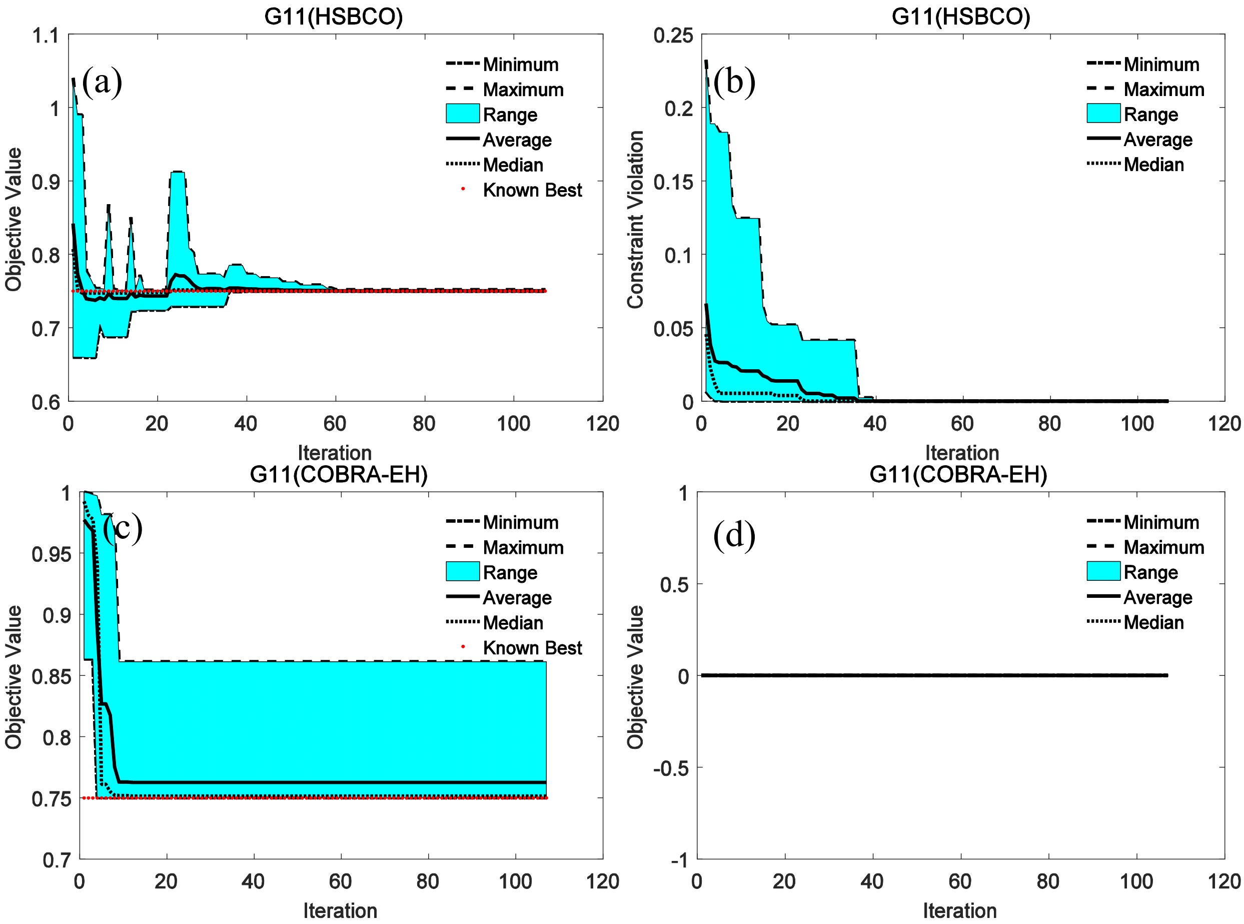

Fig. 18S. Convergence processes of objective value and constraint violation of problem G11 in 20 trials with 107 iterations (function evaluations).

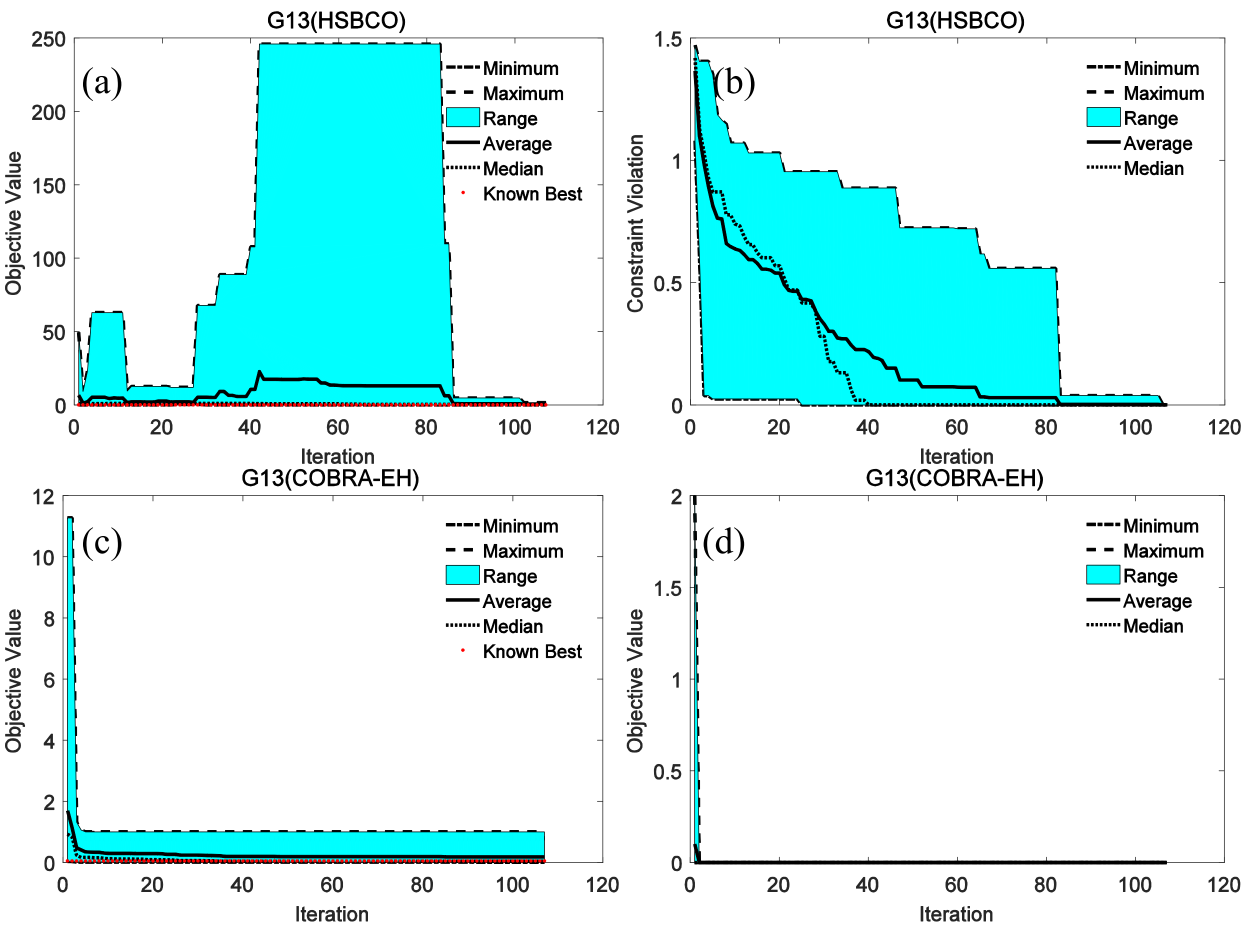

Fig. 19S. Convergence processes of objective value and constraint violation of problem G13 in 20 trials with 107 iterations (function evaluations).

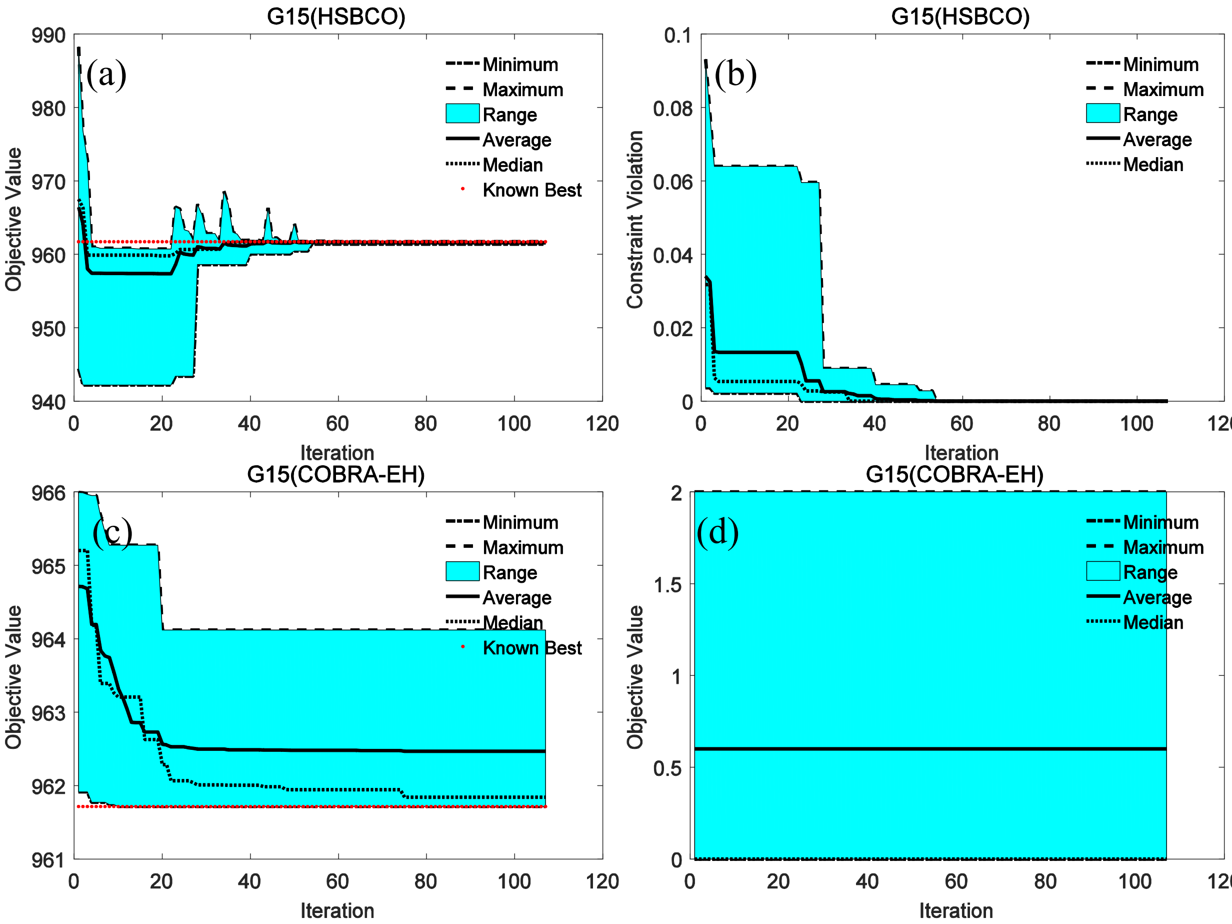

Fig. 20S. Convergence processes of objective value and constraint violation of problem G15 in 20 trials with 107 iterations (function evaluations).

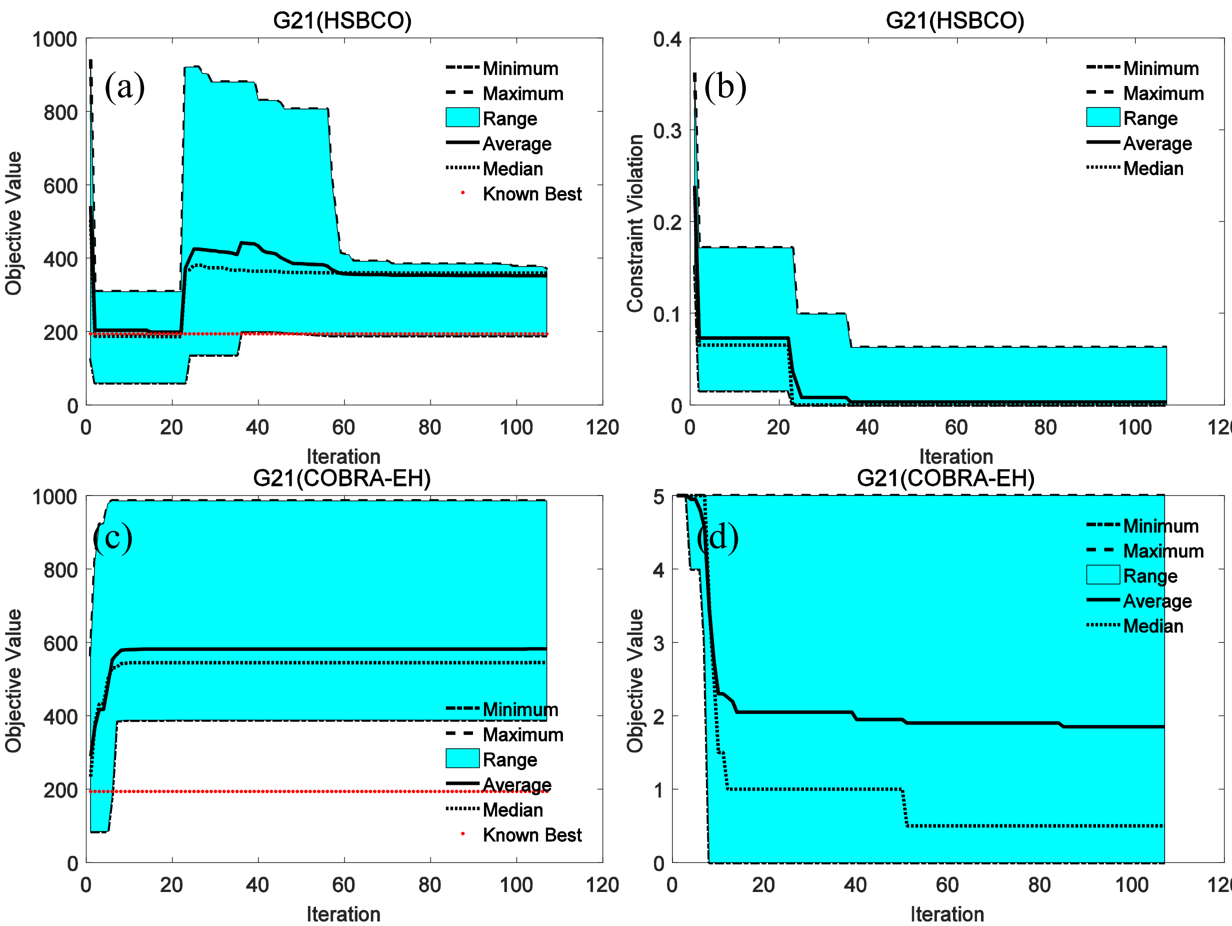

Fig. 21S. Convergence processes of objective value and constraint violation of problem G15 in 20 trials with 107 iterations (function evaluations).

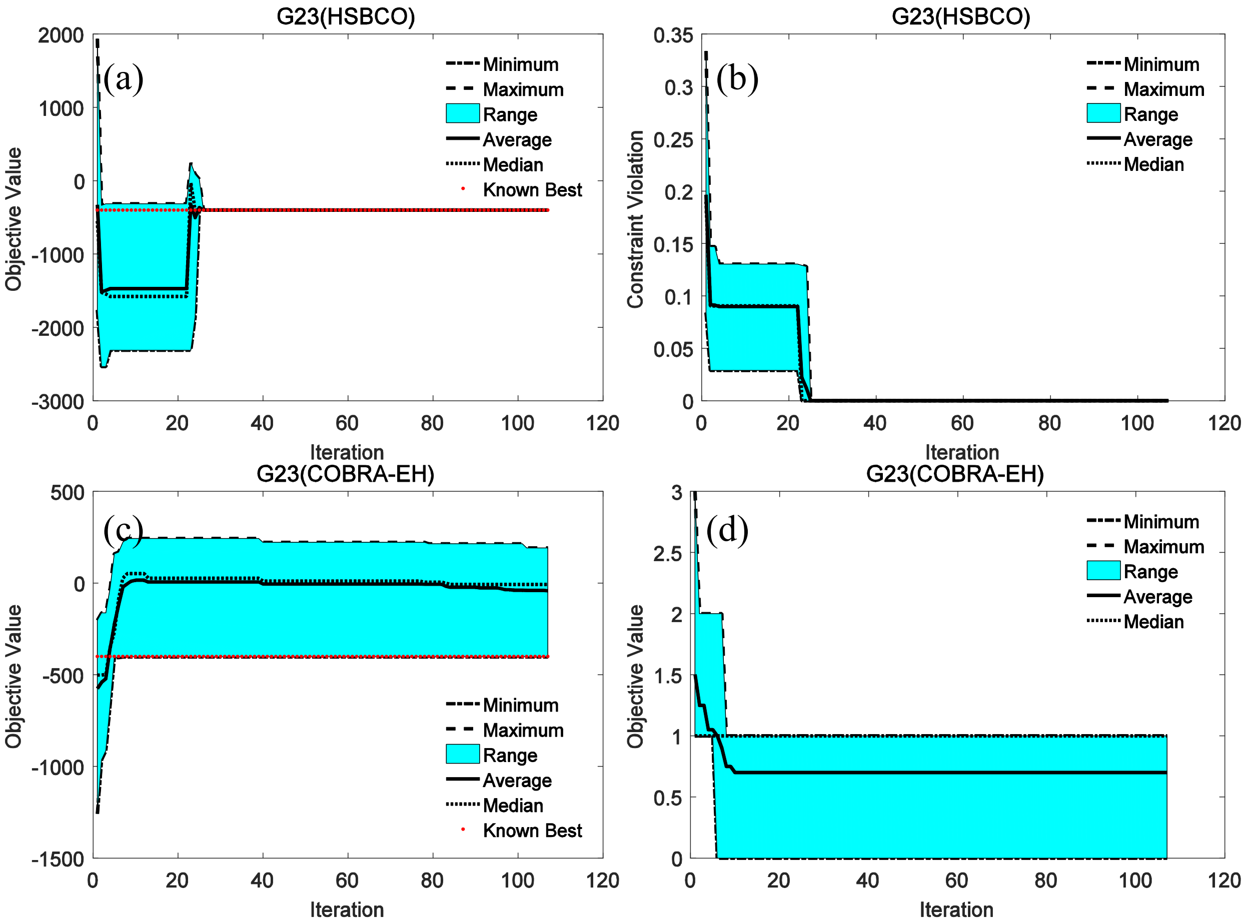

Fig. 22S. Convergence processes of objective value and constraint violation of problem G23 in 20 trials with 107 iterations (function evaluations).

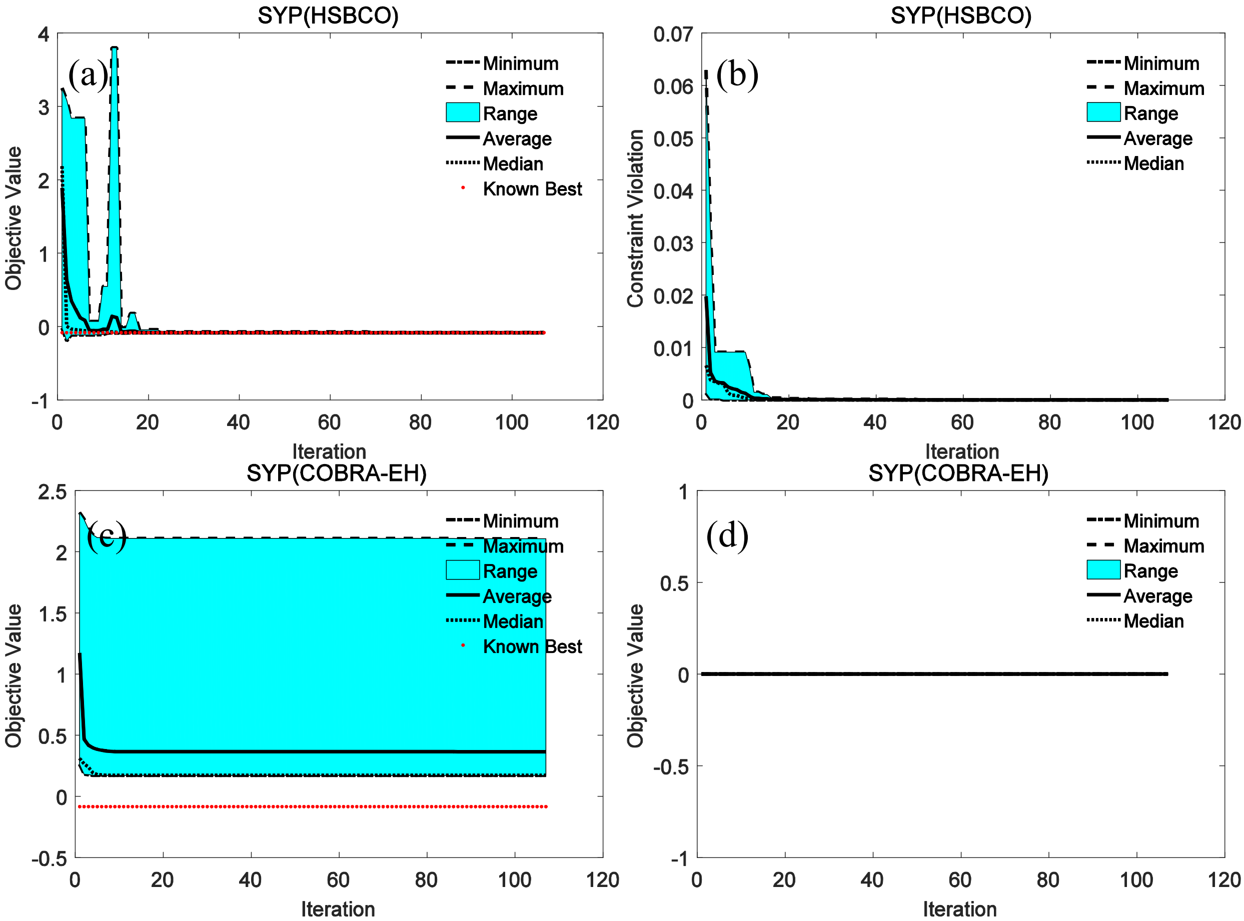

Fig. 23S. Convergence processes of objective value and constraint violation of problem SYP in 20 trials with 107 iterations (function evaluations).

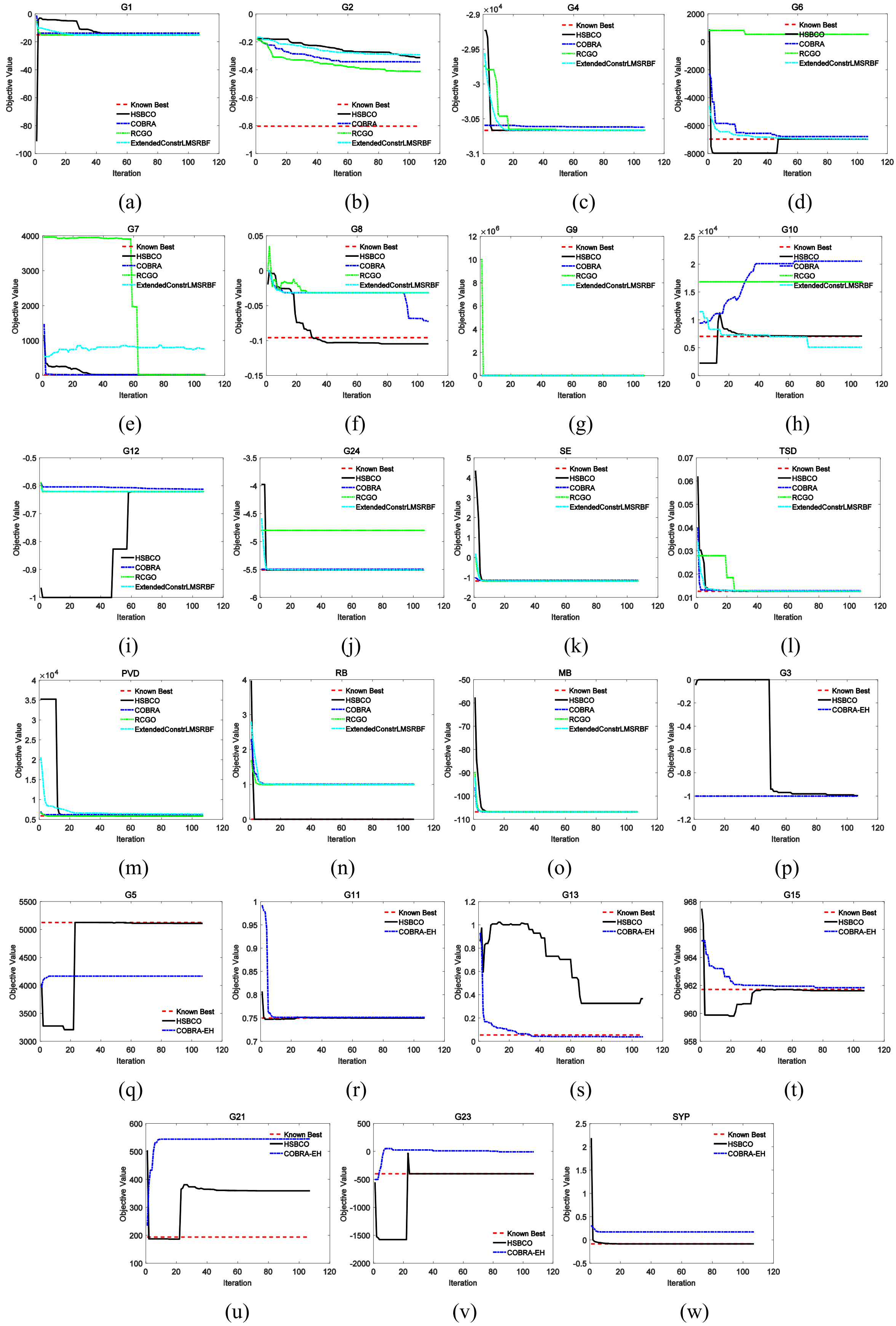

Fig. 24S. Comparison of convergences of the four algorithms on the test problems. The lines in the sub-figures reflect the changes of medians of the best solutions obtained at the iterations

Table S2 The average evaluation number for the feasible solution and the optimum found in 20 trials of the test problems

problems	HSBCO	COBRA	RCGO	Extended ConstrLMSRBF	COBRA-EH
G1	54.2	20.55	49	87.85	/
G2	98.6	65.3	81.1	103.65	/
G3	59.45	/	/	/	5.4
G4	7.15	43.25	36.15	91.9	/
G5	60.95	/	/	/	41.95
G6	48	49.9	4.4	83.6	/
G7	60.7	27.95	80	69.35	/
G8	59.4	57.15	37.6	47.35	/
G9	75.9	42.85	43.7	100.55	/
G10	54.6	82.55	107	107	/
G11	47.3	/	/	/	7.1
G12	62.5	79.9	2	86.55	/
G13	72.6	/	/	/	42
G15	60.95	/	/	/	22.5
G21	73.35	/	/	/	33.85
G23	25.55	/	/	/	33.5
G24	2.95	17.45	7.05	84.25	/
SE	25.5	32.85	22.45	82.1	/
TSD	30.2	37.4	24	70.65	/
PVD	69	73.25	30.7	95.75	/
MB	17	40.2	66.55	72.7	/
RB	7.3	43.25	22.95	82.35	/
SYP	28.55	/	/	/	27.8

III. Discussion of Adaptive Feasible Threshold Method

As discuss in the main paper, estimation error of the constraints may cause misjudgment of some feasible solutions and infeasible solutions. One way to avoid such events is to define a robust domain around the origin with radius $r_{threshold}$ in the constraint Euclidean sub-space. Thus, any point in such domain can be regarded as a feasible solution. Generally, for some simple problems, we can set a small feasible threshold to define such a robust domain. However, it may be not suitable to set a small feasible threshold for some problems with complex constraints, because modeling errors in the surrogates are inevitable, and the robust domain reflects the allowed scope of the model error perturbations. It is usually difficult to determine the bounds of the perturbations. If the surrogates of the constraints are not accurate enough, the feasible region may disappear, and we may not find any feasible solution, as the case shown in Fig. 3R. Defining a suitably robust domain can effectively deal with the uncertainty of the model error of the surrogate and ensure the convergence of the optimization process. Thus, if the union Euclidean distance of a sampling point is smaller than $r_{threshold}$, then this point can be regarded as a feasible point. However, how to set the threshold $r_{threshold}$ is an important issue. For problems such as the test problems considered in this work, if we have some *a priori* information, the threshold $r_{threshold}$ can be set as a constant, but for most practical problems, it is difficult to obtain the *a priori* information, so the threshold must be adjusted dynamically according to the model errors of the surrogates or consecutive sampled results in the optimization process. There are many machine learning methods, such as support vector machines, fuzzy logic systems and neural networks, that can be used to select a suitable value for $r_{threshold}$, but we present a data-driven control method to adjust this parameter dynamically.

Generally, parameter $r_{threshold}$ should be adjusted with the model errors of the surrogates of the constraints; in other words, it should be adjusted with the estimation error of the union Euclidian distance $\hat{r}$. Denote $\hat{r}_{true}, \hat{r}_{surrogate}$ as the true union distance and its estimation, respectively; then $\Delta\hat{r} = \hat{r}_{true} - \hat{r}_{surrogate}$ is the estimation error. Since the aim of setting the robust domain is to increase the sampling in the feasible region so as to improve the simulation performances of the surrogates of the constraints, the change of $\Delta\hat{r}$ can be regarded as an unknown nonlinear system with respect to $r_{threshold}$, i.e., it can be expressed as

$$-\left|\Delta\hat{r}_{k+1}\right| = f(-\left|\Delta\hat{r}_{k}\right|, r_{threshold,k}) \tag{1}$$

Where $\Delta\hat{r}_k$, $r_{threshold,k}$ are the estimation error and the threshold of the sample at the k-th iteration, $\Delta\hat{r}_{k+1}$ is the prediction of the estimation error of the next iteration, and $f(\cdot)$ is an unknown nonlinear function. System (1) means that the change of the estimation error is driven by the threshold. For convenience, we can express (1) in the following common form:

$$y_{k+1} = f(y_k, u_k) \tag{2}$$

Then, it has $y_k = -\left|\Delta\hat{r}_k\right|$ and $u_k = r_{threshold,k}$.

Assume that there exists a constant $b > 0$ to make system (2) satisfy the condition $\left|y_{k+1} - y_k\right| \le b\left|u_k - u_{k-1}\right|$ for any k and $\Delta u_k = u_k - u_{k-1} \ne 0$. Hou proved that such a system can be transformed into the following pseudo-partial derivative (PPD) equivalent description [Res 4]:

$$\Delta y_{k+1} = y_{k+1} - y_k = \phi_k \cdot \Delta u_k \tag{3}$$

where ϕ_k is an unknown PPD function, and must be estimated online using the sampled data. Denote $\hat{\phi}_k$ as the estimation of ϕ_k. Obviously, in our case, the output of the system should converge to zero; i.e., $\lim_{k\to\infty} y_k = 0$, so according to reference [Res 4], we can use the following update law and control law to ensure the convergence of system (3):

$$\hat{\phi}_k = \hat{\phi}_{k-1} + \frac{\eta[\Delta y_k - \hat{\phi}_k \cdot \Delta u_{k-1}] \cdot \Delta u_{k-1}}{1 + \left|\Delta u_{k-1}\right|^2} \tag{4}$$

$$u_k = u_{k-1} + \frac{\rho \cdot \hat{\phi}_k \cdot [0 - y_k]}{1 + \left|\hat{\phi}_k\right|^2} \tag{5}$$

where $0 < \eta, \rho < 1$. Thus, the threshold $r_{threshold}$ can be adjusted by (5) according to the estimation error of the union Euclidian distance $\Delta\hat{r}_k$. Theoretically, $r_{threshold}$ and $\Delta\hat{r}_k$ will necessarily finally converge to zero. However, in fact, besides the threshold parameter, other factors, such as the interior optimizer used and the model error of the surrogate of the objective function also influence the sampling of the optimization process significantly. Therefore, system (1) is not sufficient, so sometimes $r_{threshold}$ may not converge to zero, and must be limited when it becomes large. In this case, we must introduce the following rules:

Rule 1: if $\Delta\hat{r}_k = 0$, then $u_k = 0$.

Rule 2: if $u_k > \mu$, then $u_k = \mu$.

where μ is a predefined positive constant, and is not more than the number of constraints.

Although parameter $r_{threshold}$ is fixed as 0.0001 for all the test problems considered, the results indicate that the proposed method can still achieve good performance for most problems, and although COBRA,RCGO and Extended ConstrLMSRBF do not consider the model error issue of the constraint surrogate, they still achieved similar results. Perhaps the constraints of those test problems are easy to model. but for the G9 case, the proposed method cannot converge to a small neighborhood of the global optimal solution. Therefore, to demonstrate the effect of the adaptive threshold on the optimization result, we repeated the proposed method on G9, in which threshold $r_{threshold}$ is adjusted using (4) and (5). In this experiment, η, ρ, μ are set as 0.1, 0.1, 1, respectively. The initial values of $r_{threshold,k}$, $\hat{\phi}_k$ $\Delta\hat{r}_k$ are set as 0, 1, 1, respectively. The results are illustrated in Table R1 and Fig. 25S.

Table S2. Objective value of solutions found in 20 trials, each with 107 iterations, on G9 using HSBCO with fixed threshold and adaptive threshold

Test problem	Known best	$r_{threshold}$	Best	Median	Worst	Mean	STD
G9	680.6300574	fixed	724.1146	962.5548	1991.257	966.9856	284.9889
		adaptive	692.8128	826.1952	1091.228	834.9204	89.79468

From Table S2, it can be seen that although the best solution still did not converge to the true global optimal solution when an adaptive threshold was used in HSBCO, the best, worst and median values are all better than those values with a fixed threshold, and from Fig. 25S (b) and (d), we can also observe that the optimization process can sample a feasible solution at each iteration when using an adaptive threshold in most cases, while when using the fixed threshold, the optimization process often find a feasible solution only after several iterations. From Fig. 25S (e) and (f), it can be observed that the threshold increases with the estimation error of the union Euclidean distance in most cases. This situation is reasonable, because if the estimation error is large, we need to increase the robust domain to cover the unmodeled region to cover the better solution. Although using the adaptive threshold can help the proposed method to achieve better performance on the G9 problem, it should still be pointed out that HSBCO with the proposed adaptive threshold is expected to achieve good across a wide variety of problems, because the estimation error of the union Euclidean distance also significantly depends on the interior optimizer and the estimation error of the objective function. In other words, system (1) is not completely sufficient. Therefore, we need to consider much more relevant information to determine the threshold, and will research a better method to adaptively adjust this parameter in the future, which we hope will contribute further to surrogate-based optimization.

[Res 4]. *Zhongsheng Hou and Shangtai Jin. A Novel Data-Driven Control Approach for a Class of Discrete-Time Nonlinear Systems. IEEE Transactions on Control Systems Technology, vol. 19, no. 6, pp. 1549-1558, 2011.*

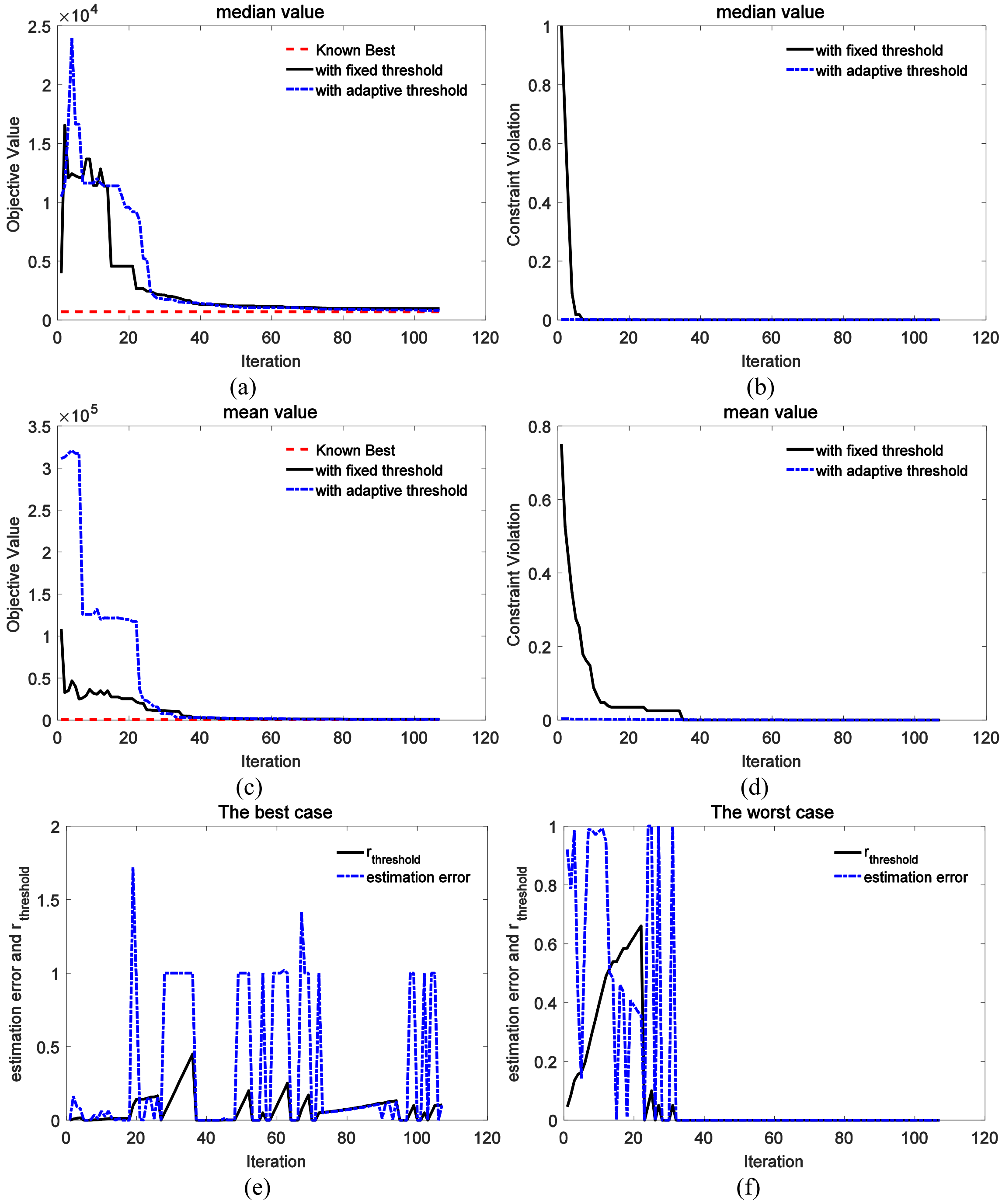

Fig. 25S. Median and mean values of the objective and constraint violation obtained by HSBCO with fixed threshold and with adaptive threshold, and the changes of model error of the union Euclidian distance at the sampled point and adaptive threshold $r_{threshold}$. (a) and (b) illustrate the median values of the objective and constraint violation, respectively; (c) and (d) shows the mean values of the objective and constraint violation, respectively, and (e) and (f) illustrate the adaptive threshold and estimation error of the union Euclidian distance.