Supplementary Materials of "Decision Space Partition Based Surrogate-Assisted Evolutionary Algorithm for Expensive Optimization"

Yuanchao Liu[a,b], Jianchang Liu[a,b,*], Shubin Tan[a,b]

[a] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China
[b] College of Information Science and Engineering, Northeastern University, Shenyang, China

1. The effect of the number of population size in PSO and the basement search algorithm

In the DSP-SAEA, classical particle swarm optimization (PSO) is adopted as the baseline search method. Therefore, the following two interesting questions are considered. The first one is whether adjusting the population size in the PSO for different number of dimensional problems can obtain better results. The second one is whether employing recently proposed PSO variant as the basement algorithm of DSP-SAEA can achieve better performance. To this end, two variants are designed, namely, DSP-SAEA-A and DSP-SAEA-CSO. In the DSP-SAEA-A, PSO is applied as the baseline search algorithm. Moreover, the population size in the PSO is set as $2 \times D$, where D is the number of variables for the test problem. In the DSP-SAEA-CSO, competitive swarm optimizer (CSO) (Cheng & Jin, 2015) is adopted as the baseline search algorithm, where CSO is a variant of PSO. In addition, the basic settings of CSO are based on the corresponding paper.

DSP-SAEA, DSP-SAEA-A and DSP-SAEA-CSO are tested on each benchmark problem with $10D$, $20D$ and $30D$. Each algorithm is independently run for 31 times on each test problem, and the Mann-Whitney U test with a significance level of 0.05 is applied to compare the significant difference between the compared methods. Moreover, each method under comparison starts with $5 \times D$ exact function evaluations and terminates after $11 \times D$ exact function evaluations are exhausted. Table 1 gives the statistical results achieved by each method.

From Table 1, we can find that DSP-SAEA-A and DSP-SAEA-CSO perform better than DSP-SAEA slightly. This is because the population size is adjusted based on the dimensions in DSP-SAEA-A. In addition, a more powerful search algorithm is applied in DSP-SAEA-CSO. Thus, competitive results can be obtained by DSP-SAEA-A and DSP-SAEA-CSO. In the future, we plan to design an adaptive strategy and a variant of PSO to further improve the performance of the DSP-SAEA.

2. Comparative Experiments on CEC2015

To further test the performance of the proposed, we compare DSP-SAEA with GSGA (Cai et al., 2020), ESAO (Wang et al., 2019), CAL-SAPSO (Wang et al., 2017) and GPEME (Liu et al., 2014) on CEC2015 (Liang et al., 2014). Each test problem has two different dimensions, i.e., $10D$ and $30D$. Each algorithm is independently run for 31 times on each test problem, and the Mann-Whitney U test with a significance level of 0.05 is applied to compare the significant difference between the compared methods. Moreover, each method under comparison starts with $5 \times D$ exact function evaluations and terminates after $11 \times D$ exact function evaluations are exhausted. Table 2 gives the statistical results achieved by each method.

From Table 2, one can find that DSP-SAEA is significantly better than GSGA on 16 out of 30 test problems. In other words, DSP-SAEA wins GSGA in 53.3% of the test problems, and losses GSGA in

*Corresponding author
Email addresses: Yuanchaoliu@126.com (Yuanchao Liu), liujianchang@ise.neu.edu.cn (Jianchang Liu), tanshubin@ise.neu.edu.cn (Shubin Tan)

Table 1: Statistical results (AVG ± STD) obtained by DSP-SAEA and two variants on the benchmark problems with $10D$, $20D$ and $30D$.

Function No.	D	DSP-SAEA	DSP-SAEA-A	DSP-SAEA-CSO
F1	10	**8.42E-08(1.42E-07)**	5.06E-07(1.02E-06)+	2.38E-07(5.77E-07)+
	20	1.95E-01(1.67E-01)	1.75E-01(1.74E-01)=	**6.86E-02(1.04E-01)-**
	30	5.28E-01(2.31E-01)	4.00E-01(1.90E-01)-	**2.63E-01(2.09E-01)-**
F2	10	**1.22E+01(3.83E+00)**	1.69E+01(3.48E+00)+	1.60E+01(5.56E+00)+
	20	**2.28E+01(1.98E+00)**	2.59E+01(5.14E+00)+	2.67E+02(3.86E+01)+
	30	**3.32E+01(2.41E+00)**	3.77E+01(3.13E+00)+	3.67E+01(2.72E+00)+
F3	10	**4.29E+00(1.56E+00)**	4.48E+00(1.73E+00)=	4.57E+00(1.34E+00)=
	20	4.82E+00(1.37E+00)	**4.55E+00(1.03E+00)=**	4.63E+00(1.95E+00)=
	30	**3.67E+00(5.12E-01)**	3.91E+00(5.91E-01)=	4.06E+00(6.75E-01)=
F4	10	**2.78E-01(4.39E-01)**	8.07E-01(5.19E-01)+	6.30E-01(4.87E-01)+
	20	1.08E+00(3.74E-02)	1.08E+00(3.52E-02)=	**9.28E-01(1.17E-01)-**
	30	1.11E+00(3.76E-02)	1.10E+00(2.92E-02)=	**9.49E-01(7.29E-02)-**
F5	10	4.08E+01(1.74E+01)	**3.11E+01(1.64E+01)-**	3.48E+01(1.56E+01)-
	20	7.17E+01(2.77E+01)	8.06E+01(2.24E+01)+	**5.69E+01(2.29E+01)-**
	30	**1.20E+02(4.00E+01)**	1.21E+02(3.95E+01)=	1.48E+02(3.33E+01)+
F6	10	7.93E+07(5.02E+07)	**6.87E+07(3.71E+07)-**	7.11E+07(4.07E+07)-
	20	6.13E+08(1.83E+08)	5.53E+08(1.54E+08)-	**4.93E+08(2.19E+08)-**
	30	1.87E+08(5.10E+07)	**1.69E+08(4.90E+07)-**	1.86E+08(7.49E+07)=
F7	10	2.63E+07(2.40E+07)	**2.61E+07(2.18E+07)=**	3.33E+07(1.61E+07)+
	20	6.17E+07(2.85E+07)	**3.80E+07(1.35E+07)-**	5.80E+07(2.27E+07)-
	30	**6.72E+07(3.13E+07)**	8.87E+07(5.31E+07)+	7.50E+07(2.75E+07)+
F8	10	-2.69E+02(1.72E+01)	-2.76E+02(9.87E+00)-	**-2.80E+02(1.49E+01)-**
	20	-2.17E+02(2.92E+01)	**-2.39E+02(2.74E+01)-**	-2.19E+02(1.67E+01)=
	30	-1.51E+02(3.26E+01)	-1.41E+02(2.50E+01)+	**-1.53E+02(3.09E+01)=**
F9	10	**-2.71E+02(1.35E+01)**	-2.62E+02(1.62E+01)+	-2.63E+02(1.99E+01)+
	20	-1.50E+02(2.05E+01)	**-1.65E+01(1.12E+01)-**	-1.52E+02(1.85E+01)=
	30	**-1.33E+02(3.03E+01)**	-1.15E+02(3.75E+01)+	-1.11E+02(2.62E+01)+
F10	10	4.47E+02(6.19E+01)	4.43E+02(1.03E+02)=	**4.07E+02(7.90E+01)-**
	20	6.33E+02(1.11E+02)	6.40E+02(1.22E+02)=	**6.03E+02(8.06E+01)-**
	30	5.94E+02(6.71E+01)	**5.18E+02(5.20E+01)-**	5.22E+02(6.87E+01)-
F11	10	**1.20E+03(9.61E+01)**	1.21E+03(7.50E+01)=	**1.20E+03(6.66E+01)=**
	20	1.27E+03(3.99E+01)	**1.25E+03(7.41E+01)=**	1.26E+03(7.01E+01)=
	30	1.02E+03(2.89E+01)	**9.99E+02(2.37E+01)=**	1.03E+03(3.29E+01)=
F12	10	**1.20E+03(6.28E+01)**	1.23E+03(6.79E+01)=	**1.20E+03(3.33E+01)=**
	20	1.26E+03(2.94E+01)	**1.24E+03(8.42E+01)=**	1.27E+03(3.03E+01)=
	30	1.03E+03(5.69E+01)	**1.02E+03(3.66E+01)=**	1.04E+03(4.70E+01)=
F13	10	1.18E+03(8.66E+01)	**1.16E+03(7.19E+01)=**	1.17E+03(7.46E+01)=
	20	1.29E+03(6.27E+01)	**1.21E+03(6.01E+01)-**	1.27E+03(4.71E+01)=
	30	1.04E+03(4.74E+01)	**1.03E+03(4.47E+01)=**	1.04E+03(4.46E+01)=
F14	10	**1.55E+03(1.34E+02)**	1.56E+03(1.19E+02)=	1.57E+03(1.26E+02)=
	20	1.59E+03(1.06E+02)	**1.48E+03(1.08E+02)-**	1.51E+03(7.35E+01)-
	30	**1.77E+03(6.24E+01)**	**1.77E+03(7.32E+01)=**	**1.77E+03(4.17E+01)=**
F15	10	**1.49E+03(1.47E+02)**	**1.49E+03(1.20E+02)=**	1.51E+03(1.14E+02)=
	20	1.62E+03(8.49E+01)	1.63E+03(7.08E+01)=	**1.59E+03(9.76E+01)=**
	30	**1.53E+03(4.23E+01)**	1.55E+03(3.12E+01)=	**1.53E+03(3.99E+00)=**
+/-/=			10/12/23	10/14/21
DSP-SAEA wins			22.2%	22.2%
DSP-SAEA losses			26.7%	31.1%

Table 2: Statistical results (AVG ± STD) obtained by DSP-SAEA, GSGA, ESAO, CAL-SAPSO and GPEME on CEC2015.

Function No.	D	DSP-SAEA	GSGA	ESAO	CAL-SAPSO	GPEME
CEC2015$_{01}$	10	**9.96E+07(4.38E+07)**	1.40E+08(7.12E+07)+	1.96E+08(8.68E+07)+	4.35E+08(1.84E+08)+	1.47E+08(9.32E+07)+
	30	**3.31E+08(1.25E+08)**	4.96E+08(1.71E+08)+	1.43E+09(3.41E+08)+	5.17E+09(1.34E+09)+	1.22E+09(5.49E+08)+
CEC2015$_{02}$	10	**3.48E+08(2.08E+08)**	7.90E+08(5.46E+08)+	1.83E+09(1.07E+09)+	3.21E+09(1.42E+09)+	5.43E+09(2.05E+09)+
	30	2.28E+09(5.76E+08)	**1.70E+09(5.73E+08)-**	1.18E+10(2.59E+09)+	1.13E+10(1.86E+09)+	8.07E+10(2.11E+10)+
CEC2015$_{03}$	10	**3.21E+02(1.91E-01)**	**3.21E+02(1.17E-01)=**	**3.21E+02(1.46E-01)=**	**3.21E+02(1.25E-01)=**	**3.21E+02(2.12E-01)=**
	30	**3.21E+02(9.08E-02)**	**3.21E+02(1.01E-01)=**	**3.21E+02(1.03E-01)=**	**3.21E+02(5.31E-02)=**	**3.21E+02(6.95E-02)=**
CEC2015$_{04}$	10	**4.44E+02(1.36E+01)**	4.89E+02(1.62E+01)+	4.94E+02(1.72E+01)+	4.71E+02(3.99E+01)+	4.91E+02(2.26E+01)+
	30	6.06E+02(3.61E+01)	7.47E+02(5.22E+01)+	7.37E+02(5.05E+01)+	**5.34E+02(3.14E+01)-**	8.78E+02(5.42E+02)+
CEC2015$_{05}$	10	**2.31E+03(4.81E+02)**	2.81E+03(3.39E+02)+	3.02E+03(1.94E+02)+	2.78E+03(4.50E+02)+	2.91E+03(2.59E+02)+
	30	8.13E+03(6.09E+02)	9.05E+03(5.46E+02)+	9.33E+03(3.43E+02)+	**6.88E+03(1.67E+02)-**	9.18E+03(5.99E+02)+
CEC2015$_{06}$	10	8.18E+06(8.69E+08)	**6.45E+06(6.260E+06)-**	1.28E+07(1.06E+07)+	1.11E+08(9.94E+07)+	1.21E+07(3.05E+07)+
	30	**4.52E+07(3.45E+07)**	6.64E+07(3.04E+07)+	1.40E+08(9.00E+07)+	4.50E+08(2.63E+08)+	6.53E+07(5.72E+07)+
CEC2015$_{07}$	10	**7.20E+02(1.02E+01)**	7.22E+02(1.23E+01)=	7.25E+02(8.39E+00)=	7.34E+02(2.07E+01)+	7.24E+02(1.27E+01)=
	30	**8.63E+02(3.60E+01)**	8.91E+02(5.52E+01)+	1.02E+03(8.74E+01)+	9.57E+02(1.11E+02)+	1.13E+03(1.56E+02)+
CEC2015$_{08}$	10	**4.95E+05(6.98E+05)**	1.03E+06(7.31E+05)+	6.23E+06(8.90E+06)+	6.88E+06(5.62E+06)+	9.51E+05(8.95E+05)+
	30	**1.79E+07(1.75E+07)**	2.44E+07(1.62E+07)+	6.91E+07(3.73E+07)+	1.27E+08(8.35E+07)+	2.21E+07(1.32E+07)+
CEC2015$_{09}$	10	1.03E+03(9.73E+00)	1.03E+03(1.49E+01)=	**1.01E+03(4.28E+00)=**	1.04E+03(2.76E+01)=	1.05E+03(3.50E+01)=
	30	1.13E+03(2.75E+01)	1.13E+03(2.43E+01)=	**1.06E+03(1.39E+01)-**	1.25E+03(1.36E+02)+	1.31E+03(7.46E+01)+
CEC2015$_{10}$	10	**2.73E+05(4.05E+05)**	1.08E+06(9.28E+05)+	2.98E+06(2.85E+06)+	3.10E+06(3.72E+06)+	7.47E+05(1.05E+06)+
	30	9.39E+07(6.83E+07)	5.23E+07(3.39E+07)-	1.50E+08(8.93E+07)+	4.29E+08(2.27E+08)+	**3.56E+07(2.26E+07)-**
CEC2015$_{11}$	10	1.55E+03(1.05E+02)	1.57E+03(1.16E+02)=	1.69E+03(1.13E+02)+	1.82E+03(8.23E+01)+	**1.53E+03(1.52E+02)=**
	30	**2.49E+03(8.41E+01)**	2.69E+03(6.294+01)+	2.71E+03(5.63E+01)+	2.57E+03(1.13E+02)+	2.75E+03(1.664E+02)+
CEC2015$_{12}$	10	1.33E+03(6.58E+00)	1.33E+03(1.23E+01)=	**1.31E+03(4.11E+00)=**	1.34E+03(1.45E+01)=	1.33E+03(1.18E+01)=
	30	**1.35E+03(9.14E+00)**	1.37E+03(1.16E+01)=	**1.35E+03(1.83E+01)=**	1.39E+03(1.67E+01)=	1.45E+03(1.48E+01)+
CEC2015$_{13}$	10	**1.30E+03(1.88E+00)**	1.31E+03(5.65E+00)=	**1.30E+03(1.06E-02)=**	**1.30E+03(1.82E+00)=**	**1.30E+03(3.71E+00)=**
	30	1.37E+03(1.05E+02)	1.43E+03(2.12E+02)+	**1.30E+03(3.29E+00)-**	**1.30E+03(4.063E+00)-**	1.35E+03(8.27E+01)=
CEC2015$_{14}$	10	1.84E+04(3.65E+03)	1.82E+04(3.03E+03)=	1.60E+04(4.71E+03)-	2.12E+04(4.59E+03)+	**1.57E+04(1.99E+03)-**
	30	8.75E+04(1.33E+04)	7.88E+04(1.08E+04)-	**5.80E+04(6.21E+03)-**	7.52E+04(1.99E+04)-	8.79E+04(1.69E+04)=
CEC2015$_{15}$	10	**2.12E+03(3.93E+02)**	2.54E+03(7.35E+02)+	2.97E+03(1.39E+03)+	3.12E+03(3.25E+03)+	2.83E+03(5.59E+02)+
	30	**8.55E+03(4.11E+03)**	3.02E+04(2.17E+04)+	4.70E+04(4.76E+04)+	4.11E+05(3.51E+05)+	7.91E+04(8.77E+04)+
+/-/=			16/4/10	19/4/7	20/4/6	19/2/9
DSP-SAEA wins			53.3%	63.3%	66.7%	63.3%
DSP-SAEA losses			13.3%	13.3%	13.3%	6.67%

13.3% of the test problems. In addition, DSP-SAEA outperforms ESAO on 19 out of 30 problems. Moreover, DSP-SAEA wins CAL-SAPSO and GPEME in 66.7% and 63.3% of the test problems, respectively.

According to the statistical results given in Table 2, the overall performance of the proposed DSP-SAEA is better than that of GSGA, ESAO, CAL-SAPSO and GPEME on CEC2015 test problems.

References

Cai, X., Gao, L., & Li, X. (2020). Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Trans. Evol. Comput.*, *24*, 365–379.

Cheng, R., & Jin, Y. (2015). A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.*, *45*, 191–204.

Liang, J., Qu, B., Suganthan, P., & Chen, Q. (2014). Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. *Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, *29*, 625–640.

Liu, B., Zhang, Q., & Gielen, G. G. E. (2014). A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. Evol. Comput.*, *18*, 180–192.

Wang, H., Jin, Y., & Doherty, J. (2017). Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans. Cybern.*, *47*, 2664–2677.

Wang, X., Wang, G. G., Song, B., Wang, P., & Wang, Y. (2019). A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Trans. Evol. Comput.*, *23*, 815–827.