DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch130

Enhancing the learning capacity of immunological algorithms: a comprehensive study of learning operators

Shangce Gao[1], Tao Gong[1], Weiya Zhong[2] and Fang Wang[3] and Beibei Chen[1]

[1]College of Information Science and Technology, Donghua University, China
[2]Department of Mathematics, Shanghai University, China
[3]Department of Information Systems and Computing, Brunel University, Uxbridge UB8 3PH, UK
{gaosc,taogong}@dhu.edu.cn

Abstract

Immunological algorithms are a kind of bio-inspired intelligence methods which draw inspiration from natural immune systems. The problem-solving performance of immunological algorithms mainly lies on the utilization of learning (i.e. mutation) operators. In this paper, nine different learning operators in a standard immune algorithmic framework are investigated. These learning operators consist of eight existing operators and a newly proposed search direction based operator. Experiments are conducted based on nine variants of immunological algorithms that use different learning operators. Simulation results on a large number of benchmark optimization problems give a deep insight into the characteristics of these operators, and further verify that the proposed new learning operator can greatly improve the performance of immunological algorithms.

Introduction

Many difficulties such as dimensionality, differentiability and multimodality are associated with the optimization of large scale problems. To address such problems, bio-inspired intelligence algorithms (Da Silva Santos et al., 2010; Gao, 2012) have attracted more and more interest, among which the immunological algorithm (IA) is a particular class of optimization methods inspired by the basic features of adaptive immune response to antigenic stimulus. Most IAs mimic the metaphors of clonal selection principle (de Castro and Zuben, 2002), hypermutation (Freitas and Timmis, 2007), receptor editing (Gao et al., 2007) and lateral interaction effect (Whitbrook et al., 2007), providing a promising search mechanism by exploiting and exploring the solution space in parallel and effectively (Dasgupta et al., 2011). The main unique property of IAs is the utilization of the clonal proliferation, and the clonal selection which returns promising solutions acquired in the learning process. It is evident that IAs possess good features of maintaining population diversity, and capability of allocating multiple optimal solutions (Haktanirlar Ulutas and Kulturel-Konak, 2011). Although IAs have achieved good performance in solving various kinds of practical problems, such as digital signal processor (Mitra and Venayagamoorthy, 2010), nonlinear classification (Ozsen et al., 2009), fault diagnosis (Hao and Sun, 2007), etc, their performance is limited in solving optimization problems (McEwan and Hart, 2009). Compared with other bio-inspired algorithms, such as the well-known evolutionary computation (Yao and Xu, 2006), IAs still greatly suffer from the issues of stagnation and slow convergence. The reason seems to be that the learning capacity (involving hypermutation and receptor editing) has not been fully exploited, i.e., no sophisticated learning operator can be found in the literature (Jansen and Zarges, 2011). Based on the above consideration, we review and analyze the existing learning operators commonly used in IAs, and propose a new search direction based learning operator (L_{sd}) to encourage the antibodies to utilize the information of its surrounding antibodies, by means of moving the antibody toward the nearby antibodies with higher affinities and meanwhile away from the antibody with lower affinities. Therefore, the L_{sd} operator can not only evolve antibodies into promising search areas to accelerate convergence speed, but also prevent antibodies from entering undesired regions to jump out of local optimal solutions. The experiments of using all learning operators in IAs are conducted based on a large number of benchmark numerical optimization problems. The results show the characteristics of each learning operator, and further indicate that the proposed L_{sd} operator manipulates the best performance.

Immunological algorithm

To investigate the effect of learning operators, a standard immunological algorithm framework (called IA) is utilized (de Castro and Zuben, 2002; Kelsey and Timmis, 2003). IA evolves a population of antibodies (B cells) towards a global optimum through a process of evaluation, cloning, learning (i.e. mutation) and selection. The evaluation procedure computes the affinity function values for all antibodies. Affinity is an important measure to represent the fitness of antibody to antigen. For a minimization optimization problem, higher affinity values of antibodies correspond to better solutions for the problem needed to be solved. The cloning proliferation is a mitotic procedure whereby the cells divide

themselves, creating a set of clones identical to the parent cell. Generally, the proliferation rate is directly proportional to the affinity level. The learning procedure (involving hypermutation or receptor editing) performs the exploration and exploitation within solution search space, guiding antibodies to the global optimum. It plays a key effect on the search performance of the algorithm. The selection procedure picks up the antibodies with higher affinities and meanwhile eliminates those with lower affinities to reduce the computational complexity.

From the perspective of optimization, a learning operator L mutate a candidate solution $Ab = (x_1, x_2, ..., x_D)$ to a trial one $Ab' = L(Ab) = (x'_1, x'_2, ..., x'_D)$, where D is the dimension of the problem. In the literature, there are several learning operators commonly used in IA. They are summarized in the following.

- Gaussian mutation L_{gs}(Yao et al., 1999; de Castro and Timmis, 2002; Xu and Zhang, 2007; Khilwani et al., 2008; Song et al., 2006; Woldemariam and Yen, 2010): $Ab' = Ab + \alpha N(0, 1)$, where $N(0, 1)$ is a random Gaussian number generated by the Gaussian function given as $f_{Gaussian}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, and α controls the learning intensity imposed on the antibodies.

- Cauchy mutation L_{cy} (Yao et al., 1999; Xu and Zhang, 2007; Khilwani et al., 2008): $Ab' = Ab + \alpha\delta_k$, where δ_k is a Cauchy random variable with the scale parameter $t = 1$ and satisfies the density function $f_{Cauchy}(x) = \frac{t}{\pi(t^2+x^2)}$.

- Static Hypermutation L_{h1} (Cutello et al., 2004; Gong et al., 2008): the number of mutations is independent from the affinity of the antibody. That is to say, Ab' will undergo a constant number c of mutation times. Each mutation act on Ab is implemented through replacing a certain number of Ab at a random dimension with a random integer between 0 and 9 (Gong et al., 2008).

- Proportional Hypermutation L_{h2} (Cutello et al., 2004): the number of mutations is proportional to the normalized affinity value, that is, $\hat{f}(Ab) \times c \times D$, where $\hat{f}(Ab)$ is the normalized affinity distributed in the interval of $[0, 1]$. c is a constant number, representing the maximum mutation intensity.

- Inversely proportional hypermutation L_{h3} (Cutello et al., 2004, 2005, 2006): the number of mutations is inversely proportional to the normalized affinity value, i.e., the higher affinity of an antibody, the less times of mutations will be carried out on it. It is reasonable to make such an inverse choice, since better antibodies usually contain more useful information for evolution. Too many mutations might have higher probability to destroy these information, thus depressing the learning performance.

- Hypermacromutation L_m (Cutello et al., 2004): the number of mutations is independent from the affinity and the parameter c. Instead, the operator mutates at most $j-i+1$ values in the interval of $[i, j]$, where two randomly generated integers i and j satisfy the condition of $i < j \leq D$.

- Lateral interaction mutation L_{li} (Cutello et al., 2006; Pavone et al., 2011): in addition to hypermutation and receptor editing, the lateral interaction during different antibodies also takes place according to the idiotypic network theory (Gao et al., 2008). In other words, each paratope on an antibody can not only recognize a foreign antigen, but also can be recognized by external idiotopes. Motivated by this mechanism, similar to the crossover operator in evolutionary computation, an antibody is attracted by other antibodies, i.e., $Ab'_i = (1 - \beta) \times Ab_i + \beta Ab_j$, where $Ab_j \neq Ab_i$ is a randomly selected antibody in the population.

- Baldwinian learning L_{bl} (Gong et al., 2010): learning mechanism can provide an easy evolutionary path towards co-adapted alleles in environments, by means of employing differential information during other antibodies. It is realized as $Ab'_i = Ab_i + s \times (Ab_j - Ab_k)$ in a probability of p, where $i \neq j \neq k$, Ab_j and Ab_k are randomly selected from the population, and s represents the Baldwinian learning strength.

Intuitively, all the above eight learning operators are able to evolve antibodies to matured ones in semi-blind manners, although some of the matured ones might possess lower affinities. However, as the parallel feature of the immune algorithm, there did exist a probability of making progress to improve the affinity of antibodies. After the clonal selection progress, the most improved antibodies are reserved and enter into the next generation of evolution.

Search direction based learning operator

Even though the above learning operators used in IA can explore/exploit the solution space in an effective manner, as we observed, they are not fully developed from the aspect of utilizing the information in environment. In Fig. 1, we summarized the characteristics of the learning operators. The solid rectangle S show the solution space of the optimization problem. The dashed circles denote contour lines of affinity, and the inner circles indicate that they represent higher affinities than the outer ones.

From Fig. 1, we can notice that the learning mechanisms used in (1)-(6) on the antibody Ab only utilize random perturbation on the antibody itself, while those in (7)-(8) make use of information in the environment. As reported in (Cutello et al., 2006; Gong et al., 2010), learning from the environment provides an encouraging alternative method, probably a more easy way to achieve better search performance. In details, the mechanism in (7) uses the information of a randomly selected antibodies in the population to

guide the current search. A successful guide is strongly depending on the quality of the selected guiding antibody Ab_j, implying that there must be amount of redundant search if the guiding antibody is far away from the global optimal solution. Instead, the mechanism in (8) utilizes the differential information between two other antibodies Ab_j and Ab_k in the population. The learning acting on this differential information might have ability to use the mutual beneficial components, thus exhibiting more promising properties for searching.

Based on the above analysis, we can find that the mechanisms used in (7) and (8) don't use any measurable knowledge from the population, i,e, the guiding antibodies (Ab_j and Ab_k) are randomly selected without any relevance of the current antibody Ab, therefore hindering the effectiveness of the learning performance. In view of the limitations of the above learning operators, we propose a new search direction based learning operator L_{sd}, not only to provide another alternative mutation method, but also aiming to achieve a better search capacity. The L_{sd} is formulated in Eqs. (1)-(3):

In these equations, $\Delta_{repulsion}$ and $\Delta_{attraction}$ are repulsion and attraction effect on Ab_i respectively. In a single learning procedure of L_{sd}, if a randomly selected guiding antibody Ab_j whose affinity is lower that the base antibody Ab_i, then the repulsion effect will be implemented with a probability of p_r, for the purpose of preventing the base antibody from entering undesired region of the search space. On the other hand, the attraction effect takes place with a probability of p_a when the affinity of the selected guiding antibody Ab_k is higher, thus enhancing the capacity of exploiting the promising regions of the search space. In addition, to improve the randomness of the learning mechanism, the attraction and repulsion scaling factors α_1 and α_2 are set as random numbers generated in the interval $[0, 1]$. From Eq. (3), it is clear that the search direction of the base antibody is guided to move towards the regions with higher affinity, and meanwhile away from the regions with lower affinity, thus enabling the algorithm to possess better exploitation capacity and the ability of jumping out of local optimum.

Experimental results and discussions

The computational progresses of nine learning operators used in IA described above have been implemented in C++ program under Visual Studio 2010. In order to evaluate the performance of the proposed L_{sd} learning operator, it is validated using some well-known benchmark numerical optimization problems obtained from the literatures (Yao et al., 1999; Cutello et al., 2006; Gong et al., 2010). Table 1 lists the details of the benchmark functions. $f_1 - f_5$ are unimodal functions which are relatively easy to be optimized, but the difficulty increases as the dimension size increases. f_6 is the step function, while f_7 is a noisy quartic function. $f_8 - f_{13}$ are multimodal functions with plenty of local minima which represent the most difficult class of problems for many optimization algorithms; $f_{14} - f_{23}$ is a multimodal function with only a few local optima. The different type of benchmark functions test the searching ability of learning operators from different aspects, that is, unimodal functions trend to reflect the convergence speed of the opearator in a direct manner, while multimodal functions are likely to estimate the operator's capacity of escaping from local optima.

Owing to the random nature of the IA and the learning operators, to evaluate the performance of each learning operator, their performance cannot be judged by the result of a single run. Many trials with independent population initialization should be made to obtain a useful conclusion of the performance of the approach. Therefore, in this study the results are obtained in 30 trials. In the experiment, the user-defined parameters are set as follows: the population size is set to be 30, the clone size is 5. It is worth mentioning that we use equal cloning strategy in this study to reduce the influences of cloning operator. By doing so, each antibody in the population has the same probability to undergo the learning mechanism, thus we can make a direct comparison of the performance during all learning operators. In addition, the termination condition of the algorithm is set to be that when the maximum number of function evaluations reaches 150000. Fig. 2 depicts the sketch of the Sphere function f_1 when its dimension D is set to be 2, and the corresponding convergence graphs of each learning operator. It is obvious that, for such unimodal function, all learning operators can evolve the antibodies effectively. In particular, the learning operator L_{sd} possesses the fast learning speed and the most precise solution.

To further demonstrate the effectiveness and robustness of the proposed L_{sd}, all learning operators are carried out on all tested 23 benchmark functions. One of the effective strategies to perform a comparative study between the variants of IAs is to use the oracle-based view of computation (Wolpert and Macready, 1997). Based on this method, the best solution should be found within a certain number of function evaluations. Herein, the best values can be used for comparison because of the equal number of function evaluation for all operators in all cases. To make an intuitive comparison during the variants, the results of best run are normalized between 0 and 1, therefore the worst and best values of each best solution are changed to 0 and 1, respectively. The normalized results for all benchmark functions are presented in Table 2. To achieve a general conclusion based on the oracle-based view, three kinds of the sum of scores and rank of each learning operator are presented in this table. The symbol $\sum_u$ denotes the sum of scores on unimodal function $f1 - f5$, while $\sum_m$ is the sum on multimodal functions $f_8 - f_{13}$. $\sum$ represents the total sum on all tested benchmark functions. At first glance, it is clear that the learning operator L_{sd} works very well because it has the best performance with the score of 22.940 to 23, and has a rank of 1 among 9 operators. Furthermore, as the $\sum_u$ of

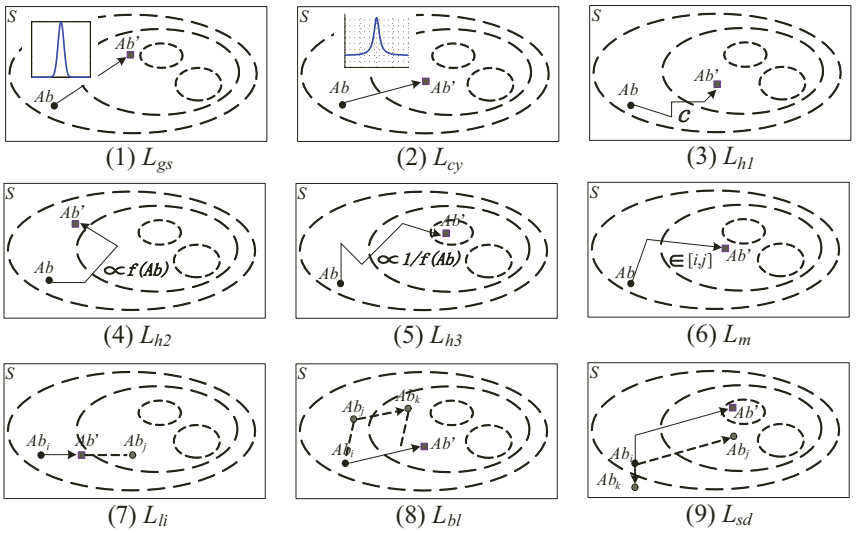


Figure 1: The learning characteristics of all nine operators.

L_{sd} is 4.941 which is also the best one among all operators, thus confirming that L_{sd} has a fastest convergence speed for unimodal functions. On the other hand, the biggest value of $\sum_m$ of L_{sd} also verifies the capability of escaping from local optimum for multimodal functions.

Based on Table 2, there are some remarks should be emphasized concerning all learning operators. The first six learning operators mutate the base antibody only utilizing random perturbation, while the last three ones make use of information in the population either semi-blind or search direction based. The population information utilization based learning operators have ranks of 1, 2, 3, significant better than the others, suggesting that the interaction of information in the population is likely to improve the search performance. Thus, we can conclude that the former operators mainly act as the exploitation in the search space, while the latter ones mainly employed as exploration. In the further, it is a promising research direction to combine one of former six operators with the one of latter ones, and it can be expected to achieve a better performance.

Conclusions

In this paper, we made a comprehensive study on the learning operators used in the immunological algorithms. Nine different learning operators, maturing the antibody either by utilizing random perturbation or by utilizing the guiding information from other antibodies, are implemented and analyzed. In view of the limitations of the existing operators, the newly proposed search direction based learning mechanism can not only attract the antibody to promising regions in search space, but also preventing from entering undesired regions by means of the information contained in the antibodies with worse affinities. Experimental results on a large number of benchmark numerical optimization problems verified the effectiveness and robustness of L_{sd}, suggesting that the useful information during the whole population should be sufficiently utilized to improve the search performance of the algorithm.

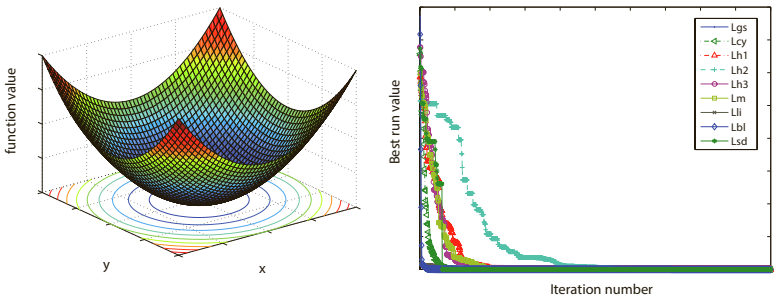
Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grants 61203325, 61271114 and 61003205, Genguang Project of Shanghai E-

$$\Delta_{repulsion} = \begin{cases} Ab_j - Ab_i, & \text{if } rand() < p_r \text{ for } \forall\, f(Ab_j) < f(Ab_i) \\ Ab_i, & otherwise \end{cases} \quad (1)$$

$$\Delta_{attraction} = \begin{cases} Ab_k - Ab_i, & \text{if } rand() < p_a \text{ for } \forall\, f(Ab_k) \geq f(Ab_i) \\ Ab_i, & otherwise \end{cases} \quad (2)$$

$$Ab' = Ab + \alpha_1 \times \Delta_{attraction} - \alpha_2 \times \Delta_{repulsion} \quad (3)$$



(a) 2D sketch (b) convergence graph

Figure 2: (a) the 2D sketch of the Sphere function f_1, (b) the convergence graphs of each learning operators on f_1.

ducational Development Foundation (No. 12CG35), Ph.D. Program Foundation of Ministry of Education of China (No. 20120075120004), and the Fundamental Research Funds for the Central Universities.

References

Cutello, V., Narzisi, G., Nicosia, G., and Pavone, M. (2005). An immunological algorithm for global numerical optimization. In *Proceedings of the 7th international conference on Artificial Evolution*, pages 284–295. Springer-Verlag.

Cutello, V., Nicosia, G., and Pavone, M. (2004). Exploring the capability of immune algorithms: A characterization of hypermutation operators. In *Third International Conference on Artificial Immune Systems, ICARIS 2004*, pages 263–276.

Cutello, V., Nicosia, G., and Pavone, M. (2006). Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 950–954. ACM.

Da Silva Santos, C. H., Goncalves, M. S., and Hernandez-Figueroa, H. E. (2010). Designing novel photonic devices by bio-inspired computing. *IEEE Photonics Technology Letters*, 22(15):1177–1179.

Dasgupta, D., Yu, S., and Nino, F. (2011). Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing*, 11:1574–1587.

de Castro, L. and Zuben, F. J. V. (2002). Learning and optimization using clonal selection principle. *IEEE Trans. on Evolutionary Computation*, 6(3):239–251.

de Castro, L. N. and Timmis, J. (2002). An artificial immune network for multimodal function optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, volume 1, pages 699–704. IEEE.

Freitas, A. A. and Timmis, J. (2007). Revisiting the foundations of artificial immune systems for data mining. *IEEE Trans. on Evolutionary Computation*, 11(4):521–540.

Gao, S. C., editor (2012). *Bio-Inspired Computational Intelligence and Their Applications*. INTECH press, New York, USA.

Gao, S. C., Dai, H., Zhang, J., and Tang, Z. (2008). An expanded lateral interactive clonal selection algorithm and its application. *IEICE Trans. Fundamentals*, E91-A(8):2223–2231.

Gao, S. C., Tang, Z., Dai, H., and Zhang, J. (2007). An improved clonal algorithm and its application to traveling salesman problems. *IEICE Trans. Fundamentals*, E90-A(12):2930–2938.

Gong, M., Jiao, L., and Zhang, L. (2010). Baldwinian learning in clonal selection algorithm for optimization. *Information Sciences*, 180(8):1218–1236.

Gong, M., Jiao, L., and Zhang, X. (2008). A population-based artificial immune system for numerical optimization. *Neurocomputing*, 72(1-3):149–161.

Haktanirlar Ulutas, B. and Kulturel-Konak, S. (2011). A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2):117–138.

Hao, X. and Sun, C. (2007). Artificial immune network classification algorithm for fault diagnosis of power transformer. *IEEE Trans. on Power Delivery*, 22(2):930–935.

Jansen, T. and Zarges, C. (2011). Analyzing different variants of immune inspired somatic contiguous hypermutations. *Theoretical Computer Science*, 412(6):517–533.

Kelsey, J. and Timmis, J. (2003). Immune inspired somatic contiguous hypermutation for function optimisation. In *Genetic and Evolutionary Computation, GECCO 2003*, pages 207–212. Springer.

Khilwani, N., Prakash, A., Shankar, R., and Tiwari, M. K. (2008). Fast clonal algorithm. *Engineering Applications of Artificial Intelligence*, 21:106–128.

McEwan, C. and Hart, E. (2009). On airs and clonal selection for machine learning. In *Artificial Immune Systems*, Artificial Immune Systems, pages 67–79. Springer.

Mitra, P. and Venayagamoorthy, G. K. (2010). An adaptive control strategy for dstatcom applications in an electric ship power system. *IEEE Trans. on Power Electronics*, 25(1):95–104.

Ozsen, S., Gunes, S., Kara, S., and Latifoglu, F. (2009). Use of kernel functions in artificial immune systems for the nonlinear classification problems. *IEEE Trans. on Information Technology in Biomedicine*, 13(4):621–628.

Pavone, M., Narzisi, G., and Nicosia, G. (2011). Clonal selection: an immunological algorithm for global optimization over continuous spaces. *Journal of Global Optimization*, pages 1–40.

Song, Z. Y., Gao, X. Z., Hiang, X. L., and Lin, H. S. (2006). A modified immune optimization algorithm. In *International Conference on Machine Learning and Cybernetics*, pages 2184–2189. IEEE.

Whitbrook, A. M., Aickelin, U., and Garibaldi, J. (2007). Idiotypic immune networks in mobile-robot control. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(6):1581–1598.

Woldemariam, K. M. and Yen, G. G. (2010). Vaccine-enhanced artificial immune system for multimodal function optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(1):218–228.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, 1:67–82.

Xu, X. and Zhang, J. (2007). An improved immune evolutionary algorithm for multimodal function optimization. In *Third International Conference on Natural Computation, ICNC 2007*, volume 3, pages 641–646. IEEE.

Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *IEEE Trans. on Evolutionary Computation*, 3(2):82–102.

Yao, X. and Xu, Y. (2006). Recent advances in evolutionary computation. *Journal of Computer Science and Technology*, 21(1):1–18.

Table 1: Benchmark problems used in the experiments.

		Function Definition	Dim.	Domain S	f_{min}
$f_1(X)$	=	$\sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(X)$	=	$\sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	$[-10, 10]^n$	0
$f_3(X)$	=	$\sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]^n$	0
$f_4(X)$	=	$max\{\|x_i\|, 1 \leqslant i \leqslant n\}$	30	$[-100, 100]^n$	0
$f_5(X)$	=	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	0
$f_6(X)$	=	$\sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^n$	0
$f_7(X)$	=	$\sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^n$	0
$f_8(X)$	=	$\sum_{i=1}^{n} -x_i \text{sin}(\sqrt{\|x_i\|})$	30	$[-500, 500]^n$	-12569.5
$f_9(X)$	=	$\sum_{i=1}^{n} [x_i^2 - 10\text{cos}(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^n$	0
$f_{10}(X)$	=	$-20\text{exp}(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})$ $-\text{exp}(\frac{1}{n}\sum_{i=1}^{n} \text{cos}(2\pi x_i)) + 20 + e$	30	$[-32, 32]^n$	0
$f_{11}(X)$	=	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \text{cos}(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^n$	0
$f_{12}(X)$	=	$\frac{\pi}{n}\{10\text{sin}^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\text{sin}^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \geq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^n$	0
$f_{13}(X)$	=	$0.1\{\text{sin}^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \text{sin}^2(3\pi x_{i+1})]$ $+(x_n - 1)[1 + \text{sin}^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0
$f_{14}(X)$	=	$[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}]^{-1}$	2	$[-65.536, 65.536]^n$	1
$f_{15}(X)$	=	$\sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	$[-5, 5]^n$	0.0003075
$f_{16}(X)$	=	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.0316285
$f_{17}(X)$	=	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}\text{cos}x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
$f_{18}(X)$	=	$[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$ $[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	2	$[-2, 2]^n$	3
$f_{19}(X)$	=	$-\sum_{i=1}^{4} c_i \text{exp}[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2]$	3	$[0, 1]^n$	-3.86
$f_{20}(X)$	=	$-\sum_{i=1}^{4} c_i \text{exp}[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$	6	$[0, 1]^n$	-3.32
$f_{21}(X)$	=	$-\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.1422
$f_{22}(X)$	=	$-\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.3909
$f_{23}(X)$	=	$-\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.53

Table 2: Normalized statistical results of learning operators, L_{gs}, L_{cy}, L_{h1}, L_{h2}, L_{h3}, L_m, L_{li}, L_{bl}, L_{sd} for the benchmark problems.

Function	L_{gs}	L_{cy}	L_{h1}	L_{h2}	L_{h3}	L_m	L_{li}	L_{bl}	L_{sd}
f_1	0.856	0.901	0.285	0.000	0.800	0.915	1.000	1.000	1.000
f_2	0.000	0.785	0.567	0.234	0.965	0.865	0.999	1.000	1.000
f_3	0.000	0.235	0.245	0.345	0.657	0.768	0.987	1.000	1.000
f_4	0.000	0.324	0.156	0.000	0.483	0.481	0.925	1.000	0.956
f_5	0.000	0.124	0.000	0.235	0.210	0.454	0.768	0.923	0.985
f_6	0.125	0.248	0.000	0.000	0.358	0.405	0.567	0.999	1.000
f_7	0.450	0.500	0.000	0.000	0.146	0.056	0.679	0.956	1.000
f_8	0.235	0.167	0.250	0.000	0.580	0.375	0.788	0.876	1.000
f_9	0.000	0.056	0.120	0.000	0.450	0.734	0.567	0.752	1.000
f_{10}	0.125	0.450	0.045	0.000	0.236	0.678	0.458	1.000	0.999
f_{11}	0.250	0.467	0.000	0.011	0.235	0.385	0.572	0.750	1.000
f_{12}	0.000	0.450	0.245	0.000	0.560	0.476	0.877	0.999	1.000
f_{13}	0.258	0.782	0.000	0.000	0.450	0.359	0.578	0.974	1.000
f_{14}	0.856	0.978	0.450	0.000	0.874	0.385	0.683	0.999	1.000
f_{15}	0.784	0.654	0.000	0.000	0.487	0.530	0.976	0.965	1.000
f_{16}	0.460	0.750	0.000	0.674	0.576	0.045	0.956	0.999	1.000
f_{17}	0.470	0.865	0.000	0.000	0.430	0.012	0.995	0.999	1.000
f_{18}	0.956	1.000	0.385	0.000	0.450	0.755	0.999	0.999	1.000
f_{19}	0.845	0.999	0.568	0.000	0.450	0.785	1.000	1.000	1.000
f_{20}	0.864	0.968	0.452	0.000	0.969	0.704	0.933	0.999	1.000
f_{21}	0.765	0.742	0.000	0.011	0.345	0.358	1.000	1.000	1.000
f_{22}	0.875	0.785	0.075	0.000	0.550	0.340	1.000	1.000	1.000
f_{23}	0.920	0.965	0.105	0.000	0.568	0.285	1.000	1.000	1.000
$\sum_u$	0.856	2.369	1.253	0.814	3.115	3.483	4.679	4.923	4.941
$\sum_m$	0.868	2.372	0.660	0.011	2.511	3.007	3.840	5.351	5.999
$\sum$	10.094	14.195	3.948	1.510	11.829	11.150	19.307	22.189	22.940
$Rank$	7	4	8	9	5	6	3	2	1