A Simple Human Learning Optimization Algorithm

Ling Wang[1], Haoqi Ni[1], Ruixin Yang[1], Minrui Fei[1], and Wei Ye[1,2]

[1] Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, 200072, China
[2] Institute of Computer Science, University of Munich, Munich, 80538, Germany
{wangling,nhq12721182,yrx816,mrfei,yeweiysh}@shu.edu.cn

Abstract. This paper presents a novel Simple Human Learning Optimization (SHLO) algorithm, which is inspired by human learning mechanisms. Three learning operators are developed to generate new solutions and search for the optima by mimicking the learning behaviors of human. The 0-1 knapsack problems are adopted as benchmark problems to validate the performance of SHLO, and the results are compared with those of binary particle swarm optimization (BPSO), modified binary differential evolution (MBDE), binary fruit fly optimization algorithm (bFOA) and adaptive binary harmony search algorithm (ABHS). The experimental results demonstrate that SHLO significantly outperforms BPSO, MBDE, bFOA and ABHS. Considering the ease of implementation and the excellence of global search ability, SHLO is a promising optimization tool.

Keywords: human learning optimization, meta-heuristic, global optimization, learning operators, optimization algorithm.

1 Introduction

The computational drawbacks of existing derivative-based numerical methods such as complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required have forced researchers to rely on meta-heuristic algorithms to solve complicated optimization problems, such as Genetic Algorithms [1], Ant Colony Optimization [2], Particle Swarm Optimization [3], Harmony Search [4], and Fruit Fly Optimization Algorithms [5]. To effectively and efficiently solve hard optimization problems, new powerful meta-heuristics inspired by nature, especially by biological systems, must be explored, which is a hot topic in evolutionary computation community now [6].

Many human learning activities are similar to the search process of meta-heuristics. For instance, when a person learns how to play Sudoku, he or she repeatedly studies and practices to master and improve new skills and evaluate his or her performance for guiding the following study while meta-heuristics iteratively generate new solutions and calculate the corresponding fitness values for adjusting the following search. In most activities human can solve problems by random learning, individual learning, and social learning. For the example of learning Sudoku again, a person may

M. Fei et al. (Eds.): LSMS/ICSEE 2014, Part II, CCIS 462, pp. 56–65, 2014.

randomly learn due to lack of prior knowledge or exploring new strategies (random learning), learn from his or her previous experience (individual learning) and learn from his or her friends and related books (social learning). Inspired by this simple learning model, a simple human learning optimization algorithm is proposed.

The rest of the paper is organized as follows. Section 2 introduces the idea, operators and implementation of SHLO in detail. Then, the presented SHLO is applied to tackle a set of 0-1 knapsack problems to evaluate its performance in Section 3. Finally, Section 4 concludes the paper.

2 Simple Human Learning Optimization Algorithm

2.1 Initialization

The binary-coding framework is adopted in SHLO, and consequently an individual in SHLO is represented by a binary string as Eq. (1),

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{iM} \end{bmatrix}, x_{ij} \in \{0,1\}, 1 \le i \le N, 1 \le j \le M \quad (1)$$

where x_i denotes the *i-th* individual, N is the size of the population, and M is the dimension of solutions. Each bit of a binary string is initialized as “0” or “1” randomly, which stands for a basic element of the knowledge or skill that people want to learn and master.

2.2 Learning Operators

2.2.1 Random Learning Operator

At the beginning of learning, people usually learn at random as there is no prior knowledge of problems. In the following studying, due to forgetting, only knowing partial knowledge of problems and other factors, individuals cannot fully replicate previous experience and therefore they still learn with a certain randomness. To emulate these phenomena of randomness in human learning, a simplified random learning operator is developed for SHLO as Eq. (2).

$$x_{ij} = Rand(0,1) = \begin{cases} 0, & 0 \le rand() \le 0.5 \\ 1, & else \end{cases} \quad (2)$$

where *rand*() is a stochastic number between 0 and 1.

2.2.2 Individual Learning Operator

Individual learning is defined as the ability to build knowledge through individual reflection about external stimuli and sources [7]. It is very common that people use their own experience and knowledge to avoid mistakes and improve their performance during the process of study. To mimic individual learning of human in SHLO, an individual knowledge database (IKD) is used to store personal best experience as Eq. (3-4)

$$IKD = \begin{bmatrix} ikd_1 \\ ikd_2 \\ \vdots \\ ikd_i \\ \vdots \\ ikd_N \end{bmatrix}, 1 \le i \le N \tag{3}$$

$$ikd_i = \begin{bmatrix} ikd_{i1} \\ ikd_{i2} \\ \vdots \\ ikd_{ip} \\ \vdots \\ ikd_{iL} \end{bmatrix} = \begin{bmatrix} ik_{i11} & ik_{i12} & \cdots & ik_{i1j} & \cdots & ik_{i1M} \\ ik_{i21} & ik_{i22} & \cdots & ik_{i2j} & \cdots & ik_{i2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{ip1} & ik_{ip2} & \cdots & ik_{ipj} & \cdots & ik_{ipM} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{iL1} & ik_{iL2} & \cdots & ik_{iLj} & \cdots & ik_{iLM} \end{bmatrix}, 1 \le p \le L \tag{4}$$

where ikd_i denotes the IKD of person i, L is the pre-defined number of solutions saved in the IKD, and ikd_{ip} stands for the p-th best experience of person i.

When SHLO conducts individual learning, it generates new solutions based on the knowledge in the IKD, which is operated as Eq.(5)

$$x_{ij} = ikd_{ipj} \tag{5}$$

2.2.3 Social Learning Operator

Although a person could learn and solve problems on his or her own experience, i.e. through individual learning, the learning process may be very slow and inefficient if problems are complicated. In the social environment, people can learn from a collective experience through social learning to further develop their ability [8, 9]. In this context, people directly or indirectly transfer knowledge and skills, and hence the efficiency and effectiveness of learning will be improved from experience share [10, 11]. For emulating this efficient learning strategy, the social knowledge data (SKD) is used to reserve the knowledge of the population as Eq. (6)

$$SKD = \begin{bmatrix} skd_1 \\ skd_2 \\ \vdots \\ skd_q \\ \vdots \\ skd_H \end{bmatrix} = \begin{bmatrix} sk_{11} & sk_{12} & \cdots & sk_{1j} & \cdots & sk_{1M} \\ sk_{21} & sk_{22} & \cdots & sk_{2j} & \cdots & sk_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{q1} & sk_{q2} & \cdots & sk_{qj} & \cdots & sk_{qM} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{H1} & sk_{H2} & \cdots & sk_{Hj} & \cdots & sk_{HM} \end{bmatrix}, 1 \le q \le H \tag{6}$$

where H is the size of the SKD and skd_q is the *q-th* solution in SKD.

Based on the knowledge in the SKD, SHLO can perform social learning as Eq. (7) to generate better solutions in the search process.

$$x_{ij} = sk_{qj} \tag{7}$$

In summary, SHLO uses the random learning operator, individual learning operator and social learning operator to yield new solutions and search for the optima based on the knowledge stored in the IKD and SKD just like human learning and improving skills by these three learning forms, which can be integrated and operated as Eq. (8)

$$x_{ij} = \begin{cases} Rand(0,1), & 0 \le rand() \le pr \\ ik_{ipj}, & pr < rand() \le pi \\ sk_{qj}, & else \end{cases} \tag{8}$$

where pr is the probability of random learning, and the values of (pi-pr) and (1-pi) represents the probabilities of performing individual learning and social learning, respectively.

2.3 Updating Operation

After all individuals generate new candidate solutions, the fitness of each individual is evaluated according to the pre-defined fitness function which is used to update IKD and SKD for the following search, just like people evaluate their performance of new practices to summarize and update their experience for leaning better in the following steps. For the updating of the IKD, the new generated solution will be stored in the IKD if its fitness value is better than the worst one in the IKD or the current number of solutions in the IKD is less than the pre-defined value. For the updating of SKD, the best solution of the current generation will be saved in the SKD if its fitness value is superior to that of the worst one in the SKD or the current number of solutions in the SKD is less than the pre-defined number. Note that the SKD updates no more than one solution at each iterative step, which can keep a better diversity of the algorithm to avoid the premature.

SHLO runs the learning operators and updates the IKD and SKD repeatedly till it finds the optima of problems or the termination criterions are met. The procedure of SHLO can be illustrated as Fig. 1.

3 Experimental Results and Discussions

To evaluate the performance of the algorithm, SHLO, as well as other four binary-coding optimization algorithms, i.e. binary PSO (BPSO) [12], modified binary differential evolution (MBDE) [13], binary fruit fly optimization algorithm (bFOA) [14] and adaptive binary harmony search algorithm (ABHS) [15], was applied to solve 0-1 knapsack problems (0-1 KPs). For a fair comparison, the recommended parameters of BPSO, MBDE, bFOA and ABHS were used to tackle these problems. As there is no adaptive strategy in the original bFOA and MBDE which significantly spoils their performance on high-dimensional problems, the adaptive strategy is introduced into these two algorithms and the parameters are set based on a parameter study. The parameters of all the algorithms are listed in Table 1. As the benchmark problems are the "single-objective" problems, the sizes of the IKD and SKD are both set to 1 based on trails and error to enhance search efficiency and reduce the cost of computation.

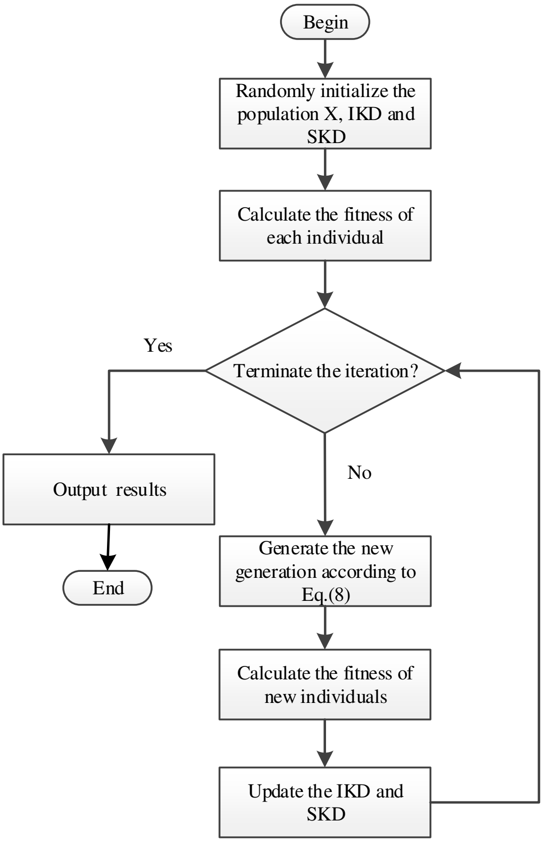

Fig. 1. The flowchart of SHLO

3.1 0-1 Knapsack Problems

Knapsack problems have been studied intensively in the last few decades, attracting both theorists and practitioners. From a practical point of view, knapsack problems can model many application problems such as capital budgeting, cargo loading and cutting stock [16].

Table 1. Parameters settings of SHLO, BPSO, MBDE, bFOA and ABHS

Algorithm	Parameters
SHLO	$pr=\frac{5}{M},\ pi=0.85+\frac{2}{M}$
BPSO [12]	c_1=1.5, c_2=1.5,$w_{\min}$=0.1, $w_{\max}$=0.9, $V_{\max}$=4, $V_{\min}$=-4
MBDE [13]	F_1=0.5, F_2=0.005, $CR_{\max}$=0.8, $CR_{\min}$=0.2
bFOA [14]	S = 3, L = 3, $b_{\max}$ = 30, $b_{\min}$=6
ABHS [15]	C = 15; PAR = 0.2; HMS = 30; NGC = 20;

* M is the dimensionality of the solution.

Given a set of n items and each item j having an integer profit p_j as well as an integer weight w_j, the 0-1 knapsack problem (0-1 KP) is defined to choose a subset of items such that their overall profit is maximized while the overall weight does not exceed a given capacity, which can be formulated as Eq. (9)

$$\begin{aligned} &Max\, f(x)=\sum_{j=1}^{N} p_j x_j \\ &s.t.\begin{cases} \sum_{j=1}^{N} w_j x_j \le C \\ x_j = 0\ or\ 1,\ j=1,2,...,N \end{cases} \end{aligned} \tag{9}$$

where the binary decision variable x_j is used to indicate whether item j is included in the knapsack or not. Without loss of generality, 0-1 KPs assume that all profits and weights are positive and all weights are smaller than the capacity C.

Note that 0-1 KPs are constrained problems, and thus the penalty function as Eq. (10) is adopted to deal with infeasible solutions of which the total weight exceeds the limit C. No heuristic strategy of KPs is introduced in this paper to avoid the influence on the real performance of the algorithm,

$$\begin{aligned} &MaxF(x)=f(x)-\lambda\times\max(0,c) \\ &c=\sum_{j=1}^{N} w_j x_j - C \end{aligned} \tag{10}$$

where λ, called the penalty coefficient, is a big constant which guarantees that the fitness of the best infeasible solution is poorer than that of the worst feasible solution.

A set of 0-1 knapsack problems were devised to validate SHLO as well as BPSO, MBDE, bFOA and ABHS. The numbers of items were set to 100, 200, 400, 600, 800, 1000, 1200, and 1500, and two instances of each scale were generated to test the performance of the algorithms more exactly. The weight w_j and the profit p_j were generated according to [16], i.e. from 5 to 20 and from 50 to 100, respectively. The weight capability C was set to 1000, 2400, 4000, 6000, 8000, 10000, 14000, and 16000, respectively. The population size and maximum generation of all the algorithms were set to 100, 10000 and 200, 40000 for the instances less and no less than 1000 items, respectively. The experimental results are presented in Table 2 and Table 3.

Table 2. The results of SHLO, BPSO, MBDE, bFOA and ABHS on low-dimensional 0-1 knapsack problems

	Algorithm	Best	Mean	Worst	Std
Kp100.1	SHLO	**6526**	**6526.0**	**6526**	**0.000**
	bFOA	6526	6525.9	6524	0.889
	ABHS	**6526**	**6526.0**	**6526**	**0.000**
	MBDE	6526	6524.2	6523	1.962
	BPSO	6526	6525.6	6522	1.265
Kp100.2	SHLO	**6824**	**6824.0**	**6824**	**0.000**
	bFOA	6824	6823.8	6823	0.731
	ABHS	**6824**	**6824.0**	**6824**	**0.000**
	MBDE	6824	6823.2	6822	0.872
	BPSO	6824	6822.9	6822	1.039
Kp200.1	SHLO	**14999**	**14999.0**	**14999**	**0.000**
	bFOA	14999	14998.0	14993	1.944
	ABHS	14999	14998.6	14997	0.843
	MBDE	**14999**	**14999.0**	**14999**	**0.000**
	BPSO	14999	14998.8	14997	0.632
Kp200.2	SHLO	**14791**	**14791.0**	**14791**	**0.000**
	bFOA	14791	14786.4	14780	4.879
	ABHS	14791	14787.7	14784	3.613
	MBDE	**14791**	**14791.0**	**14791**	**0.000**
	BPSO	**14791**	**14791.0**	**14791**	**0.000**
Kp400.1	SHLO	**27100**	**27099.1**	**27095**	**1.524**
	bFOA	27100	27094.9	27091	3.381
	ABHS	27097	27096.0	27086	4.879
	MBDE	27099	27095.4	27092	2.119
	BPSO	27100	27097.5	27092	3.126
Kp400.2	SHLO	27099	26448.7	26209	7.969
	bFOA	26657	26454.2	26253	1.643
	ABHS	26859	26425.5	26237	8.679
	MBDE	27099	26453.3	26092	2.119
	BPSO	**27099**	**26461.8**	**26250**	**3.512**
Kp600.1	SHLO	**40216**	**40216.0**	**40216**	**0.000**
	bFOA	40216	40212.4	40202	4.648
	ABHS	40216	40210.3	40204	8.644
	MBDE	40216	40212.7	40204	4.596
	BPSO	**40216**	**40216.0**	**40216**	**0.000**
Kp600.2	SHLO	**39602**	**39601.2**	**39594**	**17.769**
	bFOA	39602	39601.0	39597	12.236
	ABHS	39602	39599.5	39531	23.282
	MBDE	39602	39600.8	39583	11.155
	BPSO	39602	39600.3	39587	12.284

Table 3. The results of SHLO, BPSO, MBDE, bFOA and ABHS on high-dimensional 0-1 knapsack problems

	Algorithm	Best	Mean	Worst	Std
Kp800.1	SHLO	53855	53851.8	53837	5.473
	bFOA	53855	53845.8	53832	7.451
	ABHS	53850	53844.1	53822	4.838
	MBDE	53850	53841.9	53829	6.045
	BPSO	**53855**	**53851.9**	**53843**	**3.510**
Kp800.2	SHLO	**52705**	**52701.5**	**52692**	**8.697**
	bFOA	52703	52683.8	52667	13.312
	ABHS	52695	52691.3	52690	6.883
	MBDE	52692	52690.7	52689	1.528
	BPSO	52705	52698.3	52688	7.329
Kp1000.1	SHLO	**66882**	**66857.4**	**66844**	**13.082**
	bFOA	66867	66837.7	66829	14.930
	ABHS	66855	66840.2	66814	18.520
	MBDE	66860	66849.4	66828	9.009
	BPSO	66853	66830.8	66801	15.803
Kp1000.2	SHLO	**66905**	**66899.2**	**66898**	**23.122**
	bFOA	66891	66867.4	66849	17.097
	ABHS	66900	66887.1	66830	33.084
	MBDE	66905	66895.3	66893	16.429
	BPSO	66853	66841.6	66823	11.393
Kp1200.1	SHLO	**86823**	**86820.7**	**86805**	**5.559**
	bFOA	86805	86796.8	86776	10.497
	ABHS	86811	86710.5	86703	42.393
	MBDE	86812	86796.9	86776	10.775
	BPSO	86823	86810.2	86798	9.578
Kp1200.2	SHLO	**86715**	**86702.8**	**86698**	**8.927**
	bFOA	86701	86700.2	86694	1.304
	ABHS	86705	86698.8	86677	4.083
	MBDE	86701	86694.7	86686	6.506
	BPSO	86715	86702.2	86698	7.430
Kp1500.1	SHLO	**102657**	**102622.4**	**102551**	**28.982**
	bFOA	102608	102598.0	102586	29.541
	ABHS	102619	102575.9	102534	27.843
	MBDE	102603	102583.2	102563	12.674
	BPSO	102602	102523.1	102473	41.089
Kp1500.2	SHLO	**104860**	**104851.8**	**104748**	**30.506**
	bFOA	104860	104833.2	104742	36.573
	ABHS	104840	104820.8	104761	29.835
	MBDE	104831	104816.3	104754	19.655
	BPSO	104824	104801.6	104770	23.384

As can be seen in Table 2 and Table 3, SHLO finds the best numerical results on 14 out 16 instances and is only inferior to BPSO on Kp400.2 and Kp800.1, bFOA on Kp400.2, and MBDE on Kp400.2, respectively. For the instances in which the items are less than 200, all the algorithms can find the best-known solutions and achieve satisfactory results. When items increase to 1500, BPSO, ABHS and MBDE cannot reach the best-known values any more. Based on the ranking results given in Table 4, it is clear that SHLO outperforms BPSO, MBDE, bFOA and ABHS on the 0-1 knapsack problems.

Table 4. The ranks of SHLO, BPSO, MBDE, bFOA and ABHS on 0-1 knapsack problems

	SHLO	bFOA	ABHS	MBDE	BPSO
Kp100.1	1	3	1	5	4
Kp100.2	1	3	1	4	5
Kp200.1	1	5	4	1	3
Kp200.2	1	5	4	1	1
Kp400.1	1	5	3	4	2
Kp400.2	4	2	5	3	1
Kp600.1	1	4	5	3	1
Kp600.2	1	2	5	3	4
Kp800.1	2	3	4	5	1
Kp800.2	1	5	3	4	2
Kp1000.1	1	4	3	2	5
Kp1000.2	1	4	3	2	5
Kp1200.1	1	4	5	3	2
Kp1200.2	1	3	4	5	2
Kp1500.1	1	2	4	3	5
Kp1500.2	1	2	3	4	5
Average	1.25	3.50	3.56	3.25	3.00

4 Conclusion

Inspired by the mechanisms of human learning, this paper presents a novel meta-heuristic algorithm, named simple human learning optimization (SHLO), in which three learning operators, i.e. the random learning operator, individual learning operator, and social learning operator are developed by mimicking human learning behaviors to generate new solutions and search for the optimal solution of problems. To evaluate the performance of the proposed algorithm, low-dimensional and high-dimensional 0-1 KP benchmarks are adopted as benchmark problems to test SHLO. For a fair comparison, other four binary-coding optimization algorithms, i.e. BPSO, MBDE, bFOA, and ABHS, are also used to solve the benchmark problems with the recommended parameters. The experimental results demonstrate that SHLO outperforms BPSO, MBDE, bFOA and ABHS.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant No. 61304031, 61374044 & 61304143), and Innovation Program of Shanghai Municipal Education Commission (14YZ007).

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Boston (1989)
2. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Proceedings of First European Conference on Artificial Life. MIT Press/Bradford Books, Paris (1991)
3. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Network vol. 4, pp. 1942–1948. IEEE Press (1995)
4. Lee, K.S., Geem, Z.W.: A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice. Computer Methods in Applied Mechanics and Engineering 194, 3902–3933 (2005)
5. Pan, W.T.: A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example. Knowledge-Based Systems 26, 69–74 (2012)
6. Fister, J.I., Yang, X.S., Fister, I.: A Brief Review of Nature-Inspired Algorithms for Optimization. Elektrotehnski Vestnik 80, 1–7 (2013)
7. Forcheri, P., Molfino, M.T., Quarati, A.: ICT Driven Individual Learning: New Opportunities and Perspectives. Educational Technology & Society 3, 51–61 (2000)
8. Ickes, W., Gonzalez, R.: “Social” Cognition and Social Cognition: From the Subjective to the Intersubjective. Small Group Research 25, 294–315 (1994)
9. Ellis, A.P.J., Hollenbeck, J.R., Ilgen, D.R., Porter, C.O.L.H., West, B.J., Moon, H.: Team learning: Collectively connecting the Dots. Journal of Applied Psychology 88, 821–835 (2003)
10. Andrews, K.M., Delahaye, B.L.: Influences on Knowledge Processes in Organizational Learning: The Psychosocial Filter. Journal of Management Studies 37, 797–810 (2002)
11. McEvily, S.K., Chakravarthy, B.: The Persistence of Knowledge-based Advantage: An Empirical Test for Product Performance and Technological Knowledge. Strategic Management Journal 23, 285–305 (2002)
12. El-Maleh, A.H., Sheikh, A.T., Sait, S.M.: Binary Particle Swarm Optimization (BPSO) Based State Assignment for Area Minimization of Sequential Circuits. Applied Soft Computing 13, 4832–4840 (2013)
13. Wu, C.Y., Tseng, K.Y.: Topology Optimization of Structures Using Modified Binary Differential Evolution. Structural and Multidisciplinary Optimization 42, 939–953 (2010)
14. Wang, L., Zheng, X., Wang, S.: A Novel Binary Fruit Fly Optimization Algorithm for Solving the Multidimensional Knapsack Problem. Knowledge-Based Systems 48, 17–23 (2013)
15. Wang, L., Yang, R., Xu, Y.: An Improved Adaptive Binary Harmony Search Algorithm. Information Sciences 232, 58–87 (2013)
16. Zou, D., Gao, L., Li, S., Wu, J.: Solving 0-1 knapsack problem by a novel global harmony search algorithm. Applied Soft Computing 11, 1556–1564 (2011)