An Enhanced Discrete Human Learning Optimization for Permutation Flow Shop Scheduling Problem

Ling Wang[1], Mengzhen Wang[1], Jun Yao[1], and Muhammad Ilyas Menhas[2(✉)]

[1] Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200444, China
{wangling,Wangmengzhen,grandone0529}@shu.edu.cn

[2] Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, AJ&K, Pakistan

Abstract. Human learning optimization (HLO) is a promising meta-heuristic algorithm inspired by human learning mechanisms. As a binary-coded algorithm, HLO can be used for discrete problems without modifying the encoding scheme. However, as the scale of the problem grows, the “combinatorial explosion” phenomenon inevitably occurs, which increases the computational complexity and reduces the optimization efficiency of HLO. Therefore, this paper extends HLO and proposes an enhanced discrete HLO (EDHLO) algorithm for the Permutation Flow Shop Scheduling Problem (PFSSP), in which the original three learning operators are reformed and new improvement strategies for PFSSP are introduced. First, the Nawaz-Ensco-Ham (NEH) algorithm is utilized to initialize partial population to improve the quality of initial solutions. Second, two crossover operators are introduced into the individual learning of EDHLO to improve the diversity of the population and enhance the quality of the solution. Third, a local search approach is applied to manage escape from the local minimum. A total of 21 benchmark problems have been taken into account to evaluate the performance of EDHLO. The experimental outcomes and the comparison with other metaheuristics verify the effectiveness and superiority of the EDHLO algorithm.

Keywords: Human learning optimization · Permutation flow shop scheduling problem · Discrete human learning optimization

1 Introduction

In recent years, to break through the bottleneck limitation of traditional optimization algorithms, various intelligent optimization algorithms based on animal foraging or other learning behaviors, such as Artificial Bee Colony, Shuffled Frog Leaping Algorithm and Firefly Algorithm, have emerged. Compared with other animals in nature, human has evolved a higher level of intelligence and learning ability and is able to solve complex problems that other animals are unable to solve. Inspired by a simple yet general human learning mechanism, Wang et al. proposed Human Learning Optimization (HLO) [1],

© Springer Nature Singapore Pte Ltd. 2021
Q. Han et al. (Eds.): LSMS 2021/ICSEE 2021, CCIS 1469, pp. 245–257, 2021.
https://doi.org/10.1007/978-981-16-7213-2_24

in which the random learning operator, individual learning operator and social learning operator were designed through simulating the learning mechanism of humans to search out the optimal solution of problems.

To further improve the optimization performance of HLO, Wang et al. proposed an adaptive simplified human learning optimization algorithm (ASHLO), which set the linearly increasing and linearly decreasing adaptive strategy for the parameters pr and pi, respectively, to maintain the balance between algorithmic exploration and exploitation [2]. Based on the phenomenon that human IQ follows normal distribution on the whole and shows an upward trend, a diverse human learning optimization algorithm (DHLO) was proposed, where the social learning capability of all individuals was subject to normal distribution and tuned dynamically [3]. The simulation results indicate that DHLO possesses better global optimization performance. Yang et al. adopted an adaptive strategy based on cosine and sine functions to propose a sine-cosine adaptive human learning optimization algorithm (SCHLO) [4]. Later, An improved adaptive human learning optimization algorithm (IAHLO) [5] which used a two-stage adaptive strategy to dynamically adjust the execution probability of random learning operators is proposed, so that the algorithm focused on exploration to diversify the population at the beginning of iterations, and focused on local search at the later phase of iterations to enhance the mining ability of the algorithm. Nowadays, many various practical problems, such as extraction of text abstracts [6], financial market forecasts [4], image processing [7], mixed variable engineering optimization problems [8], and intelligent control [9–11], have been successfully solved by HLO.

As a binary algorithm, HLO is capable of solving discrete optimization problems directly. Fan W et al. adopted the ASHLO to take place local search in standard VNS to solve scheduling problems with multiple constraints [12]. Li et al. took advantage of HLO to solve the actual production scheduling problem of a dairy plant [13]. Ding et al. combined an improved PSO and some scheduling strategies to solve the flexible job shop scheduling problem under the algorithm architecture of HLO [14]. A. Shoja et al. combined ASHLO with GA and PSO, respectively, to solve the two-stage supply chain network design problem aiming at minimizing cost [15].

However, with the scale of the problem grows, the size of the feasible solution set grows exponentially, and the phenomenon of “combinatorial explosion” inevitably occurs, which results in a significant decrease in the optimization efficiency of binary HLO for discrete problems. Therefore, this paper extends HLO and proposes an enhanced discrete HLO (EDHLO) algorithm for the PFSSP, which is a classical discrete problem and plays an essential part in ensuring the steady progress of production and improving resource utilization. To solve PFSSP more efficiently, the learning operators of EDHLO are improved and the efficient heuristic algorithm NEH [16] is combined with random initialization to improve the quality of the initial solutions. Besides, a local search strategy is utilized to help EDHLO algorithm get rid of the local optima.

The structure of this paper is as follows: Sect. 2 is a description of the PFSSP and the related definitions. Section 3 describes the proposed algorithm EDHLO. The experimental results of EDHLO and other metaheuristics are given in Sect. 4. Section 5 summarizes this paper.

2 Permutation Flow Shop Scheduling Problem

PFSSP which is a simplified model of assembly-line production in many manufacturing enterprises, one of the most well-known classes of scheduling problems, has been investigated extensively and intensively due to its importance in the aspect of academy and engineering applications [17].

The current approaches for solving PFSSP mainly can be divided into 3 classifications, i.e., precision methods, heuristic rule-based methods and meta-heuristic algorithms [18]. PFSSP, as an NP-hard combinatorial optimization problem, grows exponentially in computational complexity as the problem scale increases. Although the precision methods can find the exact solution to the problem, they are only suitable for solving small-scale PFSSP considering the computational resources. Constructive heuristics are methods for solving specific problems by inductive reasoning and experimental analysis according to experience, which can obtain nearly optimal solutions. However, some of them are difficult to obtain satisfactory solutions. Compared with the other two categories, meta-heuristic algorithms are more advantageous in terms of stability, convergence speed and computational efficiency. Thus, currently, meta-heuristic algorithms have become the most effective and efficient method to solve scheduling problems, and the main hotspots of research are also concentrated on the improvement of the meta-heuristics and the development of new algorithms.

The description of PFSSP is as following: n jobs require m processes on different machines sequentially, and the processing sequence of n jobs on all machines is the same, with the goal to find the best permutation to minimize the makespan. It should be noted that each machine at the same time only one job is allowed to be machined, accordingly, each job can be handled on only one machine at a time. The processing time $H(p_j,k)$ is the time for job p_j to be processed on the k-th machine. $C(p_j,k)$ represents the processing completion time of job $p_j$on the k-th machine. PFSSP can be mathematically formulated as follows Eq. (1) [19]:

$$\begin{cases} C(p_1, 1) = H(1, 1) \\ C\big(p_j, 1\big) = H(j, 1) + C\big(p_{j-1}, 1\big) j = 2, 3, \ldots, n \\ C(p_1, k) = H(1, k) + C(p_1, k-1)\, k = 2, 3, \ldots, m \\ C(p_j, k) = H(j, k) + \max(C(p_{j-1}, k), C(p_j, k-1)) j = 2, 3, \ldots, n;\ k = 2, 3, \ldots, m \\ C_{\max} = C(p_n, m) \end{cases} . \tag{1}$$

The optimal permutation p^* can be found by the optimization in Eq. (2):

$$p^* = \arg\ \min\ C(p_n, m),\ \forall p \in \Omega. \tag{2}$$

3 Enhanced Discrete Human Learning Optimization for PFSSP

3.1 Initialization

The solution of PFSSP is a single linked list, which focuses on the position of each element in the whole permutation. Therefore, different from the standard HLO, EDHLO uses the integer encoding framework, where the individual solution in EDHLO is represented by an integer array as Eq. (3),

$$x_i = [x_{i1}\, x_{i2}\, \cdots\, x_{ij}\, \cdots\, x_{iM}],\ x_{ij} \in \{1, 2, \cdots, M\}, 1 \le i \le N, 1 \le j \le M \tag{3}$$

where x_i is the *i*-th individual, x_{ij} is the *j*-th element of the *i*-th individual, *N* is the size of population and *M* denotes the length of solutions, i.e. the number of jobs. Under the assumption that the prior knowledge of problems does not exist at first, each variable of an individual in EDHLO is initialized with an integer between 1 and *M* stochastically to present the number of jobs.

Aiming at solving PFSSP efficiently, the NEH algorithm, an efficient heuristic algorithm, is used to generate 10% of the individuals, so as to enhance the quality of the initial population and maintain the diversity of the population.

3.2 Learning Operators

The standard HLO generates new candidates to search out the optimal solution by the random learning operator, the individual learning operator and the social learning operator, but these three learning operators are not applicable to PFSSP. Thus, EDHLO redesigns and introduces the learning operators to solve PFSSP more efficiently.

3.2.1 Random Learning Operator for PFSSP-Like Problems

In reality, humans usually have to solve a problem by using the random learning strategy at the beginning due to a lack of prior knowledge. Moreover, since a person is easy to be affected by various factors like interference and forgetting, the human cannot fully replicate the previous experiences. And consequently, learning processes are always accompanied by randomness [20]. To imitate the random learning behavior for the PFSSP, the random learning operator is designed in EDHLO as Eq. (4),

$$x_{ij} = R(M) \tag{4}$$

where *R(M)* represents the random learning operator which randomly chooses a job not scheduled yet.

3.2.2 Individual Learning Operators for PFSSP-Like Problems

Individual learning refers to the ability of an individual to construct his own knowledge by reflecting on the extrinsic stimuli and sources [21]. Drawing on previous experience is conducive to efficient learning of humans. Set up an Individual Knowledge Database (IKD) to store individual optimum solutions to simulate the individual learning mechanism described above, as shown in Eqs. (5–6),

$$IKD = \begin{bmatrix} ikd_1 \\ ikd_2 \\ \vdots \\ ikd_i \\ \vdots \\ ikd_N \end{bmatrix}, \ 1 \leq i \leq N \tag{5}$$

$$ikd_i = \begin{bmatrix} ikd_{i1} \\ ikd_{i2} \\ \vdots \\ ikd_{ip} \\ \vdots \\ ikd_{iK} \end{bmatrix} = \begin{bmatrix} ik_{i11} & ik_{i12} & \cdots & ik_{i1j} & \cdots & ik_{i1M} \\ ik_{i21} & ik_{i22} & \cdots & ik_{i2j} & \cdots & ik_{i2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{ip1} & ik_{ip2} & \cdots & ik_{ipj} & \cdots & ik_{ipM} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{iK1} & ik_{iK2} & \cdots & ik_{iKj} & \cdots & ik_{iKM} \end{bmatrix}, 1 \leq p \leq K. \tag{6}$$

where ikd_i represents the IKD of individual i, K is the size of IKD, ikd_{ip} is the p-th best solution of individual i, and each ik_{ipj} is an integer between 1 and M.

EDHLO uses its previous knowledge in the IKD to generate a high-quality candidate by conducting the individual imitation learning operator (IILO) as Eq. (7).

$$x_{ij} = ik_{ipj} \tag{7}$$

Besides, to diversify the population and further enhance the quality of the solution, the individual swap learning operator (ISLO) and the individual reversal-insertion learning operator (IRLO), inspired and borrowed from two cross operators, i.e. the swap exploration operator and reversal-insertion operator, are introduced in the individual learning process. The swap exploration operator and reversal-insertion operator are successful ways of solving PFSSP in the previous studies, and their effectiveness has been verified in [22] and [23] respectively.

As shown in Fig. 1, the individual swap learning operator refers to exchanging the positions of two random jobs. The number of executions of the individual swap learning operator is thirty percent of the number of jobs. The individual reversal-insertion learning operation is described in Fig. 2 where it exchanges the position of two adjacent jobs, removes them from the original permutation and then inserts them in any other (n-2) positions. These two operators only need individual knowledge and therefore they are designed as individual learning operators. The newly generated permutation will replace the original one if and only if it has a better fitness.

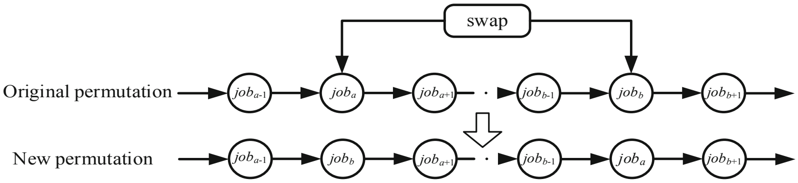

Fig. 1. The individual swap learning operator

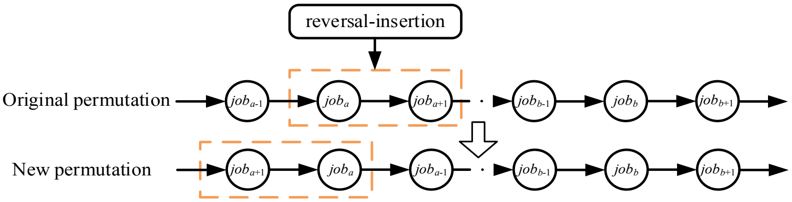

Fig. 2. The individual reversal-insertion learning operator

3.2.3 Social Learning Operator for PFSSP-Like Problems

The process of tackling complicated problems through individual learning alone can be quite slow and inefficient. Therefore, as social animals, human beings naturally learn from others in the collective to gain experience and expand their knowledge [24]. By directly or indirectly transferring experiences, humans can improve their competence and learning efficiency. To utilize the social experience productively, Social Knowledge Database (SKD) which stores the optimal solution found by the whole population in the search process is established in EDHLO as Eq. (8),

$$SKD = \begin{bmatrix} skd_1 \\ skd_2 \\ \vdots \\ skd_q \\ \vdots \\ skd_H \end{bmatrix} = \begin{bmatrix} sk_{11} & sk_{12} & \cdots & sk_{1j} & \cdots & sk_{1M} \\ sk_{21} & sk_{22} & \cdots & sk_{2j} & \cdots & sk_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{q1} & sk_{q2} & \cdots & sk_{qj} & \cdots & sk_{qM} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{H1} & sk_{H2} & \cdots & sk_{Hj} & \cdots & sk_{HM} \end{bmatrix}, 1 \leq q \leq H \tag{8}$$

where skd_q denotes the q-th optima solution in the SKD and H is the size of the SKD. When an individual performs social learning, it operates as Eq. (9).

$$x_{ij} = sk_{qj} \tag{9}$$

3.3 Updating of IKD and SKD

Although the individual learning operators and the social learning operator can learn the order of jobs scheduled for processing from the better solutions efficiently, they may produce duplicate job numbers and yield infeasible solutions. Therefore, after the new population is generated, EDHLO will find the duplicate job numbers from the random position of the individual sequence and replace them randomly with the job numbers that have not been assigned. Then the new candidates are substituted into the fitness function to calculate the fitness values. Only when the number of candidates stored in IKD is fewer than the set value K or the new candidate solution is superior to the worst in IKD, the new candidate solution will be saved in the IKD directly, otherwise, it will be discarded. For the SKD, the same updating strategy is applied. Moreover, to obtain a superior SKD and help the EDHLO escape from the local minimum, a local search approach is applied in EDHLO.

The local search method operates as follows: move each job in the optimal permutation with dimension M to the remaining (M-1) positions sequentially except the current position, and keep the rest of the permutation unchanged. After every movement, calculate the fitness of the new permutation. If the fitness is better, update the SKD. To avoid redundant computations, the local search is activated every 50 iterations.

3.4 Implementation of EDHLO

In summary, EDHLO performs the random learning operator, individual learning operators, and social learning operator with specified probabilities to search for the optimal solution. The implementation of EDHLO can be described as below:

Step 1: set the parameters of EDHLO, such as the population size, maximum number of iterations, *pr*, *pi;*
Step 2: initialize 10% of individuals by NEH and randomly yield the rest initial population;
Step 3: evaluate the fitness of the whole population and generate the initial IKDs and SKD;
Step 4: yield new candidates by executing the learning operators of EDHLO as Eq. (4), Eq. (7) and Eq. (9) with the probabilities of *pr*, (*pi*-*pr*) and (1-*pi*), respectively;
Step 5: fix the infeasible solutions and execute the ISLO and the IRLO for 20% of individuals;
Step 6: update the IKDs and SKD according to the calculated fitness of new individuals.
Step 7: perform the local search on the SKD every 50 iterations;
Step 8: Determine whether the termination condition is satisfied, if so, output the optimal solution found, otherwise go to step 4.

4 Experimental Results and Discussion

To estimate the performance, the proposed EDHLO, together with six meta-heuristic algorithms, i.e. the particle swarm optimization based memetic algorithm (PSOMA) [25], hybrid genetic algorithm (HGA) [26], discrete bat algorithm (DBA) [27], hybrid differential evolution algorithm with local search (L-HDE) [22], hybrid backtracking search algorithm (HBSA) [19] and chaotic whale algorithm (CWA) [23], are adopted to solve 21 benchmark problems proposed by Reeves. Referring to the literature [1], the IKD and SKD sizes of EDHLO are 1, and the remaining parameters of EDHLO are specified as given below: population size $popsize = 60$, maximum iterations $Gmax = 2000$, $pr = 0.1$, $pi = 0.45$. Run each instance 30 times independently to compare. The computations were carried out using a PC with AMD Ryzen 5 2600 Six-Core Processor CPU and 32GB RAM on Windows 10, 64-bit operating system.

The best relative error (BRE), the average relative error (ARE) and the worst relative error (WRE) of the optimal solution are selected as the evaluation metrics, which are defined as Eqs. (10–12),

$$BRE = \frac{S_{Best} - BKS}{BKS} \times 100\%. \tag{10}$$

$$ARE = \frac{S_{Average} - BKS}{BKS} \times 100\%. \tag{11}$$

$$WRE = \frac{S_{Worst} - BKS}{BKS} \times 100\%. \tag{12}$$

where S_{Best}, $S_{Average}$ and S_{Worst} represent the best solution, the average solution and the worst solution found by the algorithm respectively, and BKS represents the optimal solution value known so far of the benchmark problem.

Table 1 lists the experimental data of EDHLO. The corresponding data of compared algorithms are obtained from the original literature and the experimental results not given in the original literature are indicated by "–".

Figures 3 and 4 show the average values of the three evaluation metrics on the Reeves benchmark problems. Since the original literature of CWA contains results for only 19 Reeves benchmarks, for the sake of fairness, CWA is compared with EDHLO separately. It can be seen in Figs. 3 and 4 that EDHLO achieves the best results. Compared with PSOMA, HGA, DBA, L-HDE and HBSA, EDHLO ranks first in terms of the mean of BRE, ARE and WRE, with the lowest values of 0.258, 0.564 and 0.628, respectively. According to the values of the mean of BRE, ARE and WRE, EDHLO is also superior to CWA. The above experimental outcomes prove the effectiveness and stability of EDHLO.

Table 1. Comparison of EDHLO and other algorithms on Reeves benchmarks

Problem	n*m	BK		PSOM	HG	DB	L-	HBS	CW	EDHL
REC0	20*5	124	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	0.144	0.14	0.08	**0**	0.14	**0**	**0**
			WR	0.16	–	0.16	**0**	0.16	**0**	**0**
REC0	20*5	110	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	0.189	0.09	0.08	**0**	0.08	**0**	0.018
			WR	0.721	–	0.18	**0**	0.18	**0**	0.180
REC0	20*5	124	BR	0.242	**0**	0.24	0.242	0.24	**0**	**0**
			AR	0.249	0.29	0.24	0.242	0.24	**0**	**0**
			WR	0.402	–	0.24	0.242	0.24	**0**	**0**
REC0	20*1	156	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	0.986	0.69	0.57	**0**	0.46	0.16	**0**
			WR	1.149	–	1.14	**0**	1.15	0.83	**0**
REC0	20*1	153	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	0.621	0.64	0.63	0.026	0.07	0.04	**0**
			WR	1.691	–	2.40	0.260	0.65	0.32	**0**
REC1	20*1	143	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	0.129	1.1	1.16	**0**	**0**	**0**	**0**
			WR	0.978	–	2.65	**0**	**0**	**0**	**0**
REC1	20*1	193	BR	0.259	0.36	0.41	**0**	0.1	**0**	**0**
			AR	0.893	1.68	1.46	0.275	0.53	0.45	**0.140**
			WR	1.502	–	3.78	0.777	1.14	0.82	**0.259**
REC1	20*1	195	BR	0.051	0.56	0.15	**0**	0.05	**0**	**0**
			AR	0.628	1.12	1.22	0.523	0.64	0.57	**0.179**
			WR	1.076	–	2.10	1.180	1.18	1.02	**0.462**
REC1	20*1	190	BR	**0**	0.95	0.36	**0**	**0**	**0**	**0**
			AR	1.33	2.32	1.27	0.363	1	0.67	**0.074**
			WR	2.155	–	2.15	0.946	2.16	1.41	**0.368**

(*continued*)

Table 1. (*continued*)

Problem	n*m	BK		PSOM	HG	DB	L-	HBS	CW	EDHL
REC1	30*1	209	BR	0.43	0.62	0.57	0.287	0.29	0.28	**0.143**
			AR	1.313	1.32	0.92	0.702	0.81	0.53	**0.397**
			WR	2.102	–	2.02	1.242	1.29	1.05	**0.860**
REC2	30*1	201	BR	1.437	1.44	1.43	0.645	0.69	0.64	**0.406**
			AR	1.596	1.57	1.67	1.279	1.5	1.47	**1.234**
			WR	1.636	–	2.23	1.438	1.83	1.63	**1.438**
REC2	30*1	201	BR	0.596	0.40	0.79	0.348	0.45	**0.34**	**0.348**
			AR	1.31	0.87	1.17	0.428	1.28	0.85	**0.417**
			WR	2.038	–	2.38	0.497	3.08	1.93	**1.044**
REC2	30*1	251	BR	0.835	1.27	1.63	0.557	0.4	–	**0.279**
			AR	2.085	2.54	2.92	1.082	1.29	–	**0.712**
			WR	3.233	–	3.94	1.632	2.43	–	**1.353**
REC2	30*1	237	BR	1.348	1.10	1.01	0.253	0.25	**0.01**	0.253
			AR	1.605	1.83	1.41	0.851	1.27	1.00	**0.839**
			WR	2.402	–	2.29	1.222	2.57	1.72	**1.054**
REC2	30*1	228	BR	1.442	1.4	1.04	0.831	0.57	0.49	**0.087**
			AR	1.888	2.7	2.58	1.049	1.42	1.24	**0.822**
			WR	2.492	–	3.93	1.443	2.97	2.31	**1.443**
REC3	50*1	304	BR	1.51	0.43	2.29	0.427	0.43	0.42	**0.263**
			AR	2.254	1.34	3.39	**0.644**	1.91	0.99	0.785
			WR	2.692	–	4.53	**0.920**	2.66	1.47	1.478
REC3	50*1	311	BR	**0**	**0**	0.61	**0**	**0**	**0**	**0**
			AR	0.645	0.78	0.72	0.244	0.59	**0**	0.328
			WR	0.834	–	1.73	0.835	1.28	**0**	0.739
REC3	50*1	327	BR	**0**	**0**	**0**	**0**	**0**	**0**	**0**
			AR	**0**	**0**	0.03	**0**	**0**	**0**	**0**
			WR	**0**	–	0.09	**0**	**0**	**0**	**0**
REC3	75*2	495	BR	2.101	3.75	3.37	2.565	1.92	–	**1.827**
			AR	3.537	4.9	4.87	3.001	2.93	–	**2.467**
			WR	4.039	–	5.97	3.555	4.2	–	**3.272**
REC3	75*2	508	BR	1.553	2.2	2.28	1.730	0.9	**0.08**	0.806
			AR	2.426	2.79	3.85	1.832	1.88	1.57	**1.246**
			WR	2.83	–	5.34	2.005	3.38	1.98	**1.553**
REC4	75*2	496	BR	2.641	3.64	3.81	2.661	1.69	1.45	**1.008**
			AR	3.684	4.92	5.09	3.350	2.72	2.34	**2.185**
			WR	4.052	–	6.53	3.770	3.55	3.04	**2.742**

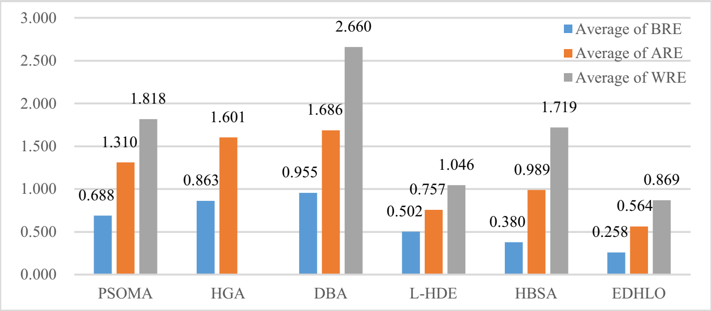

Fig. 3. The comparison of PSOMA, HGA, DBA, L-HDE, HBSA and EDHLO on Reeves benchmarks

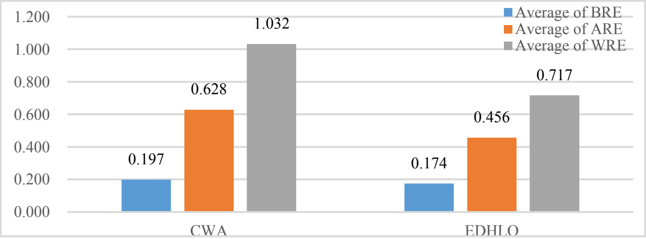

Fig. 4. The comparison of CWA and EDHLO on Reeves benchmarks

5 Conclusions

To further study HLO and extend it to solve discrete problems more efficiently, this paper proposed an enhanced discrete human learning optimization (EDHLO) algorithm for solving PFSSP. EDHLO is based on the standard learning mechanisms of HLO, but it redesigns and introduces the learning operators to solve PFSSP efficiently. In the proposed EDHLO, the efficient heuristic algorithm NEH is integrated with random initialization to improve the original population quality and maintain the diversity of the original population, two cross operators are introduced into the individual learning of EDHLO to upgrade the local exploitation capacity, and a local search method is used to facilitate EDHLO to get rid of the local optimum. The experimental results on 21 benchmark problems and the comparisons with other meta-heuristics verify the effectiveness and stability of the EDHLO.

Acknowledgments. This work is supported by National Key Research and Development Program of China (No. 2019YFB1405500), National Natural Science Foundation of China (Grant No. 92067105 & 61833011), Key Project of Science and Technology Commission of Shanghai Municipality under Grant No. 19510750300 & 19500712300, and 111 Project under Grant No. D18003.

References

1. Wang, L., Ni, H.Q., Yang, R.X., et al.: A simple human learning optimization algorithm. In: Fei, M., Peng, C., Su, Z., Song, Y., Han, Q. (eds.) LSMS/ICSEE 2014, Part II. CCIS, vol. 462, pp. 56–65. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45261-5_7
2. Wang, L., Ni, H.Q., Yang, R.X., et al.: An adaptive simplified human learning optimization algorithm. Inf. Sci. **320**, 126–139 (2015)
3. Wang, L., An, L., Pi, J., Fei, M., Pardalos, P.M.: A diverse human learning optimization algorithm. J. Global Optim. **67**(1–2), 283–323 (2016). https://doi.org/10.1007/s10898-016-0444-2
4. Yang, R., Xu, M., He, J., Ranshous, S., Samatova, N.F.: An intelligent weighted fuzzy time series model based on a sine-cosine adaptive human learning optimization algorithm and its application to financial markets forecasting. In: Cong, G., Peng, W.-C., Zhang, W.E., Li, C., Sun, A. (eds.) ADMA 2017. LNCS (LNAI), vol. 10604, pp. 595–607. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69179-4_42
5. Wang, L., Pei, J., Wen, Y.L., et al.: An improved adaptive human learning algorithm for engineering optimization. Appl. Soft Comput. **71**, 894–904 (2018)
6. Alguliyev, R., Aliguliyev, R., Isazade, N.: A sentence selection model and HLO algorithm for extractive text summarization. In: 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), pp. 206--209 (2016)
7. Bhandari, A.K., Kumar, I.V.: A context sensitive energy thresholding based 3D Otsu function for image segmentation using human learning optimization. Appl. Soft Comput. **82**, 105570 (2019)
8. Wang, L., Pei, J., Menhas, M.I., et al.: A Hybrid-coded human learning optimization for mixed-variable optimization problems. J. Knowl-Based Syst. **127**, 114–125 (2017)
9. Han, Z., Qi, H., Wang, L., Menhas, M.I., Fei, M.: Water level control of nuclear power plant steam generator based on intelligent virtual reference feedback tuning. In: Li, K., Zhang, J., Chen, M., Yang, Z., Niu, Q. (eds.) ICSEE/IMIOT -2018. CCIS, vol. 925, pp. 14–23. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-2381-2_2
10. Menhas, M.I., Wang, L., Ayesha, N.U., et al.: Continuous human learning optimizer based PID controller design of an automatic voltage regulator system. In: 2018 Australian & New Zealand Control Conference (ANZCC), pp. 148–153 (2018)
11. Wen, Y., Wang, L., Peng, W., Menhas, M.I., Qian, L.: Application of intelligent virtual reference feedback tuning to temperature control in a heat exchanger. In: Li, K., Fei, M., Du, D., Yang, Z., Yang, D. (eds.) ICSEE/IMIOT -2018. CCIS, vol. 924, pp. 311–320. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-2384-3_29
12. Fan, W., Pei, J., Liu, X., et al.: Serial-batching group scheduling with release times and the combined effects of deterioration and truncated job-dependent learning. J. Glob. Optim. **71**(1), 147–163 (2017)
13. Li, X.Y., Yao, J., Wang, L., et al.: Application of human learning optimization algorithm for production scheduling optimization. In: Fei, M., Ma, S., Li, X., Sun, X., Jia, L., Su, Z. (eds.) Advanced Computational Methods in Life System Modeling and Simulation. ICSEE 2017, LSMS 2017. Communications in Computer and Information Science, vol. 761, pp. 242--252. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-6370-1_24
14. Ding, H.J., Gu, X.S.: Hybrid of human learning optimization algorithm and particle swarm optimization algorithm with scheduling strategies for the flexible job-shop scheduling problem. Neurocomputing **414**, 313–332 (2020)
15. Shoja, A., Molla-Alizadeh-Zavardehi, S., Niroomand, S.: Hybrid adaptive simplified human learning optimization algorithms for supply chain network design problem with possibility of direct shipment. Appl. Soft Comput. **96**, 106594 (2020)

16. Nawaz, M., et al.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega **11**, 91–95 (1983)
17. Reeves, C.R.: A genetic algorithm for flowshop sequencing. Comput. Oper. Res. **22**(1), 5–13 (1995)
18. Zwied, A.N.H., Ismal, M.M., Mohamed, S.S.: Permutation flow shop scheduling problem with makespan criterion: literature review. J. Theor. Appl. Inf. Technol. **99**, 4 (2021)
19. Lin, Q., Gao, L., Li, X., et al.: A hybrid backtracking search algorithm for permutation flow-shop scheduling problem. Comput. Ind. Eng. **85**, 437–446 (2015)
20. Ali, M.Z., Awad, N.H., Suganthan, P.N., et al.: A modified cultural algorithm with a balanced performance for the differential evolution frameworks. J. Knowl.-Based Syst. **111**, 73–86 (2016)
21. Forcheri, P., Molfino, M.T., Quarati, A.: ICT driven individual learning: new opportunities and perspectives. J. Educ. Technol. Soc. **3**, 51–61 (2000)
22. Liu, Y., Yin, M., Gu, W.: An effective differential evolution algorithm for permutation flow shop scheduling problem. Appl. Math. Comput. **248**, 143–159 (2014)
23. Li, J., Guo, L.H., Li, Y., et al.: Enhancing whale optimization algorithm with chaotic theory for permutation flow shop scheduling problem. Int. J Comput. Intell. Syst. **14**(1), 651–675 (2021)
24. Roediger, H.L.: Reflections on intersections between cognitive and social psychology: a personal exploration. J. Eur. J. Soc. Psychol. **40**(2), 189–205 (2010)
25. Liu, B., Wang, L., Jin, Y.H.: An effective PSO-based memetic algorithm for flow shop scheduling. IEEE Trans. Syst. Man Cybern. B Cybern. 37(1), 18–27 (2007)
26. Tseng, L.-Y., Lin, Y.-T.: A hybrid genetic algorithm for no-wait flowshop scheduling problem. Int. J. Prod. Econ. **128**(1), 144–152 (2010)
27. Luo, Q., Zhou, Y., Xie, J., et al.: Discrete bat algorithm for optimal problem of permutation flow shop scheduling. Sci. World J. **2014**, 630–280 (2014)