

# Application of Human Learning Optimization Algorithm for Production Scheduling Optimization

Xiaoyu Li<sup>1</sup>, Jun Yao<sup>1</sup>(✉), Ling Wang<sup>1</sup>,  
and Muhammad Ilyas Menhas<sup>2</sup>

<sup>1</sup> School of Mechatronic Engineering and Automation, Shanghai University,  
Shanghai 200072, China

grandone0529@shu.edu.cn

<sup>2</sup> Department of Electrical (Power) Engineering, Mirpur University of Science  
and Technology, MUST, Mirpur AJ&K, Pakistan

**Abstract.** In this paper, Human Learning Optimal (HLO) algorithm is presented to solve the scheduling problem. HLO is a meta-heuristic search algorithm which is inspired by the process of human learning. Three learning operators are developed to generate new solutions and search for the optima by mimicking the learning behaviors of human. This new algorithm has been proved to be very effective in solving optimization problems. HLO is applied to solve an actual production scheduling problems in a dairy factory and the performance of HLO is compared with that of two other meta-heuristics algorithms, BSO-PSO and HGA. Comparison results demonstrate that HLO is a promising optimization algorithm.

**Keywords:** Human learning optimization · Meta-heuristic · Scheduling problem

## 1 Introduction

Scheduling optimization problems are wide ranging and plentiful, which people always find in various fields of engineering, scientific, economic management and so on [1]. However, optimization problems are becoming more and more complicated with the development of science and technology, and traditional gradient-based methods are inefficient and inconvenient for such problems as they require substantial gradient information, depend on a well-define starting point, and need a large amount of enumeration memory. Meta-heuristics mimic nature biological systems to solve various kinds of optimization problems. For instance, Genetic Algorithms (GAs) [2], Particle

---

J. Yao—This work was financially supported by the Science and Technology Commission of Shanghai Municipality of China under Grant (No.17511107002).

Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], Tabu Search [5] are popular with optimizers. In recent years, more and more novel algorithms have been proposed to tackle optimization problems, such as the Binary Differential Evolution algorithm (BDE) [6], the Binary Artificial Bee Colony Algorithm (BABCA) [7], the Binary Bat Algorithm (BBA) [8], the Binary Flower Pollination Algorithm (BFPA) [9], the Binary Gravitation Search Algorithm (BGSA) [10], the Binary Simulated Annealing Algorithm (BSAA) [11], and the Bi-Velocity Discrete Particle Swarm Optimization (BVDPSO) [12].

To solve hard optimization problems more effectively and efficiently, new powerful meta-heuristics inspired by nature, especially by biological systems, must be explored, which is a hot topic in evolutionary computation community [13]. As is known to all, human beings are the smartest creatures in the earth and invent numbers of machines and tools to make our lives convenient, which indicates human beings are gifted to tackle many complicated problems. People master and improve skills through repeatedly learning which is similar to iteratively searching for best solutions by optimal algorithms. The process can be considered as an optimization of iterative process. For the example of learning Sudoku, a person may learn randomly due to the 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 books (social learning) [14, 15]. Inspired by human learning mechanisms, a new meta-heuristic algorithm called human learning optimization (HLO) algorithm is presented. The performance of HLO to solve the continuous optimization problems such as a suit of numerical benchmark functions, deceptive problems, 0–1 knapsack problems [16, 17] has been proved. HLO will be applied to an actual production scheduling problems in this work.

The rest of the paper is organized as follows. Section 2 introduces the presented HLO in detail. In Sect. 3, HLO is applied to an actual production scheduling problems in a dairy factory and the results are compared with those of other meta-heuristics collected from recent works to validate its performance. Finally, Sect. 4 gives a conclusion of the work in this paper.

## 2 Human Learning Optimization Algorithm

### 2.1 Initialization

HLO adopts the binary-coding framework in which each bit corresponds to a basic component of knowledge to solve problems. Therefore, an individual, i.e. a candidate solution, is represented by a binary string as Eq. (1) which is initialized as “0” or “1” randomly assuming that there is no prior-knowledge of problems,

$$x_i = \begin{bmatrix} x_{i11} & x_{i12} & \cdots & x_{i1j} & \cdots & x_{i1M} \\ x_{i21} & x_{i22} & \cdots & x_{i2j} & \cdots & x_{i2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{ip1} & x_{ip2} & \cdots & x_{ipj} & \cdots & x_{ipM} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{iP1} & x_{iP2} & \cdots & x_{iPj} & \cdots & x_{iPM} \end{bmatrix} \tag{1}$$

$$x_{ipj} \in \{0, 1\}, 1 \leq p \leq P, 1 \leq j \leq M$$

where  $x_i$  is the  $i$ th individual,  $P$  is the number of product, and  $M$  is the number of machine i.e.

After all the individuals are initialized, the initial population of HLO is generated as Eq. (2).  $N$  is the number of individuals of the population.

$$X = [x_1 \quad x_2 \quad \cdots \quad x_i \quad \cdots \quad x_N]^T, 1 \leq i \leq N \tag{2}$$

## 2.2 Learning Operators

**Random Exploration Learning Operator.** While learning to solve unfamiliar problems, people usually learn randomly because of lacking or forgetting knowledge. Simulating these phenomena, HLO performs the random exploration learning with some probability as Eq. (3)

$$x_{ij} = RE(0, 1) = \begin{cases} 0, & 0 \leq rand() \leq 0.5 \\ 1, & else \end{cases} \tag{3}$$

Where  $RE(0, 1)$  is a random number in  $[0, 1)$ .

**Individual Learning Operator.** Individual learning is defined as the ability to build knowledge through individual reflection about external stimuli and sources [18]. Every person learns in conscious or unconscious states, which is a fundamental requirement of existence. In HLO, an individual learns to solve problems by the individual learning operator based on its own experience which is stored in the Individual Knowledge Database (IKD) as Eqs. (4) and (5),

$$x_{ipj} = ik_{ipj} \tag{4}$$

$$IKD_i = \begin{bmatrix} ikd_{i1} \\ ikd_{i2} \\ \vdots \\ ikd_{ip} \\ \vdots \\ ikd_{iP} \end{bmatrix} = \begin{bmatrix} ik_{i11} & ik_{i12} & \cdots & ik_{i1j} & \cdots & ik_{i1M} \\ ik_{i21} & ik_{i22} & \cdots & ik_{i2j} & \cdots & ik_{i2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{ip1} & ik_{ip2} & \cdots & ik_{ipj} & \cdots & ik_{ipM} \\ \vdots & \vdots & & \vdots & & \vdots \\ ik_{iP1} & ik_{iP2} & \cdots & ik_{iPj} & \cdots & ik_{iPM} \end{bmatrix} \quad (5)$$

$$IKD = [ikd_1 \quad ikd_2 \quad \cdots \quad ikd_i \quad \cdots \quad ikd_N]^T$$

$$1 \leq i \leq N, 1 \leq i \leq N, 1 \leq p \leq P, 1 \leq j \leq M$$

where  $IKD_i$  is the individual knowledge database of person  $i$  which stands for the  $i$ th best solution of person  $I$  and  $P$  denotes the size of the IKDs.

**Social Learning Operator.** Social learning is a transmission of knowledge and skills through direct or indirect interactions among individuals. In the social context, people can learn from not only their own direct experience but also the experience of the other members, and therefore they can develop further their abilities and achieve the higher efficiency with an effective knowledge sharing. To possess the efficient search ability, the social learning mechanism is mimicked in HLO. Like human learning, each individual of HLO studies the social knowledge which is stored in the Social Knowledge Database (SKD) with some probability as Eqs. (6) and (7) when it yields a new solution,

$$x_{ipj} = sk_{pj} \quad (6)$$

$$SKD = \begin{bmatrix} sk_{11} & sk_{12} & \cdots & sk_{1j} & \cdots & sk_{1M} \\ sk_{21} & sk_{22} & \cdots & sk_{2j} & \cdots & sk_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{p1} & sk_{p2} & \cdots & sk_{pj} & \cdots & sk_{pN} \\ \vdots & \vdots & & \vdots & & \vdots \\ sk_{P1} & sk_{P2} & \cdots & sk_{Pj} & \cdots & sk_{PN} \end{bmatrix} \quad (7)$$

$$1 \leq p \leq P, 1 \leq j \leq M$$

where the SKD is the best knowledge of the social collectivities. The  $S$  best individuals are selected as initial social knowledge database stored in the SKD. In the following iterated searching, the IKD and SKD will be updated if new knowledge is better than that in the IKD and SKD.

In summary, HLO yields a new solution by means of random exploration learning, individual learning and social learning with certain rates, which can be simplified and formulated as

$$x_{ipj} = \begin{cases} RE(0,1) & 0 \leq rand \leq pr \\ ikd_{ipj} & pr \leq rand \leq pi \\ skd_{pj} & else \end{cases} \tag{8}$$

where  $x_{ipj}$  is the productivity of the  $p$ th product on  $j$ th machine,  $pr$  is the probability of random exploration learning,  $(pi-pr)$  and  $(1-pi)$  represent the rates of individual learning and social learning, respectively.

### 2.3 Updating of the IKD and SKD

After individuals accomplish learning in each generation, the fitness values of new solutions that evaluated through the fitness function  $f(x)$  are obtained. If fitness values of new candidate solutions are better than the worst one in the IKD, or the dimension of individual knowledge which is stored in the IKD is less than  $N \times P$ , the new candidate solutions will be saved in the IKD. In the same way, the SKD is updated. The SKD is updated by at least one solution every iteration in case of falling into local optimum.

## 3 Establishment of the Model and Experimental Results

### 3.1 Establishment of the Model

Scheduling problems have a major impact on the productivity of a manufacturing system, which can be described as follows. Given a number of tasks which must be carried out by some processors, it is required to find the best resource assignments and tasks sequencing, For example, the total completion time needs to be as small as possible. The study of production scheduling problem in this paper is based on actual production procedure in a dairy factory. The efficiency and capacity of each product on each machine are fixed and the demands for the output of dairy products in daily order are variant. The purpose of production scheduling is to distribute different kinds of dairy products to different kinds of equipment and minimize production time. The production scheduling model can be described as follows.

$$\begin{aligned} \min T &= \max(t_1, t_2, \dots, t_i, \dots, t_m) \\ t_i &= \sum_{j=1}^n \left( \frac{c_{ij}}{\eta_{ij} \nu_{ij}} + b \times tc_{ij} \right) \\ s.t. \quad c_i &= \sum_{j=1}^n c_{ij} \end{aligned} \tag{9}$$

where  $t_i$  is one day runtime of the  $i$ th machine,  $c_{ij}$  is the amount of the  $i$ th product on the  $j$ th machine,  $\eta_{ij}$  is the Productivity of the  $i$ th product on  $j$ th machine,  $\nu_{ij}$  is the capacity of the  $i$ th product on  $j$ th machine,  $b$  is 0 or 1, If it needs to switch to other milk after the  $i$ th product,  $b$  is equal to 1 and if not,  $b$  is equal to 0. The number of  $i$ th production on the  $j$ th machine  $c_{ij}$  is between  $\min c_{ij}$  and  $\max c_{ij}$  as Eq. (10)

$$\min c_{ij} \leq c_{ij} \leq \max c_{ij} \tag{10}$$

In summary, the procedure of HLO for scheduling problems in a dairy factory can be concluded as follows:

- 
- Step 1: Set the parameters and initialize the population of HLO;
  - Step 2: Repair the initial solution as Eq. (10), calculate the fitness of initial individuals and initialize the IKDs and SKD ;
  - Step 3: If reaching the maximum generation, the results will be output; otherwise go to step 5.
  - Step 4: Generate new individuals by performing the three learning operators as Eq. (8);
  - Step 5: If the new individuals is feasible solution, go to step 7, otherwise repair the new solution as Eq. (10);
  - Step 6: Evaluate the fitness of each individual and update the IKD, SKD ;
  - Step 7: If reaching the maximum generation, the results will be output; otherwise repeat step 4-step 6.
- 

### 3.2 Experimental Results and Discussions

Production scheduling based on HLO is carried out according to production plan of a certain day in a dairy factory to verify the effectiveness of HLO. 25 kinds of milk products need to be produced on the 10 machines. The efficiency and capacity of each product on each machine are show in Tables 1 and 2. Table 3 is orders of one day in a dairy factory.

**Table 1.** Capacity on each machine.

(piece/h)10 <sup>3</sup>	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
P1	18	20	18	29	29	29	39	39	39	39
P2	18	20	18	/	/	/	39	39	39	39
P3	18	20	18	29	29	29	/	/	/	/
P4	15	/	18	/	/	/	38	38	38	38
P5	/	/	18	29	29	29	35	35	35	35
P6	/	/	/	29	29	29	35	35	35	35
P7	/	19	/	29	29	29	/	/	/	/
P8	18	19	/	29	29	29	/	/	/	/
P9	18	16	16	/	/	/	35	35	35	35
P10	18	16	16	25	25	25	/	/	/	/
P11	16	18	17	/	/	/	30	30	30	30
P12	16	19	/	/	/	/	30	30	30	30

(continued)

**Table 1.** (continued)

(piece/h)10 <sup>3</sup>	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
P13	15	16	16	/	/	/	30	30	30	30
P14	18	20	/	25	25	25	/	/	/	/
P15	/	20	18	/	/	/	30	30	30	30
P16	/	18	20	25	25	25	/	/	/	/
P17	/	/	/	25	25	25	30	30	30	30
P18	16	/	18	25	25	25	/	/	/	/
P19	/	20	/	/	/	/	30	30	30	30
P20	/	/	/	20	20	20	30	30	30	30
P21	/	/	/	20	20	20	/	/	/	/
P22	/	/	/	20	20	20	30	30	30	30
P23	15	20	19	/	/	/	30	30	30	30
P24	/	/	/	/	/	/	30	30	30	30
P25	/	19	20	/	/	/	30	30	30	30

**Table 2.** Efficiency on each machine.

(0 ~ 1)	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
P1	0.6	0.7	0.65	0.8	0.78	0.62	0.54	0.48	0.72	0.45
P2	0.58	0.72	0.65	/	/	/	0.66	0.71	0.58	0.67
P3	0.42	0.68	0.68	0.62	0.71	0.68	/	/	/	/
P4	0.62	/	0.55	/	/	/	0.77	0.64	0.63	0.71
P5	/	/	0.77	0.55	0.53	0.56	0.62	0.65	0.48	0.51
P6	/	/	/	0.51	0.43	0.53	0.47	0.61	0.55	0.48
P7	/	0.63	/	0.55	0.61	0.43	/	/	/	/
P8	0.6	0.45	/	0.57	0.44	0.48	/	/	/	/
P9	0.62	0.57	0.64	/	/	/	0.61	0.65	0.45	0.61
P10	0.57	0.7	0.54	0.62	0.61	0.74	/	/	/	/
P11	0.44	0.63	0.76	/	/	/	0.55	0.65	0.59	0.61
P12	0.7	0.54	/	/	/	/	0.44	0.38	0.62	0.58
P13	0.55	0.33	0.46	/	/	/	0.51	0.56	0.77	0.49
P14	0.62	0.5	/	0.71	0.58	0.62	/	/	/	/
P15	/	0.73	0.73	/	/	/	0.68	0.55	0.57	0.61
P16	/	0.71	0.68	0.48	0.55	0.57	/	/	/	/
P17	/	/	/	0.58	0.59	0.44	0.47	0.6	0.49	0.52
P18	0.54	/	0.64	0.52	0.65	0.58	/	/	/	/
P19	/	0.82	/	/	/	/	0.71	0.62	0.65	0.55
P20	/	/	/	0.6	0.71	0.58	0.45	0.49	0.51	0.61
P21	/	/	/	0.7	0.68	0.69	/	/	/	/
P22	/	/	/	0.65	0.56	0.71	0.68	0.62	0.58	0.71
P23	0.32	0.45	0.55	/	/	/	0.54	0.51	0.43	0.47
P24	/	/	/	/	/	/	0.61	0.58	0.54	0.59
P25	/	0.66	0.43	/	/	/	0.54	0.47	0.42	0.51

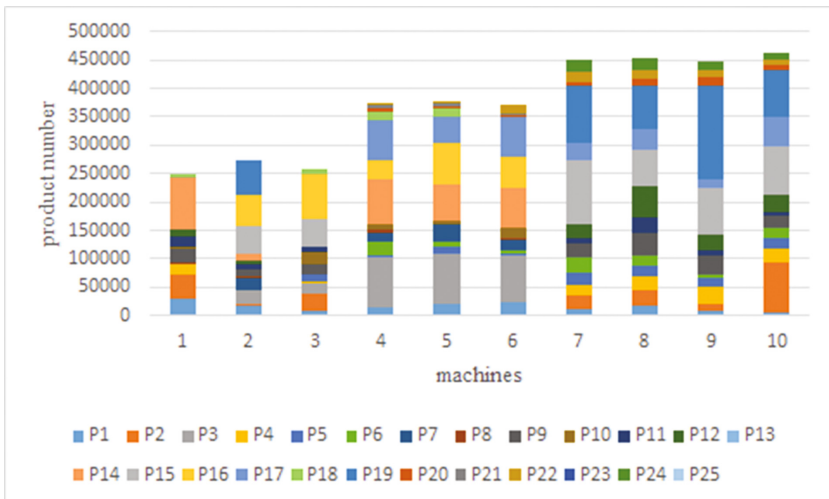
**Table 3.** Orders of each production.

Product	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Output (10 <sup>3</sup> )	100	150	200	80	60	56	50	8	100	40
Product	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Output (10 <sup>3</sup> )	50	80	/	200	280	180	/	28	320	30
Product	P21	P22	P23	P24	P25					
Output (10 <sup>3</sup> )	10	50	/	40	/					

The performance of HLO is compared with the BSO-PSO (Brain Storm Optimization with Discrete Particle Swarm Optimization) [19] and HGA (Hybrid Genetic Algorithms) [20]. All experimental tests were implemented on a PC of Intel Core CPU i7-2700 K @ 3.50 GHz with 8 GB RAMs. A set of fair parameters are chosen for all three algorithms in this paper. For example, population size is set to 20 and iteration times are 300. Other diverse parameters that are used in HLO, BSO-PSO, and HGA are listed in Table 4.

**Table 4.** Parameters settings of HLO, BSO-PSO, and HGA.

Algorithms	Parameters	
HLO	$Pr = 0.01$	$Pi = 0.07$
BSO-PSO	$Pone = 0.6$	$Pinverse = 0.8$
HGA	$PC = 0.9$	$Pm = 0.02$



**Fig. 1.** Histogram of equipment production on each machine.



Histograms of task arrangement are given in Fig. 1 and histograms of runtime are given in Fig. 2 to display the task arrangement.

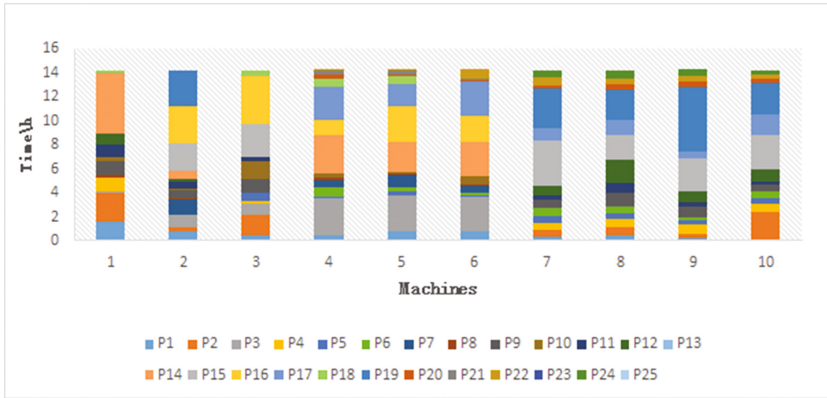


Fig. 2. Histogram of Runtime on each machine.

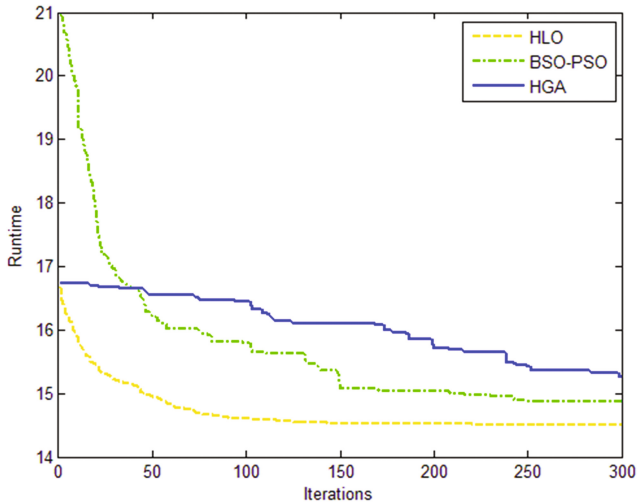
Figure 1 displays the task arrangement and illustrates that the task is evenly distributed to each machine. Figure 2 indicates that the run time of each machine is almost the same and no machine is idle. Therefore the utilization of machines are improved.

Table 5 shows that HLO, BSO-PSO and HGA find solutions respectively after running 30 times. Row “Best” denotes the best solution from each algorithm. Row “Mean” denotes the mean value of the total run solutions from each algorithm. Row “Worst” denotes the worst solution found from each algorithm. As we can see, all results from HLO in Table 5 are better than other two algorithms.

Table 5. Result of HLO, BSO-PSO, HGA.

Algorithms	HLO	BSO-PSO	HGA
Best	14.57	14.96	15.32
Mean	14.93	15.19	16.04
Worst	16.5	18.3	18.01

Figure 3 illustrates the iteration curves of the optimal algorithms in this paper. It is no surprise that the convergence speed of HLO is faster than both BSO-PSO and HGA.



**Fig. 3.** The iteration curves of HLO, BSO-PSO and HGA.

## 4 Conclusion

In this paper, a novel human learning optimization algorithm (HLO) is presented which is inspired by the human learning process. In this method, three learning operations, i.e. the random learning operator, the individual learning operator, and the social learning operator, are developed by mimicking human learning behaviors to generate new solutions and search for the optimal solution of problems. The performance of this proposed method is validated by applying to an actual production scheduling problems in a dairy factory. The experimental results show that the performance of the proposed method is better than the performance of compared methods.

## References

1. Mullen, R.J., Monekosso, D., Barman, S., et al.: A review of ant algorithms. *J. Expert Syst. Appl.* **36**(6), 9608–9617 (2009)
2. Elsayed, S.M., Sarker, R.A., Essam, D.L.: A new genetic algorithm for solving optimization problems. *J. Eng. Appl. Artif. Intell.* **27**(1), 57–69 (2014)
3. Fang, W., Sun, J., Chen, H., et al.: A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population. *J. Inf. Sci.* **330**, 19–48 (2016)
4. Prakasam, A., Savarimuthu, N.: Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants. *J. Artif. Intell. Rev.* **45** (1), 1–34 (2015)
5. Qin, J., Xu, X., Wu, Q., et al.: Hybridization of tabu search with feasible and infeasible local searches for the quadratic multiple knapsack problem. *J. Comput. Oper. Res.* **66**, 199–214 (2016)
6. Chen, Y., Xie, W., Zou, X.: A binary differential evolution algorithm learning from explored solutions. *Neurocomputing* **149**, 1038–1047 (2015)

7. Zhang, X., Zhang, X.: A binary artificial bee colony algorithm for constructing spanning trees in vehicular ad hoc networks. *Ad Hoc Netw.* **58**, 198–204 (2017)
8. Mirjalili, S., Mirjalili, S.M., Yang, X.S.: Binary bat algorithm. *Neural Comput. Appl.* **25**(3–4), 1–19 (2014)
9. Rodrigues, D., Silva, G.F.A., Papa, J.P., et al.: EEG-based person identification through binary flower pollination algorithm. *Expert Syst. Appl.* **62**, 81–90 (2016)
10. Automatic channel selection in EEG signals for classification of left or right hand movement in Brain Computer Interfaces using improved binary gravitation search algorithm
11. Li, X., Ma, L.: Minimizing binary functions with simulated annealing algorithm with applications to binary tomography. *Comput. Phys. Commun.* **183**(2), 309–315 (2012)
12. Shen, M., Zhan, Z.H., Chen, W.N., Gong, Y.J., Zhang, J., Li, Y.: Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks. *IEEE Trans. Ind. Electron.* **61**(12), 7141–7151 (2014)
13. Fister, J.I., Yang, X.S., Fister, I.: A brief review of nature-inspired algorithms for optimization. *Elektroteh. Vestn.* **80**(3), 1–7 (2013)
14. Wang, L., Ni, H., Yang, R., et al.: A simple human learning optimization algorithm. *J. Commun. Comput. Inf. Sci.* **462**, 56–65 (2014)
15. Wang, L., An, L., Pi, J., et al.: A diverse human learning optimization algorithm. *J. Global Optim.* **67**(1–2), 1–41 (2016)
16. Wang, L., Yang, R., Ni, H., et al.: A human learning optimization algorithm and its application to multi-dimensional knapsack problems. *J. Appl. Soft Comput.* **34**, 736–743 (2015)
17. Wang, L., Ni, H., Yang, R., et al.: An adaptive simplified human learning optimization algorithm. *J. Inf. Sci.* **320**, 126–139 (2015)
18. Forcheri, P., Molfino, M.T., Quarati, A.: ICT driven individual learning: new opportunities and perspectives. *J. Educ. Technol. Soc.* **3**(1), 51–61 (2000)
19. Hua, Z., Chen, J., Xie, Y.: Brain storm optimization with discrete particle swarm optimization for TSP. In: 2016 12th International Conference on Computational Intelligence and Security (CIS), pp. 190–193. IEEE (2016)
20. Kundakci, N., Kulak, O.: Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *J. Comput. Ind. Eng.* **96**, 31–51 (2016)