Learner Performance-based Behavior Optimization Algorithm: A Functional Case Study

Shahad Ahmed  
Da Vinci Studio School

Tarik A. Rashid  (✉ tarik.ahmed@ukh.edu.krd )  
University of Kurdistan Hewler

Method Article

Keywords: Metaheuristic, Learner Performance-based Behavior, Optimization Algorithm

Posted Date: June 15th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1688246/v4

License: This work is licensed under a Creative Commons Attribution 4.0 International License. 
Read Full License
Learner Performance-based Behavior Optimization
Algorithm: A Functional Case Study

Shahad A. Salih1,2, Tarik A. Rashid3
1 Computer Science Dept., University of Duhok, Duhok, KRG, Iraq
2 Da Vinci Studio School, Duhok, KRG, Iraq
3 Computer Science and Engineering Dept., University of Kurdistan Hewler, Erbil, KRG, Iraq

Abstract. A novel algorithm called learner performance-based behavior algorithm (LPB) was proposed for single and multi-objective by Chnoor M. Rahman and Tarik A. Rashid in 2021. LPB proved its ability to deal with complex optimization problems compared to the dragonfly algorithm (DA), genetic algorithm (GA), and particle swarm optimization (PSO). This paper presents and explains the implementation of the LPB algorithm, and it applies it as a model in a case study to maximize a fitness function. As a result, the LPB algorithm is successfully improved the initial population and achieved the optimal solution.

Keywords: Metaheuristic, Learner Performance-based Behavior, Optimization Algorithm.

1 Introduction

Metaheuristic refers to higher-level heuristics, which have been developed for solving a variety of optimization problems. Various metaheuristic algorithms have recently been successful in tackling intractable situations. The advantage of employing these algorithms to solve complicated problems is that they produce the approximate solutions in a short amount of time, even for very complex problems [1]. Optimization is used in almost all areas of our lives, such as engineering, medicine, business planning, control, energy, etc. These algorithms intelligently select the best solutions from a wide range of choices [2]. Some widely used optimization algorithms are extracted from natural systems, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) [3].

Recently, a learner performance-based behavior algorithm (LPB) has been proposed by Chnoor M. Rahman and Tarik A. Rashid in 2021 [4], [5] to mainly tackle single and multi-objective problems. The main contribution of the paper is to apply LPB to a case study to maximize and obtain optimal solutions. A simple step-by-step guide is demonstrated for this purpose. The paper also serves as a guide for researchers to develop, enhance or hybridize the algorithm.
The rest of the paper is organized into sections, where section two explains LPB in brief. In section three, a case study has been designed to evaluate LPB. Finally, in section four, the main points are concluded.

2 Learner Performance-based Behavior Algorithm

LPB algorithm is inspired by the idea of accepting graduates from high school to university. Through the steps that are applied during the admission of Learners, which are the methods used to divide learners and group them according to their cumulative rate. Also, these methods are used to improve the behavior and level of performance of the individuals after admission to the departments. Learners need to use new study habits because the methods they used to study in junior high do not work properly in college [6, 7, 8]. In this algorithm as a first step, a group of individuals is selected from the population. These individuals are then divided into sub-groups, and the best individuals are then selected from the subgroups depending on their fitness. Their behavior and performance are then improved by having them work as groups. Where teamwork will provide information sharing among themselves when they study (crossover), this method will affect their behavior randomly (mutation). LPB uses crossover and mutation techniques of the genetic algorithm. LPB algorithm works as shown in Figure 1.

Fig. 1 LPB Pseudocode
Figure 2 shows the flowchart of LPB.

Fig. 2 LPB algorithm flowchart

Figure 3 shows how to divide M population flowchart.
Fig. 3 Divide M population flowchart

Fig. 4 Apply Crossover & Mutation
3 A Case Study

For a maximization type of optimization as opposed to a minimization type; consider the following function: \( f(x) \), where \( f(x) = x_1^2 + x_2^2 \); for integer \( x_1 \), \( 0 \leq x_1 \leq 5000 \) and \( 0 \leq x_2 \leq 5000 \).

**Step 1:** let \( M \) consists of 16 individuals. Then evaluate the fitness of all \( M \) individuals by the fitness equation. Calculate the summation, average and find maximum fitness form \( M \) as shown in Table 1:

**Table 1.** Randomly create an \( M \) population and evaluate the fitness of all individuals.

| \(Mi\) | \(X_1\) | \(X_2\) | \(X_1^2\) | \(X_2^2\) | Fitness (\(Mi\)) = \(X_1^2 + X_2^2\) |
|-------|--------|--------|--------|--------|----------------------------------|
| M1    | 3770   | 3362   | 14,212,900 | 11,303,044 | 25,515,944                     |
| M2    | 703    | 1264   | 494,209    | 1,597,696  | 2,091,905                      |
| M3    | 3235   | 2191   | 10,465,225 | 4,800,481  | 15,265,706                     |
| M4    | 3989   | 3940   | 15,912,121 | 15,523,600 | 31,435,721                     |
| M5    | 1716   | 1065   | 2,944,656  | 1,134,225  | 4,078,881                      |
| M6    | 2394   | 4020   | 5,731,236  | 16,160,400 | 21,891,636                     |
| M7    | 1600   | 2559   | 2,560,000  | 6,548,481  | 9,108,481                      |
| M8    | 3639   | 2013   | 13,242,321 | 4,052,169  | 17,294,490                     |
| M9    | 3500   | 3275   | 12,250,000 | 10,725,625 | 22,975,625                     |
| M10   | 1890   | 1105   | 3,572,100  | 1,221,025  | 4,793,125                      |
| M11   | 4044   | 4038   | 16,353,936 | 16,305,444 | 32,659,380                     |
| M12   | 4065   | 1122   | 16,524,225 | 1,258,884  | 17,783,109                     |
| M13   | 2673   | 3580   | 7,144,929  | 12,816,400 | 19,961,329                     |
| M14   | 3829   | 1473   | 14,661,241 | 2,169,729  | 16,830,970                     |
| M15   | 245    | 2484   | 60,025     | 6,170,256  | 6,230,281                      |
| M16   | 2212   | 3382   | 4,892,944  | 11,437,924 | 16,330,868                     |

Sum fit. (\(Mi\))= 264,247,451
Average fit. (\(Mi\))= 16,515,466
Max fit. (\(Mi\))= 32,659,380

**Step 2:** Randomly create a subpopulation; let \( O \) consists of 8 individuals. Then Sort \( O \) descending order. Divide \( O \) into two groups; Good and Bad. Select that individual with the highest fitness from good and bad as shown in Table 2.
Table 2. O population, finding the highest fitness from Good and Bad groups.

| Mi   | Oi | Fitness (Oi) | Groups |
|------|----|--------------|--------|
| M11  | O1 | 32,659,380   | Good   |
| M9   | O2 | 22,975,625   |        |
| M8   | O3 | 17,294,490   |        |
| M14  | O4 | 16,830,970   |        |
| M16  | O5 | 16,330,868   | Bad    |
| M7   | O6 | 9,108,481    |        |
| M10  | O7 | 4,793,125    |        |
| M5   | O8 | 4,078,881    |        |

**Good High Fitness is O1 = 32,659,380**  
**Bad High Fitness is O5 = 16,330,868**

**Step 3:** Compare $M$ individuals with $O$ Good and Bad highest fitness to divide $M$.

Table 3. Comparing $M$ individuals with $O$ Good and Bad highest fitness to divide $M$, Fitness (Fit.).

| Mi   | Fitness (Mi) | Fit. (Mi) $<=$ Fit. (O5) | Fit. (Mi) $<=$ Fit. (O1) | Perfect Group |
|------|--------------|-------------------------|--------------------------|---------------|
| M1   | 25,515,944   | Good                    |                          |               |
| M2   | 2,019,905    | Bad                     |                          |               |
| M3   | 15,265,706   | Bad                     |                          |               |
| M4   | 31,435,721   | Good                    |                          |               |
| M5   | 4,078,881    | Bad                     |                          |               |
| M6   | 21,891,636   | Good                    |                          |               |
| M7   | 9,108,481    | Bad                     |                          |               |
| M8   | 17,294,490   | Good                    |                          |               |
| M9   | 22,975,625   | Good                    |                          |               |
| M10  | 4,793,125    | Bad                     |                          |               |
| M11  | 32,659,380   | Good                    |                          |               |
| M12  | 17,783,109   | Good                    |                          |               |
| M13  | 19,961,329   | Good                    |                          |               |
| M14  | 16,830,970   | Good                    |                          |               |
| M15  | 6,230,281    | Bad                     |                          |               |
| M16  | 16,330,868   | Bad                     |                          |               |
**Step 4:** Check if the Perfect population is not empty, select individuals from PF. If the PF is empty select individuals from the Good population (GP) when it’s not empty, if the GP is empty, then select individuals from the Bad population (BP). The selected individuals will be used in the crossover operator. Note that the number of selected individuals equals the number of required individuals $N$, which we specify in the first step.

**Table 4.** Selected individuals.

| Selected Individuals to Crossover with O: Good Individua’s: |   |
|-----------------------------------------------------------|--|
| M1 25,515,944 Good | M4 31,435,721 Good |
| M6 21,891,636 Good | M12 17,783,109 Good |

**Step 5:** Apply crossover between selected individuals from Table 4 and good individuals from Table 2.

**Table 5.** Apply Crossover.

| P, Ch | Ind. | $X_1$ | $X_2$ | $X_1$ (Binary) | $X_2$ (Binary) |
|-------|------|-------|-------|----------------|----------------|
| P1    | M1   | 3770  | 3362  | 1110 1011 1010 | 1101 0010 0010 |
| P2    | O1   | 4044  | 4038  | 1111 1100 1100 | 1111 1100 0110 |
| Ch1   | New  | 3724  | 3334  | 1110 1000 1100 | 1111 1111 0100 |
| Ch2   | New  | 4090  | 4066  | 1111 1111 1010 | 1111 1110 0010 |
| P3    | M4   | 3989  | 3940  | 1111 1001 0101 | 1111 0110 0100 |
| P4    | O2   | 3500  | 3275  | 1101 1010 1100 | 1100 1100 1101 |
| Ch3   | New  | 4012  | 3915  | 1111 1010 1100 | 1111 0100 1011 |
| Ch4   | New  | 3477  | 3300  | 1101 1001 0101 | 1100 1110 0100 |
| P5    | M6   | 2394  | 4020  | 1001 0101 1010 | 1111 1011 0100 |
| P6    | O3   | 3639  | 2013  | 1110 0011 1111 | 0111 1111 1101 |
| Ch5   | New  | 2423  | 3997  | 1001 0111 0111 | 1111 1001 1101 |
| Ch6   | New  | 3610  | 2036  | 1110 0001 1010 | 0111 1111 0100 |
| P7    | M12  | 4065  | 1122  | 1111 1110 0001 | 0100 0110 0010 |
| P8    | O4   | 3829  | 1473  | 1110 1111 0101 | 0101 1110 0001 |
| Ch7   | New  | 4085  | 1089  | 1111 1111 0101 | 0100 0100 0001 |
| Ch8   | New  | 3809  | 1506  | 1110 1110 0001 | 0101 1110 0010 |

**Step 6:** Apply mutation on new individuals (Child), to maximize the function (randomly convert 0 --> 1) 1 bit for each individual.
Table 6. Apply mutation.

| Child | X1 | X2  | X1 Binary | X2 Binary |
|-------|----|-----|-----------|-----------|
| Ch1   | 3724 | 3334 | 1110 1000 1100 | 1101 0000 0110 |
| Ch2   | 4090 | 4066 | 1111 1111 1010 | 1111 1110 0010 |
| Ch3   | 4012 | 3915 | 1111 1010 1100 | 1111 0100 1011 |
| Ch4   | 3477 | 3300 | 1101 1001 0101 | 1100 1110 0100 |
| Ch5   | 2423 | 3997 | 1001 0111 0111 | 1111 1001 1101 |
| Ch6   | 3610 | 2036 | 1110 0001 1010 | 0111 1111 0100 |
| Ch7   | 4085 | 1089 | 1111 1111 1010 | 1000 0100 0001 |
| Ch8   | 3809 | 1506 | 1110 1110 0001 | 0101 1110 0010 |

New individuals are shown in Table 7:

Table 7. New individuals after mutation.

| X1 new (Binary) | X2 new (Binary) | X1  | X2  |
|-----------------|-----------------|-----|-----|
| 1110 1000 1100  | 1111 0000 0110  | 3788 | 3846 |
| 1111 1111 1011  | 1111 1111 0010  | 4091 | 4082 |
| 1111 1110 1100  | 1111 1000 1011  | 4076 | 4043 |
| 1101 1001 0111  | 1110 1110 0100  | 3479 | 3812 |
| 1001 1111 0111  | 1111 1001 1111  | 2551 | 3999 |
| 1110 0101 1010  | 1111 1111 0100  | 3674 | 4084 |
| 1111 1111 1101  | 0100 1100 0001  | 4093 | 1217 |
| 1111 1110 0001  | 0101 1111 0010  | 4065 | 1522 |

Step 7: Calculate the fitness of new individuals as shown in Table 8.

Table 8. Calculate Fitness for New Individuals.

| Child | X1 | X2 | X1^2 | X2^2 | Fitness (Chi) = X1^2 + X2^2 |
|-------|----|----|------|------|-----------------------------|
| Ch1   | 3788 | 3846 | 14,348,944 | 14,791,716 | 29,140,660 |
| Ch2   | 4091 | 4082 | 16,736,281 | 16,662,724 | 33,399,005 |
| Ch3   | 4076 | 4043 | 16,613,776 | 16,345,849 | 32,969,625 |
| Ch4   | 3479 | 3812 | 12,103,441 | 14,531,344 | 26,634,785 |
| Ch5   | 2551 | 3999 | 6,507,601  | 15,992,001 | 22,499,602 |
| Ch6   | 3674 | 4084 | 13,498,276 | 16,679,056 | 30,177,332 |
| Ch7   | 4093 | 1217 | 16,752,649 | 1,481,089  | 18,233,738 |
| Ch8   | 4065 | 1522 | 16,524,225 | 2,316,484  | 18,840,709 |
**Step 8:** Find the sum, average, and max between new individuals from Table 8 and the parents (Pi) from Table 5.

**Table 9. Find Sum, Average, and Max.**

| Ind. | X₁ | X₂ | X¹² | X²² | Fitness (Ind.) = X¹² + X²² |
|------|----|----|-----|-----|---------------------------|
| M1   | 3770 | 3362 | 14,212,900 | 11,303,044 | 25,515,944 |
| O1   | 4044 | 4038 | 16,353,936  | 16,305,444  | 32,659,380 |
| M4   | 3989 | 3940 | 15,912,121  | 15,523,600  | 31,435,721 |
| O2   | 3500 | 3275 | 12,250,000  | 10,725,625  | 22,975,625 |
| M6   | 2394 | 4020 | 5,731,236   | 16,160,400  | 21,891,636 |
| O3   | 3639 | 2013 | 13,242,321  | 4,052,169   | 17,294,490 |
| M12  | 4065 | 1122 | 16,524,225  | 1,258,884   | 17,783,109 |
| O4   | 3829 | 1473 | 14,661,241  | 2,169,729   | 16,830,970 |
| Ch1  | 3788 | 3846 | 14,348,944  | 14,791,716  | 29,140,660 |
| Ch2  | 4091 | 4082 | 16,736,281  | 16,662,724  | 33,399,005 |
| Ch3  | 4076 | 4043 | 16,613,776  | 16,345,849  | 32,959,625 |
| Ch4  | 3479 | 3812 | 12,103,441  | 14,531,344  | 26,634,785 |
| Ch5  | 2551 | 3999 | 6,507,601   | 15,992,001  | 22,499,602 |
| Ch6  | 3674 | 4084 | 13,498,276  | 16,679,056  | 30,177,332 |
| Ch7  | 4093 | 1217 | 16,752,649  | 1,481,089   | 18,233,738 |
| Ch8  | 4065 | 1522 | 16,524,225  | 2,316,484   | 18,840,709 |

**Sum fitness = 398,272,331**  
**Average fitness = 24,892,021**  
**Max fitness = 33,399,005**

The new population is shown in table 9. Table 10 just rename the (Ind.) column to (Mi) to show that it’s the New M Population. That should generate O subpopulation from it.

**Table 10. The new M population.**

| Mi  | X₁ | X₂ | X¹² | X²² | Fitness (Ind.) = X¹² + X²² |
|-----|----|----|-----|-----|---------------------------|
| M1  | 3770 | 3362 | 14,212,900 | 11,303,044 | 25,515,944 |
| M2  | 4044 | 4038 | 16,353,936  | 16,305,444  | 32,659,380 |
| M3  | 3989 | 3940 | 15,912,121  | 15,523,600  | 31,435,721 |
| M4  | 3500 | 3275 | 12,250,000  | 10,725,625  | 22,975,625 |
| M5  | 2394 | 4020 | 5,731,236   | 16,160,400  | 21,891,636 |
| M6  | 3639 | 2013 | 13,242,321  | 4,052,169   | 17,294,490 |
The previous steps (2 to 8) will be repeated till the required number of iterations or the stop condition is met, then the optimal solution is returned.

Repeat steps 2 to 8, to do the second iteration.

**Step 2:** Create a new $O$ sub-population from new $M$ population table 10.

**Table 11.** Randomly create $O$ population and select the highest fitness from Good and Bad.

| Mi   | Oi | Fitness ($O_i$) | Groups |
|------|----|-----------------|--------|
| M11  | O1 | 32,959,625      | Good   |
| M3   | O2 | 31,435,721      |        |
| M9   | O3 | 29,140,660      |        |
| M12  | O4 | 26,634,785      |        |
| M1   | O5 | 25,515,944      |        |
| M16  | O6 | 18,840,709      | Bad    |
| M15  | O7 | 18,233,738      |        |
| M7   | O8 | 17,783,109      |        |

**Good High Fitness is $O_1 = 32,959,625$**

**Bad High Fitness is $O_5 = 25,515,944$**

**Step 3:** Compare the new $M$ individuals table 10 with $O$ Good and Bad highest fitness.
Table 12. Comparing M individuals with O Good and Bad highest fitness.

| Mi | Fitness (Mi) | Fit. (Mi) =< Fit. (O5) | Fit. (Mi) =< Fit. (O1) | Perfect Group |
|----|--------------|----------------|----------------|----------------|
| M1 | 25,515,944   | Bad            |                 |                |
| M2 | 32,659,380   |                | Good            |                |
| M3 | 31,435,721   |                | Good            |                |
| M4 | 22,975,625   | Bad            |                 |                |
| M5 | 21,891,636   | Bad            |                 |                |
| M6 | 17,294,490   | Bad            |                 |                |
| M7 | 17,783,109   | Bad            |                 |                |
| M8 | 16,830,970   | Bad            |                 |                |
| M9 | 29,140,660   |                | Good            |                |
| M10| 33,399,005   |                | Perfect         |                |
| M11| 32,959,625   | Good           |                 |                |
| M12| 26,634,785   | Good           |                 |                |
| M13| 22,499,602   | Bad            |                 |                |
| M14| 30,177,332   | Good           |                 |                |
| M15| 18,233,738   | Bad            |                 |                |
| M16| 18,840,709   | Bad            |                 |                |

**Step 4:** Check if the Perfect population is not empty select individuals from PF, if the PF is empty, select individuals from the good population when it’s not empty, if the GP is empty, select individuals from the BP. The selected individuals will be used in the crossover operator.

**Note:** The number of selected individuals will equal the number of required individuals \( N \), which we specify in the first step.

Table 13. Selected individuals.

| Selected Individuals to Crossover with O Good Individual’s: |
|------------------------------------------------------------|
| M10  | 33,399,005 | Perfect |
| M2   | 32,659,380 | Good    |
| M14  | 30,177,332 | Good    |
| M4   | 22,975,625 | Bad     |

**Step 5:** Apply crossover between selected individuals from Table 13 and good individuals from Table 11.
### Table 14. Apply Crossover.

| P, Ch | Ind. | X₁ | X₂ | X₁ (Binary) | X₂ (Binary) |
|-------|------|----|----|-------------|-------------|
| P1    | M10  | 4091 | 4082 | 1111 1111 1011 | 1111 1111 0010 |
| P2    | O1   | 4076 | 4043 | 1111 1110 1100 | 1111 1100 1011 |
| Ch1   | New  | 4076 | 4043 | 1111 1110 1100 | 1111 1100 1011 |
| Ch2   | New  | 4091 | 4082 | 1111 1111 1011 | 1111 1111 0010 |
| P3    | M2   | 4044 | 4038 | 1111 1100 1100 | 1111 1100 0110 |
| P4    | O2   | 3989 | 3940 | 1111 1001 0101 | 1111 0110 0100 |
| Ch3   | New  | 4053 | 4068 | 1111 1101 0101 | 1111 1110 0100 |
| Ch4   | New  | 3980 | 3910 | 1111 1000 1100 | 1111 0100 0110 |
| P5    | M14  | 3674 | 4084 | 1110 0101 1010 | 1111 1111 0100 |
| P6    | O3   | 3788 | 3846 | 1110 1100 1100 | 1111 0000 0110 |
| Ch5   | New  | 3660 | 4038 | 1110 0100 1100 | 1111 1100 0110 |
| Ch6   | New  | 3802 | 3910 | 1110 1101 1010 | 1111 0011 0100 |
| P7    | M4   | 3500 | 3275 | 1101 1010 1100 | 1100 1100 1011 |
| P8    | O4   | 3479 | 3812 | 1101 1001 0111 | 1110 1100 0100 |

**Step 6:** Apply mutation on new individuals (Child) to maximize the function (randomly convert 0 bit to 1 bit) 1 bit for each individual.

### Table 15. Apply mutation process on the new child.

| Child | X₁ | X₂ | X₁ Binary | X₂ Binary |
|-------|----|----|-----------|-----------|
| Ch1   | 4076 | 4043 | 1111 1110 1100 | 1111 1100 1011 |
| Ch2   | 4091 | 4082 | 1111 1111 0111 | 1111 1111 0010 |
| Ch3   | 4053 | 4068 | 1111 1101 0101 | 1111 1110 0100 |
| Ch4   | 3980 | 3910 | 1111 1000 1100 | 1111 0100 0110 |
| Ch5   | 3660 | 4038 | 1110 0100 1100 | 1111 1100 0110 |
| Ch6   | 3802 | 3892 | 1110 1101 0111 | 1110 1100 0100 |
| Ch7   | 3479 | 3300 | 1101 1001 0111 | 1100 1110 0100 |
| Ch8   | 3500 | 3787 | 1101 1010 1100 | 1110 1100 1011 |

New individuals are shown in Table 16.
Step 7: Calculate the fitness of new individuals by fitness equation, as shown in Table 17.

| Child | $X_1$ | $X_2$ | $X_1^2$ | $X_2^2$ | Fitness (Chi) = $X_1^2 + X_2^2$ |
|-------|-------|-------|---------|---------|-----------------------------------|
| Ch1   | 4092  | 4047  | 16,744,464 | 16,378,209 | 33,122,673 |
| Ch2   | 4095  | 4090  | 16,769,025 | 16,728,100 | 33,497,125 |
| Ch3   | 4061  | 4069  | 16,491,721 | 16,556,761 | 33,048,482 |
| Ch4   | 4044  | 3942  | 16,353,936 | 15,539,364 | 31,893,300 |
| Ch5   | 3662  | 4054  | 13,410,244 | 16,434,916 | 29,845,160 |
| Ch6   | 4058  | 3956  | 16,467,364 | 15,649,936 | 32,117,300 |
| Ch7   | 3991  | 3316  | 15,928,081 | 10,995,856 | 26,923,937 |
| Ch8   | 3564  | 3819  | 12,702,096 | 14,584,761 | 27,286,857 |

Step 8: Find the sum, average, and max between new individuals from Table 17 and the parents (Pi) from Table 14.

| Ind. | $X_1$ | $X_2$ | $X_1^2$ | $X_2^2$ | Fitness (Ind.) = $X_1^2 + X_2^2$ |
|------|-------|-------|---------|---------|-----------------------------------|
| M10  | 4091  | 4082  | 16,736,281 | 16,662,724 | 33,399,005 |
| O1   | 4076  | 4043  | 16,613,776 | 16,345,849 | 32,959,625 |
| M14  | 4084  | 4084  | 15,912,121 | 15,523,600 | 31,435,721 |
| M1   | 3770  | 3362  | 14,212,900 | 11,303,044 | 25,515,944 |
Thus, it can be noted from the results that the population has been improved and the efficiency of individuals is increased. Table 19 shows a comparison between the summation and the average and the optimal value for each iteration.

Table 19. Comparison between the results of 3 iterations.

| Iteration | Sum       | Average   | Max        |
|-----------|-----------|-----------|------------|
| Iteration 0 | 264,247,451 | 16,515,466 | 32,659,380 |
| Iteration 1 | 398,272,331 | 24,892,021 | 33,399,005 |
| Iteration 2 | 489,657,286 | 30,603,580 | 33,497,125 |

4 Conclusion

This study presents Learner based Performance algorithm. A case study is designed to describe the crossover and mutation processes that may confuse readers of this algorithm. In the experimental results, LBP proved its potential in improving and developing qualities and likewise finding the optimal solution. LPB improves and obtains better solutions iteratively. The authors recommend that this algorithm can be enhanced to reduce the processing time. In addition, changing the GA’s crossover and mutation operations with other operations or mathematical equations from other competitive optimization algorithms might lead to a better performance of LBP.
References

1. Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. and Cosar, A., 2019. A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering, 137, p.106040.
2. Hussain, K., Mohd Salleh, M.N., Cheng, S. and Shi, Y., 2019. Metaheuristic research: a comprehensive survey. Artificial Intelligence Review, 52(4), pp.2191-2233.
3. Wong, W.K. and Ming, C.I., 2019, June. A review on metaheuristic algorithms: recent trends, benchmarking and applications. In 2019 7th International Conference on Smart Computing & Communications (ICSCC) (pp. 1-5). IEEE.
4. Rahman, C.M. and Rashid, T.A., 2021. A new evolutionary algorithm: Learner performance based behavior algorithm. Egyptian Informatics Journal, 22(2), pp.213-223.
5. Rahman, C.M., Rashid, T.A., Ahmed, A.M., Seyedali Mirjalili, 2022. Multi-objective learner performance-based behavior algorithm with five multi-objective real-world engineering problems. Neural Comput & Applic. https://doi.org/10.1007/s00521-021-06811-z
6. Qayyum, A., 2018. Student help-seeking attitudes and behaviors in a digital era. International Journal of Educational Technology in Higher Education, 15(1), pp.1-16.
7. Chew, S.L., 2010. Improving classroom performance by challenging student misconceptions about learning. Association for Psychological Science - APS. Available at: https://www.psychologicalscience.org/observer/improving-classroom-performance-by-challenging-student-misconceptions-about-learning [Accessed February 10, 2022].
8. Cao, L. & Nietfeld, J.L., 2006. College students' metacognitive awareness of difficulties in learning the class content does not automatically lead to adjustment of study strategies. Australian Journal of Educational & Developmental Psychology. Available at: https://eric.ed.gov/?id=EJ815620 [Accessed February 10, 2022].