

Enhancing three variants of harmony search algorithm for continuous optimization problems

Alaa A. Alomoush¹, Abdul Rahman A. Alsewari², Kamal Z. Zamli³, Ayat Alrosan⁴, Waleed Alomoush⁵, Khalid Alissa⁶

^{1,2,3}Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia

⁴Deanship of Information and Communication Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

⁵Computer Department, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

⁶Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman bin Faisal University, Dammam, Saudi Arabia

Article Info

Article history:

Received Jul 18, 2020

Revised Dec 15, 2020

Accepted Dec 28, 2020

Keywords:

Evolutionary algorithms

Harmony search algorithm

Hybrid algorithms

Meta-heuristics

Optimization algorithms

ABSTRACT

Meta-heuristic algorithms are well-known optimization methods, for solving real-world optimization problems. Harmony search (HS) is a recognized meta-heuristic algorithm with an efficient exploration process. But the HS has a slow convergence rate, which causes the algorithm to have a weak exploitation process in finding the global optima. Different variants of HS introduced in the literature to enhance the algorithm and fix its problems, but in most cases, the algorithm still has a slow convergence rate. Meanwhile, opposition-based learning (OBL), is an effective technique used to improve the performance of different optimization algorithms, including HS. In this work, we adopted a new improved version of OBL, to improve three variants of Harmony Search, by increasing the convergence rate speed of these variants and improving overall performance. The new OBL version named improved opposition-based learning (IOBL), and it is different from the original OBL by adopting randomness to increase the solution's diversity. To evaluate the hybrid algorithms, we run it on benchmark functions to compare the obtained results with its original versions. The obtained results show that the new hybrid algorithms more efficient compared to the original versions of HS. A convergence rate graph is also used to show the overall performance of the new algorithms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Alaa A. Alomoush

Faculty of Computing,

Universiti Malaysia Pahang

Pekan 26600, Pahang, Malaysia

Email: allaa1030@gmail.com

1. INTRODUCTION

Optimization algorithms were invented to find the fittest element from groups of choices subjected to specific constraints. The use of the optimization algorithm to solve real-world problems started in the early of this century [1, 2]. A well-known sector of optimization algorithms called metaheuristic algorithms, which were able to solve different types of problems in different domains [3-6]. The reasons behind the usage of metaheuristic algorithms in different fields, as they were able to find near-optimal solutions with fast search speed, and its flexibility to suit different types of problems [7-9] which is very important in our modern life specially in software development [10]. Effective metaheuristic algorithms were created and used in the

literature such as simulated annealing [11], particle swarm [12], harmony search algorithm [13], Firefly [14], and cuckoo search [15].

The effectiveness of meta-heuristic algorithms relies on the utilization of exploration (global search) and exploitation (local search) ranges through a search. The explorative process is defined as the ability of an algorithm to investigate uncovered areas rapidly within large search sizes. The exploitative process is achieved by utilizing the information gained to guide the search toward its goal. Overall the algorithm performance will improve if a balance between exploration and exploitation features is achieved [1]. Geem *et al.* [13], created the Harmony Search algorithm (HS) by mimicking the creation of new music tune, and it has been used to solve different types of problems by many researchers in different areas such as engineering [16], computer science [17] and many other fields [18].

Even though the HS has strong exploration, the problem of HS it has weak exploitation. The weak exploitation because it has a slow convergence rate, which means the HS is able to discover solution space using its exploration process, but it has difficulty in term of finding the global optima within this space through the exploitation process. To improve the HS performance and fix this problem, researchers proposed different variants of HS in the literature, by adopting different techniques, such as chaotic map [19], hybrid algorithms [1], and opposition based learning (OBL) [20].

Even though many variants introduced in the literature to improve the overall performance of HS, they continue to suffer from the weak exploitation process, while others improved the convergence rate, but they tend to fall in local optima after removing some of the HS parameters. Overall, most of these variants were unable to provide sufficient results and handle different types of problems. In this work we introduce hybrid algorithms between HS variants and improved version of OBL, to enhance the performance of these variants. OBL is an effective technique created by Tizhoosh [21] to enhance optimization algorithms, and in this work, we adopted an improved version of OBL, which utilizes randomness to create a new possible solution. The improved OBL (IOBL) will be used in the HS update process.

In the following section, we will provide a brief description of the HS and its variants that we are going to hybridize with the IOBL. After that to verify the effectiveness of using the IOBL technique we will apply the proposed hybrid algorithms on 9 standard benchmark functions as characterized in Table 1. As the result shows, the new hybrid algorithms show significant improvement for all the HS variant's performance, as the IOBL increased the convergence speed and enhanced the HS variant's exploitation. The subsequent parts will be organized as follow: part 2 will present the original structure of HS and some of its variants, part 3 will present the proposed hybrid algorithms, part 3 will present the obtained results discussion, and in the final part, the conclusion and future work will be provided.

2. ORIGINAL HS STRUCTURE AND SOME VARIANTS

First, we will describe the standard HS algorithm, and its main components, after that a description of the other variants that we used in this, and how they differ from the standard HS.

2.1. Standard structure of HS as described by its author

The HS simulates the musician process to create a new harmony music tune. HS tunes a new prospective value to achieve global optima, in a similar manner of the tuning process to create a new beautiful tone. The standard HS algorithm has three major phases, described in Figure 1 as pseudocode:

- a. In the first phase, the HS specifies the static parameter values bandwidth (BW), pitch adjustment rate (PAR), harmony memory acceptance rate (HMCR), and harmony memory size (HMS).
- b. In the second phase the algorithm will create a new population randomly inside the HM, using (1):

$$x'_j = LB + R5 \times (UB - LB)$$

- c. In the third phase, the algorithm will improvise the population inside the HM, based on its parameters (BW, PAR, and HMCR). Through this phase the algorithm will have to choices based on HMCR as follows:
 - If ($R1 > HMCR$), a stochastic value will be processed in the next equation ($R1$ is a stochastic value between 0~1):

$$x'_j = LB + rand \times (UB - LB)$$

- If ($R2 < HMCR$), the algorithm will pick a random HM, and if ($R2 \leq PAR$) the value of the chosen HM will be tuned as follows:

$$x'_j = x_j \pm R4 \times bw.$$

- d. In the third phase, the new improvised value x'_j will replace the worst one in the HM, if it has a superior objective function value.
- e. Finally, the improvisation process of the HS algorithm will end once the algorithm reaches a stopping cause such as the highest number of iterations.

Step (2)	1:Initialize HS population: 2: for($i = 1$ to HMS) 3: for($j = 1$ to D) 4: $x_i = \text{lower bound} + R1 \times (\text{upper bound} - \text{lower bound})$ 5: end for 6: end for
Step (3)	7:Improvise new hamrony: 8: while ($t < \text{Max number of iterations}$) 9: for ($j = 1$ to D) {D is number of dimensions} 10: If ($R2 \leq \text{HMCR}$) {Memory consideration} 11: $x'_i = x_{i,j}$ {i is a random integer (1, ..., HMS)} 12: if ($R3 \leq \text{PAR}$) {Pitch adjustment} 13: $x'_j = x'_j \pm R4 \times bw$ 14: end if 15: else 16: $x'_j = \text{LB} + R5 \times (\text{UB} - \text{LB})$ 17: end if 18: end for
Step (4)	19:Update HM: 20: if (x'_j better than worst x_j { $x_j \in \text{HM}$ }) 21: $x_j = x'_j$ 22: $t = t + 1$ 23:
Step (5)	24:End while 25:return best harmony

Figure 1. Harmony search algorithm

2.2. HS variants

The HS algorithm has some advantages such as flexibility and easy to implement, and because of that, many researchers use it to fix several kinds of complex problems. Similar to other metaheuristic algorithms, HS has some weaknesses, such as the weak exploitation process, and its parameter tuning. To the HS performance and its limitations, several HS variants and hybridization approaches have been introduced in the literature.

Hence sometimes these variants and hybridization fall in local optima or still have a slow convergence rate. In this article, we aim to enhance the performance of these variants, by improving their convergence speed. To do that we present a new technique, based on opposition-based learning, to enhance the performance of three recent variants of HS.

In this part, we will describe the three variants that have been enhanced in this work:

- a. The first variant of HS was introduced in 2007 as an improved harmony search (IHS) [22]. The new variant aims to improve the original HS performance by solving its parameter tuning problem, and to do that two parameters (PAR and BW) updated through iterations using specific functions. The new variant provided a decent result compared to standard HS but still has weak exploitation.
- b. The second variant of HS named an exploratory power of the harmony search (EHS) [23], in this work, the authors analyzed the HS and proposed a new variant of HS. The new variants are similar to the original except it has a new BW modification process, which improved the overall performance of the algorithm, but in some cases, the new variant still has a slow convergence rate.
- c. The third variant of HS is called improved global-best harmony search algorithm (IGHS) [24]. The new variant is different from the original HS by focusing on the exploration process at the beginning of the search, and on the exploitation process at the end of a search. In this article, they used standard OBL only in the initialization process. The overall result was better than previous HS variants, but it still has slow convergence in some cases.

The HS variants that have been introduced in the literature show some improvement in the algorithm performance, but they have the same updating process as in Figure 1, step 4, which can be improved by adopting the OBL or other techniques. In this work, we will implement a new improved OBL technique on the aforementioned variants to increase their convergence rate and improve the overall results.

3. PROPOSED ALGORITHMS

To overcome HS weak exploitation, many researchers proposed different variants of HS. The modification covered different parts of the HS, such as initialization, improvisation, or parameter selection. Yet all these variants have the same updating procedure, similar to the original HS. This work proposes new hybrid algorithms of HS variants with a new updating procedure, named improved OBL, to enhance the convergence speed and avoid falling in local optima for three variants of HS, IHS, EHS, and IGHS.

The following section presents a new improved opposition-based learning technique (IOBL), which we aim to use as part of the updating process of the hybrid algorithms. The goal of using IOBL to improve the local search process, of the three described variants. All the new variants will be compared before and after the use of IOBL in the evaluation part.

3.1. IOBL structure

The first OBL was created by Tizhoosh [21], and after that, different variants of it were developed and used in different research areas [24-26]. The original OBL was able to enhance the performance of different optimization algorithms, including HS [27]. The current study presents an improved version of the original OBL by including randomness in the process which enhances the diversity of the solution, to provide better performance than the original OBL for continuous optimization problems. The improved opposition was applied through the HS updating phase to increase HS exploitation as the following Figure 2 present. In Figure 2, x_d is the obtained results from the improvisation process, r stochastic number between (0~2), D reflect the dimensions, and \bar{x} stands for the improved value using OBL.

Improved opposition
$x_d = x_1, x_2, \dots, x_d$ $r = \text{create stochastic number between } (0, 1)$ While ($i < D$) do $\bar{x}_i = x\{i\} \times r;$ if ($\text{objective function}(\bar{x})$ $< \text{objective function}(x)$) $x = \bar{x}$ End if End for

Figure 2. Pseudo code of the improved opposition algorithm

4. RESULTS AND DISCUSSION

To present the performance of the proposed hybrid algorithms, we will compare the HS variants before and after adding the IOBL in the updating part. The evaluation process will be made using 15 benchmark functions to find the global optima. After that, we will compare the variant and its enhanced one based on the convergence rate speed. The HS variants implemented in this work same as they described by their authors, except for two parameters that we use as a fixed value, (HMS=5, and the highest number of evaluation function= 10^4). Table 1 describes the benchmark functions we used in this paper, their optimal values, and the range for each function.

4.1. Comparison results before and after adding IOBL

All the results of three variants and their hybrid version with IOBL are presented in Table 2. In this table, we provide the average, and time consuming to find the global optima for each HS variants, against its hybrid version with the IOBL.

- The column (IHS) in Table 2, presents the results of the first variant, and in the second column, we present the result of Hybrid-IHS. As the table shows: the results of (Hybrid-HS) consume less running time for all test cases. Also, it provides a better result or being closer to global optima for all benchmark

functions except for F5. The reason behind this improvement was the usage of IOBL as it increases the diversity of the proposed solution and increased the convergence rate of IHS. Meanwhile, for F5 the original IHS provided better results, as this function requires more focus on exploration, which is the opposite of the IOBL role.

- b. The column (EHS) in Table 2, presents the results of the second variant and the next column presents the results of the (Hybrid-EHS). The results in Hybrid-EHS shows significant enhancement except for F5, similar to the previous example.
- c. The column (IGHS) in Table 2, presents the results of the third variant and the next column (Hybrid-IGHS) present the results of hybrid IGHS with IOBL, and the obtained results by the hybrid algorithm is better than the original IGHS for all functions, except F7.

According to the Table 2, the hybrid algorithms provided better performance for most cases with lower running time, and the reason is the usage of IOBL, which enhanced the exploitation process of HS variants. The hybrid algorithm of IGHS with IOBL provided the best overall result compared to the other variants and their hybrids.

Table 1. The performance of

Function	Global Optima	Range
F1: Sphere	0	-100, 100
F2: Schwefel's 2.22	0	-10, 10
F3: Step	0	-100, 100
F4: Rosenbrock	0	-30, 30
F5: Schwefel's 2.26	-12569.5	-500, 500
F6: Rastrigin	0	-5.12, 5.12
F7: Ackleys	0	-32, 32
F8: Griewank	0	-600, 600
F9: Rotated hyper-ellipsoid	0	-100, 100

Table 2. Average and running time the best results (for 30 dimensions)

Function		IHS	Hybrid-IHS	EHS	Hybrid-EHS	IGHS	Hybrid-IGHS
F1	Mean	202	1.54E-7	59	1.304E-321	1.52E-136	0.0
	Time	0.568	0.309	0.734	0.622	0.438	0.459
F2	Mean	4.68	5.24E-4	1.1E-4	6.76E-161	1.01E-86	0.0
	Time	0.552	0.380	0.678	0.446	0.291	0.197
F3	Mean	206	0.0	69	0.0	0.0	0.0
	Time	0.586	0.373	0.738	0.391	0.369	0.296
F4	Mean	8807	28.65	977	28.38	35	28.66
	Time	0.504	0.263	0.733	0.345	0.285	0.171
F5	Mean	-12132	-11728.74	-12263	-12238	-11557	-12569
	Time	0.824	0.602	0.974	0.659	0.599	0.494
F6	Mean	17.4	1.77E-5	78	0.0	10	2.20E-7
	Time	0.823	0.580	1.011	0.643	0.579	0.470
F7	Mean	4.64	1.79E-4	3.41	4.44E-16	5.77E-15	8.42E-5
	Time	0.875	0.634	1.034	0.675	0.593	0.500
F8	Mean	2.84	3.69E-9	1.86	0.0	0.0	0.0
	Time	0.827	0.669	1.001	0.708	0.634	0.529
F9	Mean	2568	6.42E-7	551	2.22E-8	1.65E-97	0.0
	Time	0.641	0.397	0.869	0.471	0.426	2.521

4.2. Convergence rate before and after adding IOBL

In this part, A comparison between the HS variants before and after adding the IOBL to their structure, the following graphs present the convergence rate of each variant. All the variants applied on the same objective functions (number 1 and 6), with 100 number of iterations. As figures Figure 3 till Figure 7 presents, the IOBL enhanced the convergence speed for all HS variants. For Figures 3 and 4 we compared the original IHS and its new variant IHS-IOBL, using two objective functions, numbers 1 and 6. The figure shows the convergence rate increase and the algorithm reaches global optima with a smaller number of iterations. After that in Figures 5 and 6, we presented the obtained results of applying EHS and its new variant EHS-IOBL, and as we can see in Figures 5 and 6 the convergence speed highly increased after utilizing the IOBL which shows a high improvement compared to the original EHS. For Figures 7 and 8 we compared the variant IGHS and its new variant IGHS-IOBL, and as we can see in graph number 7 the algorithm performance slightly improves, meanwhile graph number 8 shows the convergence rate highly improved. Overall, these graphs present how much the original variant performance-enhanced after adopting the IOBL, and the overall convergence rate increased for all the variants.

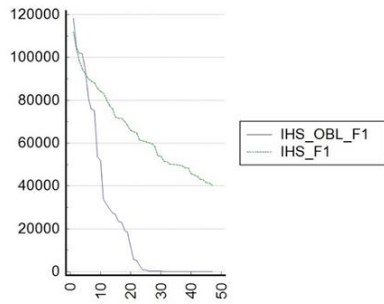


Figure 3. Convergence rate of IHS and IHS-IOBL, F1

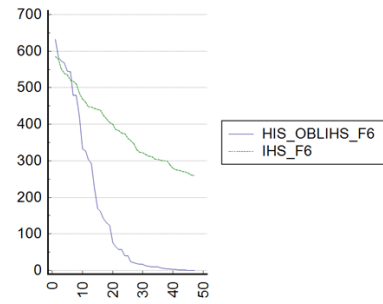


Figure 4. Convergence rate of IHS and IHS-IOBL, F6

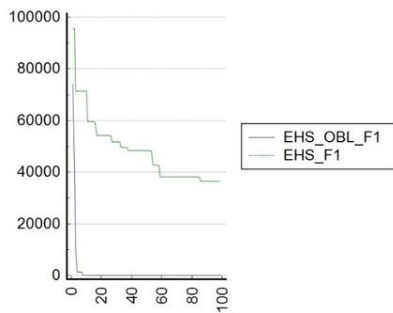


Figure 5. Convergence rate of EHS and EHS-IOBL, F1

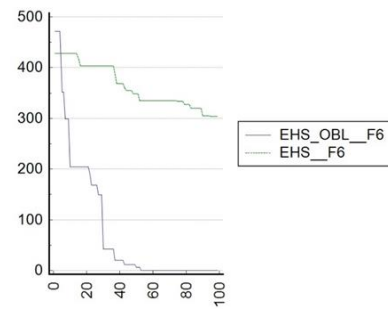


Figure 6. Convergence rate of EHS and EHS-IOBL, F6

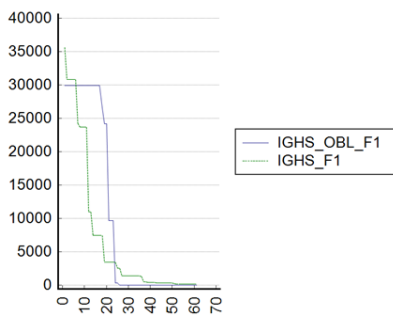


Figure 7. Convergence rate of IGHS and IGHS-IOBL, F1

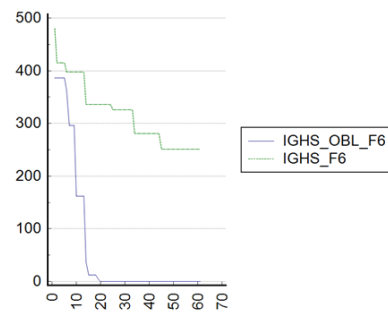


Figure 8. Convergence rate of IGHS and IGHS-IOBL, F6

5. CONCLUSION

HS is a well-known metaheuristic, that has advantages such as simplicity and easy to apply to different problems. But similar to other metaheuristics it has weaknesses such as a slow convergence rate, which causes the algorithm to have a weak exploitation process. Many variants introduced in the literature to address the HS problems, and they have enhanced the algorithm performance, yet most of these variants of HS still have insufficient convergence rate. In this work, we have implemented an improved opposition-based learning technique in the updating phase of the HS recent variants, to enhance the overall algorithm performance, by improving the exploitation process. The proposed hybrid algorithms evaluated, against its original one, using 9 benchmark functions. Moreover, a convergence rate analysis was conducted to present the algorithm enhancement using the IOBL. The hybrid HS variants provided a better result than the original HS variant, with higher convergence speed and lower running time. Overall the IGHS variant with IOBL shows the highest-obtained results in the evaluation test compared to the others. For future work, the enhanced variants can be used to solve a real-world optimization problem. Also, the IOBL technique can be used to enhance other metaheuristics to increase the convergence rate and improve the overall performance.

ACKNOWLEDGEMENTS

The research supported by Universiti Malaysia Pahang (UMP) under the grant scheme number UMP (RDU190334): A Novel Hybrid Harmony Search Algorithm with Nomadic People Optimizer Algorithm for Global Optimization and Feature Selection, and (Ref: PRGS/1/2019/ICT02/UMP/02/1) IoT based Intelligent Combinatorial Test Cases Generator System Based on Kidney Inspired Algorithm with Opposition Approach.

REFERENCES

- [1] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Aloufi, and K. Z. Zamli, "Hybrid harmony search algorithm with grey wolf optimizer and modified opposition-based learning," *IEEE Access*, vol. 7, pp. 68764-68785, 2019.
- [2] K. Alomari, O. Almarshdi, A. Marashdh, and B. Zaqaibeh, "A New Optimization on Harmony Search Algorithm for Exam Timetabling System," *Journal of Information & Knowledge Management*, vol. 19, no. 1, 2020.
- [3] M. Alauthman, A. Almomani, M. Alweshah, W. Omoushd, and K. Alieyane, "Machine Learning for phishing Detection and Mitigation," *Machine Learning for Computer and Cyber Security*, vol. 26, pp. 48-74, 2019.
- [4] W. Alomoush, S. N. H. S. Abdullah, S. Sahran, and R. I. Hussain, "MRI Brain Segmentation via Hybrid Firefly Search Algorithm," *Journal of Theoretical & Applied Information Technology*, vol. 61, no. 1, pp. 73-90, 2014.
- [5] W. Alomoush and K. Omar, "Dynamic fuzzy C-mean based firefly photinus search algorithm for MRI brain tumor image segmentation," PhD, Computer science, Universiti Kebangsaan Malaysia, Malaysia, 2015.
- [6] W. Alomoush and A. Alrosan, "Metaheuristic Search-Based Fuzzy Clustering Algorithms," arXiv preprint *arXiv:1802.08729*, 2018.
- [7] E. Alhroob and N. Ab Ghani, "Fuzzy min-max classifier based on new membership function for pattern classification: A conceptual solution," *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, 2018, pp. 131-135.
- [8] A. Alsewari, R. Poston, K. Zamli, et al., "Combinatorial test list generation based on Harmony Search Algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-17, 2020.
- [9] Alomoush, Alaa A., et al., "Pressure Vessel Design Simulation Using Hybrid Harmony Search Algorithm," *Proceedings of the 2019 3rd International Conference on Big Data Research*, 2019, pp. 37-41.
- [10] H. Fadhl, R. B. A. Bakar, and M. A. Abdulgaber, "Investigation of requirements interdependencies in existing techniques of requirements prioritization" *Tehnički vjesnik*, vol. 26, no. 4, pp. 1186-1190, 2019.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.
- [13] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *simulation*, vol. 76, pp. 60-68, 2001.
- [14] X. -S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International journal of bio-inspired computation*, vol. 2, no. 2, pp. 78-84, 2010.
- [15] Yang, Xin-She, and Suash Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330-343, 2010.
- [16] Alomoush, Alaa A., et al., "Pressure Vessel Design Simulation Using Hybrid Harmony Search Algorithm," *ICBDR 2019: Proceedings of the 2019 3rd International Conference on Big Data Research*, 2019, pp. 37-41.
- [17] Alsewari, Abdul Rahman A., et al., "Software Product Line Test List Generation based on Harmony Search Algorithm with Constraints Support," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 1, pp. 605-610, 2019.
- [18] A. Ala'a, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Comprehensive review of the development of the harmony search algorithm and its applications," *IEEE Access*, vol. 7, pp. 14233-14245, 2019.
- [19] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, pp. 2687-2699, 2010.
- [20] Shiva, Chandan Kumar, and Ritesh Kumar, "Quasi-oppositional Harmony Search Algorithm Approach for Ad Hoc and Sensor Networks," *Nature Inspired Computing for Wireless Sensor Networks*, 2020, pp. 175-194.
- [21] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vienna, 2005, pp. 695-701.
- [22] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied mathematics and computation*, vol. 188, no. 2, pp. 1567-1579, 2007.
- [23] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B.K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 89-106, 2010.
- [24] Xiang, Wan-li, et al., "An improved global-best harmony search algorithm for faster optimization," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5788-5803, 2014.
- [25] X. Gao, X. Wang, S. Ovaska, and K. Zenger, "A hybrid optimization method of harmony search and opposition-based learning," *Engineering Optimization*, vol. 44, no. 8, pp. 895-914, 2012.
- [26] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao, "A review of opposition-based learning from 2005 to 2012," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 1-12, 2014.
- [27] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning: A literature review," *Swarm and evolutionary computation*, vol. 39, pp. 1-23, 2018.