Albahit Journal of Applied Sciences ISSN 2708-244X Vol. 1, Issue 1, 2020, 38-43

Global Optimization of SMHS to Enhance The Performance of Diversity and Sensitivity via Comparing with HS and Other Existing Methods

Khamiss M. S. Ahmed[1], *Mahmmoud Alawan*[2], *Fatima Mustafa*[3]
[1, 2, 3] *Department of Computer Science, Faculty of Information Technology, Sebha University*
Sebha, Libya
km.ahmed@sebhau.edu.ly[1], m.lawan@sebhau.edu.ly[2], fatimasaleem@gmail.com[3]

Abstract—**The optimization process is concerned with the finding of the best solution from all possible solutions for a given problem. In this paper, the focus is on the importation step of HS algorithm using Scramble Mutation. The essential idea of new type of modification on variable. The selection method of Scramble mutation hybridized onto harmony search. The resulting variant of HS is called the Scramble Mutation Harmony Search algorithm (SMHS). For optimization problems to avoid premature convergence problem due by using one of the mutations to increase the number of solutions proposed, thus improving the performance of the Harmony Search (HS) algorithm, by maintaining diversity in the use of different rates to modify the proposed solutions. Compare the SMHS with HS and other existing methods to validate the efficiency of the SMHS. The results obtained by comparing the SMHS with basic HS and two other methods (i.e. MHS and DLHS), using ten benchmark functions illustrated that SMHS outperformed the basic HS for majority of the functions. It is therefore concluded that the SMHS algorithms is highly sensitive for the HMCR and obtains the best results at high value of HMCR. For harmony memory size, the HS performs better when HMS is relatively small. In the same vein, the SMHS performs better when the number of domain ( ) is compatible with HMS.**

Keywords- Optimizations, Evolutionary Algorithm (EA), Harmony Search (HS), Genetic Scramble Mutation Harmony Search (SMHS).

I. INTRODUCTION

Evolutionary Algorithm (EA) has widely been used by many researchers in different disciplines. EA has been implemented successfully on different optimization problems in different field such as: engineering problems, medical problems, economy problems, intelligent community problems, and many other. EA is an optimization technique that aims to employ survival of fittest theory in the biological context for the optimization problem to come up with a global optimum. The types of evolutionary algorithms are: Evolutionary Programming (EP) [1]; Evolutionary Strategies (ES) [2]; Genetic Algorithms (GA) [3]; and Genetic Programming (GP) [4].

These methods start with a random population as a set of individuals (chromosomes). It improves upon the individual chromosomes using a general method to solve several problems, no mathematical calculation is needed. It can provide a fairly 1 optimal solution in reasonable amount of time for the complex optimization problems [5]. The size and sitting of population is very important to ensure balance between exploration and exploitation, which hamper the findings of global optimal solutions. The exploration is to find new solutions in search space which have not been visited before. The exploitation is to improve on the existing study, and combine the traits of the current solutions.

The Harmony Search algorithm (HS) is one of the recent evolutionary algorithms used for solving optimization problems [6]. HS imitates the musical improvisation process of music players, which tries to seek a pleasing harmony as determined by an audio-aesthetic standard. The solution vector of an optimization problem is similar to harmonies of music [7].

Genetic algorithms (GA) are adaptive algorithms that are based on the evolutionary ideas of natural and genetic selection [8]. The basic concept of genetic algorithms is designed to simulate processes in the natural system necessary for evolution [9].

The structure of the GA Starting with an initial group of population (which may be randomly generated or implanted by other heuristics), select parents from this population are identified for mating. Apply the crossover and mutation to generate parents to create new off-springs [10]. Finally, these springs replace with individuals in the population and repeat this process. In this way, GA actually tries to mimic human evolution to some extent.

Mutation is one of the processes of the genetic algorithm that changes the structure of the chromosome, through which a change is made at the level of the Gen, where I suggest many types of mutations in this research market is addressed to the (Scramble Mutation) [11], It his idea that a subset of the genes in the chromosome is randomly selected and then reordered in those positions randomly. Subgroup should not be contiguous.

This paper focuses on improving the performance of basic HS proposed by [2]. To overcome the constraints imposed by premature convergence and the local minimum. The Scramble mutation method is integrated into the HS algorithm to increase diversity by reorganizing Vectors in a given domain.

This study will investigate whether the proposed SMHS algorithm is better than the basic HS algorithm Standard mathematical functions are often used evaluate the proposed method SMHS [12].

This study aims to compare the SMHS with HS and other existing methods to validate the efficiency of the SMHS.

II. GENETIC ALGORITHM

Genetic algorithms (GA) are adaptive algorithms that are based on the evolutionary ideas of natural and genetic selection [8]. The basic concept of genetic algorithms is designed to simulate processes in the natural system necessary for evolution. The basic concept of genetic algorithms is designed to simulate processes in the natural system necessary for evolution. The structure of the GA is as follows:

Starting with an initial group of population (which may be randomly generated or implanted by other heuristics), select parents from this population are identified for mating. Apply the crossover and mutation to generate parents to create new off-springs. Finally, these springs replace with individuals in the population and repeat this process. In this way, GA actually try to mimic human evolution to some extent [9]. Generally, Genetic algorithm has 5 steps as follows:

A. *Population Initialization*

The process begins with a group of individuals called "population". Everyone is a solution to the problem you want to solve. The individual is characterized by a set of parameters (variables) known as genes. The genes in the chain are bound to form a chromosome (solution). In the GA the gene array of an individual is represented by a string, in terms of the alphabet. Typically, (string of 1s and 0s) is used for binary values that we say we encodes genes in the chromosome," representation of the population depends on the problems".

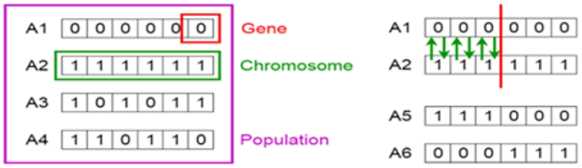


Figure .1 The basic structure of a GA

B. *Fitness function*

The efficiency function determines the individual's ability to compete with other individuals. It gives a degree of efficiency for each individual. The probability of choosing an individual to reproduce depends on the degree of fitness.

C. *Selection*

The idea of the selection stage is to select the most appropriate people and allow them to pass their genes to the next generation. A pair of individuals (parents) is selected based on their degree of competence. Individuals with high fitness have a greater chance of being chosen for breeding".

D. *Crossover*

The crossover is the most important stage in the genetic algorithm. Each pair of parents is mated, the intersection point is chosen randomly from within the gene. The offspring are created through the exchange of genes between the parents until the intersection is reached.

E. *Mutation*

In some new offspring formed, it can be exposed to some of its genes with a low probability of random mutation. This means that some bits in the bit string can be reversed) [13]. Mutation is one of the processes of the genetic algorithm that changes the structure of the chromosome, through which a change is made at the level of the Gen, where I suggest many types of mutations in this research market is addressed to the (Scramble Mutation) [11], It his idea that a subset of the genes in the chromosome is randomly selected and then reordered in those positions randomly. Subgroup should not be contiguous.

III. THE HARMONY SEARCH ALGORITHM

The Harmony Search HS algorithm has several characteristics that make it an important and powerful meta-heuristic algorithm [6]. It differs from other meta-heuristic algorithms:

a) In contrast to GAs, which consider only two vectors (two parents), the HS algorithm considers all existing vectors to generate a new vector.

b) Each vector variable is independently considered;

c) It does not require a fixed number of decision variables [14].

Generally, basic HS has 5 steps as follows:

1. Initialize the problem and HS parameters: The parameters of HS in this step are: the harmony memory size (HMS), Harmony Memory Consideration rate (HMCR), Pitch Adjusting Rate (PAR), and number of iteration (NI);
2. Initialize the Harmony Memory (HM): The initial harmony memory is a matrix that contains generated solution vectors up to Harmony Memory Size (HMS);
3. Improvise a new harmony: Improvising a new harmony. This generation follows specific operators known as: Harmony memory consideration, pitch adjustment and random selection;
4. Update harmony memory: Updating harmony memory, knowing that New Harmony is better than the worst vector in (HM); and
5. Check the stop criterion: Terminate the number (NI) of improvisation, when the maximum number is reached.

A. *Hybridized version of Harmony Search*

Pan Introduce a new population structure using dynamic sub-populations called (DLHS). In the proposed method, the HM is divided into many small-sized sub-HM, which are regrouped through regrouping schedule, and the harmony search is performed in each sub-HM independently [12].

Geem Proposed harmony search algorithm based on mutation to determine Web services and prepare them with minimal imperfections. In this algorithm, some mutations (insert, swap, reverse) have been used with HS .The results of this study reveal that MBHS can support a range of different services more efficiently than other meta-heuristic methodologies. In addition, it helps in finding appropriate solutions to create services based on work plans [14].

[15] Proposed a hybrid Harmony Search algorithm by combining the bee colony algorithm (ABC) with the Harmonic Search algorithm (HS), ABC and its variants are used to improve Harmony Memory (HM). The numerical results of the hybrid algorithm show that better solutions can be obtained when compared to HS and other heuristic algorithms for many global optimization problems.

B. *Scramble mutation harmony search SMHS*

Scramble mutation is used with permutation encoded chromosome in this mutation, one has to select a subset of the genes at random and then rearrange the alleles randomly in those positions [16]. A subset should not be contiguous.

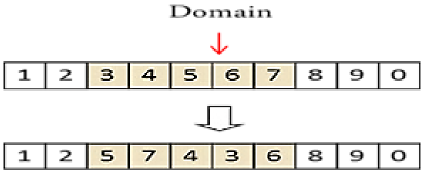


Figure .2 Scramble mutations

If a mutation is applied to an encoded chromosome, a random location is selected and the bits next to that position are reversed and a child chromosome is produced [17].

The Mutation factor does not change the value of the selected gene with a uniform random value determined between the specified user and the upper limit of the gene. It is used for real representation and correct integers.

In a crawling boom, a randomized gene is identified and its value is changed by a random value between the minimum and the maximum. It is used in real representation mode.

Inversion mutation is used for chromosomes with permutation encoding. In order to perform inversion, pick two alleles at random and then invert the substring between them. It preserves most adjacency information and only breaks two links but it leads to the disruption of order information [18].

IV. METHODOLOGY OF SCRAMBLE MUTATION HARMONY SEARCH SMHS

The proposed SMHS has procedures for optimization problems, whose idea is applied in terms of adjacent values only in this work. After improvisation, the basic HS algorithm begins with a step-by-step hybrid of the genetic algorithm known as mutation with basic HS, and a type of mutant called Scramble Mutation is used. The HM is modified to replace the worst solution with the new solution. The proposed HS method is evaluated using a set of global optimization functions. Finally, a comparative analysis is performed with other methods. Fig. 3 illustrates the search schema, SMHS.

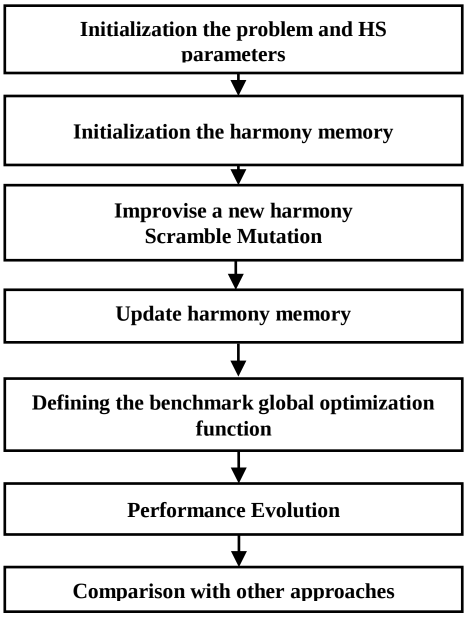


Figure .3 Proposed Research Method (SMHS)

Scramble mutation provides a specific mechanism for HS to improve its performance in terms of maintaining diversity through the iterations. In SMHS, a specific range of the solution is specified at each frequency. The next flowchart Fig. 2 shows the procedures of Scramble Mutation harmony search algorithm (SMHS).The detailed steps of SMHS is discussed in the next subsection:

A. *Initialize SMHS parameters and optimization problem*

It is clear that the successful search of any meta-heuristic method is based on the skillful parameter setting. The parameters have different effect on optimization solutions. The parameters of SMHS are: Harmony Memory Size (HMS), Harmony Memory Consideration Rate (HMCR), Pitch Adjusting Rate (PAR), and number of improvisations (NI) and the range of values (Dom) calculated according to the number of decision variables in a single solution. Table (1) shows the list of parameters for the proposed method (SMHS).

TABLE I. PARAMETERS USED OF SCRAMBLE MUTATION HARMONY SEARCH ALGORITHMS

Parameter	Description
HMS	The size of the harmony memory
HMCR	The probability of choose the on-value form decision variable from HM
Bw	The distance bandwidth PAR The rate of selected candid
NI	The maximum number of improvisations
DOM	The domain of values that convey Scramble mutation operation carried out
f(x)	The objective function for solution x
x´	Decision variable in the new solution

B. *Initialize harmony memory*

The harmony memory is an augmented matrix where the all solutions are stored, the initialize harmony memory with random generation of solution vectors is stored in HM matrix as length as harmony memory size.

C. *Generate a new solution*

In this step generate a new harmony vector (new solution vector) x´ = ((x´)_1,(x´)_(2)⋯⋯⋯,(x´)_N is generated (called improvisation) is based on three operators:(1) Scramble Mutation process, (2)Pitch Adjustment, (3)Random Selection.

D. *Scramble Mutation*

It's process by random selection from HM, the begins and its mechanism is as follows:

1. A certain range of values is determined based on the number of variables specified previously, i.e.: (DOM = NVAR / 2).
2. Choose two values randomly from NVAR provided that the range between them is equal to DOM.
3. The values in the range are randomly switched. This process is repeated by DOM +1 of the frequencies.

E. *Pitch Adjustment*

If decision variable ´x_i is selected from the HM, the pitch adjusting ads certain value with probability of PAR to selected decision variable is P (0, 1)

´x_i= ´x_i±U(−1,1) × bw (1)

Where bw is an arbitrary distance bandwidth, r is generating random number between (-1, 1).

F. *Random Consideration*

The random section is generating random number according the rang of the decision variable with a probability (1-HMCR).

G. *Update the harmony memory*

Replace the worst harmony vector within the set of S the New Harmony vector, if better.

H. *Check stopping criterion*

The In this step, Stop if the maximum number of the improvisation is reached, otherwise the algorithm repeats Step 4 and 5.

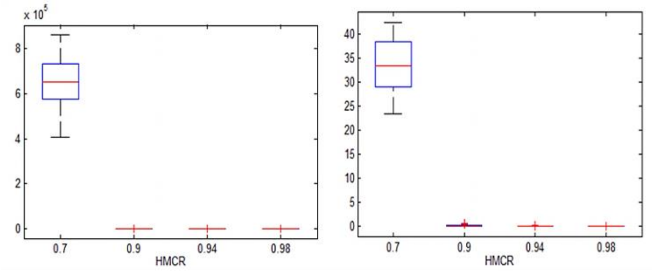
In paper investigate whether the proposed SMHS algorithm is better than the basic HS algorithm. Where integrated the Scramble mutation method into the HS algorithm to increase diversity by reorganizing Vectors in a given domain.

V. RESULT

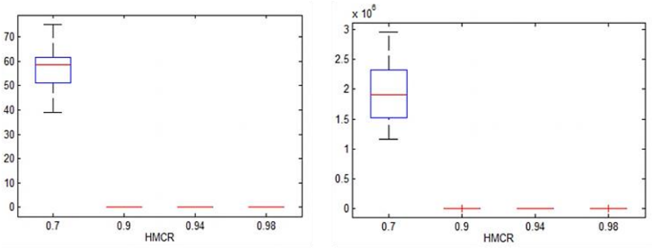
In this paper, the proposed SMHS algorithm is experimentally evaluated by comparing it with other state-of-art techniques using the same benchmark functions. The common parameters among all algorithms used in the experiments are set based on empirical guidelines [6]. The parameters setting used for evaluating the SMHS method are as follows: HMS=25, HMCR=0.98, N=30, DOM=15, NI=5 × (10) ^ 4. All the experiments are run using a computer with Intel(R) Core (TM) i3-3110 with 4GB of RAM. The operating system used is Microsoft windows 7 Service Pack1. The source code is implemented using MATLAB Version 7.14.0.739 (R2012a).

It is revealed that both SMHS and the basic HS are not sensitive to the harmony memory size in different way. Where, SMHS obtained the best results for big HMS as shows in Fig. 4. All cases have effect on result proportion to the size of the Domain (DOM). Thus, to obtain the good solution in functions abovementioned HMS needs small Domain (i.e. HMS=16 need to DOM =8 or bigger). From point of view, SMHS performs better for most of the functions because the overlap of the Domain provides an implicit mechanism of migration to the SMHS. Since the best solutions spread smoothly through the whole population, the SMHS diversity in the population is preserved longer than in HS.

HM is analogous to the short-term memory of a musician that is known to be small. A plausible interpretation may rely on the high number of similar harmonies within the HM when the HMS. That leads to shortages of diversity and, hence, leads to falling into local minima. Since SMHS is likely to be capable of maintaining more ability than HS with the cellular structure.



Griewank function.



Rastrigin function **RosenBrock function**

Figure .4 Scramble mutations

All functions are implemented in 30 dimensions (30D). For the scalability study, the functions are implemented in 10 dimensions (10D). The stopping criteria used is the maximum number of improvisation (NI) and it is set to 5×(10) ^ 4 for 10D.

In order the reveal the impact of control parameter settings and compare the SMHS to HS, a study of different parameter settings of the common parameters (HMS, HMCR is conducted for SMHS and HS. Table (2) shows the results of comparing the proposed method for ten functions (f1 − f10) with thirty-dimensional (N=30) with thirty-dimensional (N=30). The results of MHS and DLHS are obtained from [12]. The best results highlighted in bold font.

TABLE II. COMPARISON RESULTS BETWEEN HS, MHS, DLHS AND SMHS

Function	**HS**	**MHS**	**DLHS**	**SMHS**
f1 : Sphere	1.035E-04 (2.018E-05)	2.762E-05 (8.937E-06)	**1.299E-09 (2.766E-09)**	5.645E-05 (1.284E-05)
f2: Schwefel's problem 2.22	2.336E-02 (3.177E-03)	1.258E-02 (1.886E-03)	**1.234E-04 (2.268E-04)**	1.808E-02 (5.803E-03)
f3 : Step	9.793E-05 (2.439E-05)	3.000E-01 (5.349E-01)	1.333E+00 (2.771E+00)	**5.568E-05 (9.632E-06)**
f4: Rosenbrock	7.175E+01 (4.273E+01)	1.319E+02 (1.642E+02)	2.283E+02 (2.508E+02)	**6.017E+01 (2.289E+02)**
f5: Rotated hyper-ellipsiod	2.724E-01 (2.596E-01)	5.483E+03 (2.205E+03)	9.029E+02 (4.663E+02)	**4.877E-02 (6.772E-02)**
f6: Schwefel's problem 2.26	1.257E+04 (1.070E+00)	1.342E-02 (6.963E-05)	**6.785E-03 (6.907E-03)**	1.257E+04 (1.531E-06)
f7 : Ratrigin	1.498E-02 (3.026E-03)	7.157E-02 (2.520E-01)	1.863E+00 (1.340E+00)	**1.076E-02 (1.413E-03)**
f8 : Ackely's	7.03E-03 (6.994E-04)	6.025E-03 (1.947E-03)	1.910E+00 (6.838E-01)	**5.308E-03 (4.190E-04)**
f9 : Griewank	6.645E-01 (1.916E-01)	1.001E+00 (3.006E-05)	1.000E+00 (1.166E-06)	**2.058E-03 (5.611E-03)**
f10: Six-Hump-Camel Back	**1.032E+00 (2.183E-11)**	5.774E-08 (1.35408)	4.651E-08 (1.531E-13)	1.032E+00 (2.422e-11)

The results in Table (2) show the superior performance of SMHS over other benchmark optimization functions.

Many variations of HS are presented in the literature [20]. To make the comparison fair, two improved variations of HS in addition to the classic HS algorithm are selected. The methods used to compare the proposed variant of HS are summarized in Table (3) as follows:

TABLE III. THE KEY FOR COMPARATIVE STUDY

Method	**Denotation**	**Reference**
New heuristic optimization algorithm harmony search	**HS**	(Geem, et al., 2001)
An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis	**MHS**	(Cheng, et al., 2008)
A local-best harmony search algorithm with dynamic sub-populations	**DLHS**	(Pan, et al., 2010)

The results obtained by comparing the SMHS with basic HS and two other methods (i.e. MHS and DLHS), using ten benchmark functions illustrated that SMHS outperformed the basic HS for majority of the functions. The multi-modal functions outperformed the other two approaches.

VI. DISCUSSION

The resulting variant of HS is called the Scramble Mutation Harmony Search algorithm (SMHS). The harmony search is started by initializing the harmony memory, in select process according to Scramble Mutation, where select variable during specific domain. In the third step of harmony search, the improvised New Harmony is determined by using the three operations (Memory Consideration, Pitch Adjustment and Random Consideration). It is therefore concluded that the SMHS algorithms is highly sensitive for the HMCR and obtains the best results at high value of HMCR. For harmony memory size, the HS performs better when HMS is relatively small. In the same vein, the SMHS performs better when the number of domain (DOM) is compatible with HMS.

VII. CONCLUSION

In conclusion, the study, the focus is on the importation step of HS algorithm using Scramble Mutation. Whereas, the main objective of this study was to enhance the performance of the basic harmony search algorithm in terms of improving the diversification aspect and thus to avoid premature convergence using the Scramble Mutation(SM) structures. SM introduced a new mechanism within the improvisation step in HS, through which the best solutions were obtained by maintaining diversity and reducing the speed of convergence. The results SMHS were compared to the results obtained three variants of HS, the SMHS obtained better results for the most of optimization functions.

VIII. FUTURE WORK

1. Investigate the effect of the parameter sittings for Scramble Mutation, where use big domain (DOM) (i.e.20, C25, C30) with NVAR>30.
2. Investigate the performance of the proposed method further by exploiting various combinatorial problems to resolve harmony problems, such Timetabling.

REFERENCES

[1] L. Fogel, "Evolutionary programming in perspective," The top-down view, Computational Intelligence, 1994, pp. 135-146.

[2] T. Dack, F. Hoffmeister, and H. Schwefel, "A survey of evolution strategies," s.l.:s.n., 1991.

[3] J. Holland, "Adaptation in natural and artificial systems," S.L.:Michigan Press, 1975.

[4] E. Burke, and G. Kendall, "Search methodologies. introductory tutorials in optimization and decision support techniques," Springer Verlag, 2005.

[5] L. Eshelman, and J. Schaffer, "Preventing premature convergence in genetic." Proceedings of 4th International Conference, pp. 115-122, 1991.

[6] Z. W. Geem, J-H Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," Simulation, vol. 76, pp. 60–68, 2001.

[7] K. Lee, Geem, and Bae, "The harmony search heuristic algorithm for discrete," Engineering Optimization, vol. 37, pp. 663–684, 2005.

[8] S. N. Sivanandam, and S. N. Deepa, "Introduction to genetic algorithms," Springer Science & Business Media, 2007.

[9] V. Mallawaarachchi, "Introduction to Genetic Algorithms — Including Example Code 2017," 2019 [Online]. Available at: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3 [Accessed 18 7 2019].

[10] A. Hussain, and Y. Muhammad, "Optimization through Genetic Algorithm with a new and efficient crossover operator," International Journal of Advances in Mathematics, vol. 1, pp. 1-14, 2018.

[11] N. Soni, and T. Kumar, "Study of various mutation operators in Genetic Algorithms." International Journal of Computer Science and Information Technologies, vol. 5, pp. 4519-4521, 2014.

[12] Q. Pan, P. Suganthan, J. Liang, and M. Tasgetiren, "A local best harmony search algorithm with dynamic subpopulations," Engineering Optimization, vol. 42, pp. 101–117, 2010.

[13] E. Maythaisong, and W. Songpan, "Mutation-Based Harmony Search Algorithm for Hybrid Testing of Web Service Composition," Hindawi Limited, Computational Intelligence and Neuroscience, Article ID 4759405, 2018.

[14] Z. W. Geem, and Y. Yoon, "Harmony search optimization of renewable energy charging with energy." Electrical Power and Energy Systems, vol. 86, pp. 120–126, 2017.

[15] W. Bin, Q. Cunhua, N. Weihong, and F. Shuhai, "Hybrid harmony search and artificial bee colony algorithm for global optimization problems," Computers & Mathematics with Applications, vol. 64, pp. 2621-2634, 2012.

[16] S. S. Reddy, "Optimal power flow using hybrid differential evolution and harmony search algorithm," International Journal of Machine Learning and Cybernetics, vol. 10, pp 1077–1091. 2019.

[17] R. Mikaeil, Y. Ozcelik, M. Ataeiorcid, and S. H. Shaffiee, "Application of harmony search algorithm to evaluate performance of diamond wire saw," Journal of Mining and Environment (JME), vol. 10, pp. 27-36, 2019.

[18] Y. Jin, L. Chao, and L. Guomin, "A literature review on latest developments of Harmony Search and its applications to intelligent manufacturing," Mathematical Biosciences and Engineering, vol. 16, pp. 2086-2117, 2019.

[19] Y. Cheng, L. Li, T. Lansivaara, and S. Chi, "An improved harmony search minimization algorithm using different slip surface generation methods forslope stability analysis." Engineering Optimization, vol. 40, pp. 95-115, 2008.

[20] Z. Geem, "State-of-the-art in the structure of harmony search algorithm," Recent Advances In Harmony Search Algorithm, Studies in Computational Intelligence, Springer-Verlag Berlin Heidelberg, vol. 270, pp. 1-10, 2010.