ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017
JOURNAL OF SCIENCES AND TECHNOLOGIES, VOL.(7), No.(2) Dec., 2023

PERFORMANCE EVALUATION OF THE HARMONY SEARCH ALGORITHM USING ADAPTIVE HARMONY MEMORY CONSIDERATION RATE APPROACH

Jalal Alhoussin Misbah[1], Amer M. Daeri[2]
Enjila Higher Institute of Industrial Technology
Tripoli - Libya
jalalhiit@yahoo.com

ABSTRACT

There are many techniques that are employed for optimization in the literature. Each of these algorithms has its advantages and disadvantages as well as the suitable environment where it can be applied depending on a set of requirements and constraints. In this study, we explore the effect of using a dynamically changed Harmony Memory Consideration Rate (HMCR) on the performance of the Harmony search algorithm (HS). The objective is to see if the utilization of a dynamic value of the HMCR can improve the optimization results compared to using a fixed HMCR value. We analyze the performance of the HS algorithm with both dynamic and fixed HMCR values by implementing two scenarios in MATLAB simulation. The comparison of the results obtained from the two scenarios revealed that the performance of the HS algorithm when using a dynamically adjusted HMCR value outperformed the performance when using a fixed HMCR value. These results contribute to the area of the optimization algorithms by emphasizing the importance of considering using adapted HMCR to enhance the efficiency of the algorithm.

Keywords: HMCR, HS, optimization, algorithm, performance

المستخلص

هناك العديد من تقنيات التحسين المستخدمة اليوم. كل من هذه التقنيات والخوارزميات لها ميزاتها وعيوبها الخاصة بها، وكذلك البيئة المناسبة لاستخدامها من حيث المتطلبات والقيودات. في هذه الورقة سنستكشف تاثير استخدام قيم متغيرة للمعامل المسمى HMCR و هو احد معاولات خوارزمية البحث المتناغمة على اداء هذه الخوارزمية. الهدف من هذه الدراسة هو معرفة ما اذا كان تبني استخدام قيم متغيرة لهذا المعامل سيؤدي الى التحسين من اداء هذه الخوارزمية وذلك بالمقارنة مع استخدام قيم ثابتة بدلا من ذلك. سيتم تحليل اداء خوارزمية التحسين في الحالتين عند استخدام قيم تتغير باستمرار وقيم ثابتة وذلك ببناء وتشغيل الخوارزمية في برنامج المحاكاة ماتلاب (MATLAB). عند مقارنة النتائج المتحصل عليها من عملية تشغيل الخوارزمية سابقة الذكر تبين ان تشغيل الخوارزمية باستخدام قيم متعددة للمعامل المذكور زاد من اذائها وان جودة الخوارزمية قد تحسنت وتفوقت على ادائها في الحالة الاخرى اي عند استخدام قيم ثابتة للمعامل المذكور. النتائج المتحصل عليها ساهمت في تعزيز والتاكيد على اهمية تبني واستخدام قيم متعيرة للمعامل HMCR لتحسين اداء الخوارزمية.

1. INTRODUCTION

Optimization involves searching for and identifying the optimal values of the design variables within a given search range while satisfying specific constraints. The objective is to maximize fitness and/or minimize cost. The success of the optimization process heavily depends on the values chosen for the decision variables [1]. Selecting

[1] **Jalal Alhoussin Misbah:** Computer Department, higher institute of industrial technology-Enjila, Tripoli – Libya - jalalhiit@yahoo.com
[2] **Amer M. Daeri (IEEE S.M.):** Computer Department - Engineering Faculty - Zawia University, Zawia - Libya amer.daeri@zu.edu.ly

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

the best possible solutions is critical task as it significantly impacts the performance and outcomes of the optimization process. To accomplish this, optimization algorithms have been developed. There are various optimization algorithms available today, including meta-heuristic (stochastic) algorithms, which are widely recognized. These algorithms have been extensively applied to solve diverse optimization problems. Unlike deterministic optimization algorithms, meta-heuristic algorithms don't rely on problem-specific information and type [2]. They employ a random search mechanism to generate new potential solutions within the search space. Numerous meta-heuristic algorithms have been developed to date, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA), all of which are well-known stochastic algorithms. Among these algorithms is the Harmony Search (HS) algorithm. It is a meta-heuristic algorithm introduced by Zong Woo Geem and his colleagues in 2001. It is inspired by the improvisation process of musicians who make slight modifications to their instrument's notes to achieve better harmony. This process continues until the best harmony is attained among all instruments [3]. The Harmony Search algorithm incorporates four main parameters: Harmony Memory (HM), Harmony Memory Consideration Rate (HMCR), Pitch Adjustment Rate (PAR), and Bandwidth (BW) [1]. The Harmony Memory is a matrix that stores the best possible solutions. Initially, it is randomly populated from the allowed search range. The Harmony Memory has an MxN size, where M represents the Harmony Memory size (number of rows), and N represents the number of design variables involved in the objective function. The HMCR plays a crucial role in the search stage, it determines whether the algorithm obtains the new sets of design variables from the stored values in the Harmony Memory or randomly from the search space. If the algorithm retrieves new values from the Harmony Memory, these values may undergo further modification based on the PAR. If the Harmony Search decides to adjust the newly obtained values, it does so by adding the BW; the updated values are given by $x_{i\,new} = x_i \pm$ Random (BW). This step is repeated for each of the N design variables in every iteration. The fitness of the selected values is assessed by applying them to the objective function. If the fitness of the newly generated value surpasses the fitness of the worst values stored in the memory, the Harmony Search replaces the old values with the newly obtained ones. Otherwise, the Harmony Search continues searching for a new solution using the same process. This iterative process persists until the stopping criteria are met [3, 4]. At the end of the search process, all potential values are stored in the Harmony Memory of the Harmony Search Algorithm (HS). The rest of this paper considers related work in section II. Section III discusses the Harmony Search Algorithm (HS). The proposed Modified Harmony Search algorithm is explained in IV, where the methodology being pursued to apply this algorithm is explained in section V. The results and discussion as well as conclusion are dealt with in sections VI and VII respectively.

2. RELATED WORK

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

The Harmony Search (HS), is a well-known meta-heuristic optimization algorithm. It has gained popularity since its introduction in 2001 and has been widely applied to solve various real-life optimization problems. Its advantages over other optimization algorithms have contributed to its widespread adoption. In the pursuit of enhancing the performance of the HS algorithm, scientists and researchers have made continuous efforts and explored different approaches to achieve better results. One such improvement proposed by [5] involves dynamic updates to the values of key parameters in the HS algorithm. Unlike the original HS structure, which uses fixed values for Harmony Memory Consideration Rate (HMCR), Pitch Adjustment Rate (PAR), and Bandwidth (BW), the improved version dynamically adjusts these parameters during the optimization process. HMCR value is adjusted from HMCRmax (maximum) to HMCRmin (minimum), while PAR and BW values are changed from small to large. Another improved version, known as Improved Harmony Search Algorithm (IHSA), was developed by [6]. In this version, the values of PAR and BW, which are the two important control parameters in the HS algorithm, are adjusted in each iteration of the search process. Additionally, a new variant called Global Dynamic Harmony Search Algorithm (GDHSA) was implemented by [7]. In GDHSA, both HMCR and PAR are dynamically changed according to the following steps: in the first 50% of the search process, the values of HMCR and PAR increase, while in the last 50% of the search process, the values of HMCR and PAR decrease. [8] Proposed the Dual Strategies and Adaptive Parameters Harmony Search Algorithm (DSAHS). In this approach, HMCR and PAR are initialized at 0.95 and 0.5, respectively. During the evaluation process, the values of these two parameters are updated based on the fitness of the solution vector.

3. HARMONY SEARCH ALGORITHM (HS)

Harmony search algorithm is a popular Meta-heuristic optimization algorithm. It was introduced by Geem and his colleague in 2001. It was developed based on the improvisation process which musicians follow in order to find the desired harmony [9]. HS algorithm has many advantages over other optimization algorithms:.

1. It does not require initial settings for the design variables,
2. it has a simple structure and few control parameters,
3. The derivative information is not required, because of the random search nature of the algorithm,
4. It has a good balance between exploration and exploitation [1,10].

As a result, HS algorithm has been applied widely in different science and engineering optimization problems [11]. During the improvisation process musicians have three options:

1. To play a known pitch stored in the memory.
2. To play something close to the known pitch stored in the memory.
3. To play s new pitch within the standard scope.

HS algorithm follows the same abovementioned concept during the search phase. The following steps describing the standard HS algorithm searching phases:

1- Formulating the optimization problem.

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

$$\min f(X) \qquad \text{where } x \in [LB, UB]$$

where f(X) is the objective function, and X is a set of possible solutions (decision variables). LB is the lower band and UB is the upper band of the boundary of the search space.

2- Initializing the algorithm control parameters. HS algorithm has number of control parameters including Harmony Memory (HM), Harmony Memory Consideration Rate (HMCR), Pitch Adjustment Rate (PAR), Bandwidth (BW), Maximum Iterations (MI).

3- Initializing the harmony memory. The HM is MxN matrix, and populated arbitrarily with values from the search space. The M is the size of HM (the number of rows of the HM matrix), and N is the number of decision variables of the optimization problem.

$$HM = \begin{vmatrix} x_1^1 & x_2^1 & . & . & x_n^1 \\ x_1^2 & x_2^2 & . & . & x_n^2 \\ x_1^3 & x_2^3 & . & . & x_n^3 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_1^m & x_2^m & . & . & x_n^m \end{vmatrix}$$

4- Generating a new set of harmony $X_{new=}(x_1, x_2, \ldots, x_n)$. The creating of the new possible solution is based on the following three rules:
   a. If R ≤ HMCR, choose the value from the HM. Where R is a random number in range [0 1].
   b. If R > HMCR, choose the value randomly from the available search space range.
   c. If the value was chosen from the HM (rule a), and R ≤ PAR, the value is changed slightly
   d. If the value was choosing from the HM (rule a), and R >PAR, the value remains without further modification.
   e. Repeat steps a to d for the variables (x_1 to x_n).

5- Updating the HM. If the fitness of the new generated harmony is better than the fitness of worst harmony stored in HM, then, the new harmony is included in the HM, and the worst harmony is excluded from the HM. Otherwise, do nothing.

6- Repeat the steps 4 to 5 until the maximum number of iterations is reached.

The proposed Modified Harmony Search algorithm

The Harmony Search Algorithm (HSA) has been widely applied to various optimization problems since 2001. The performance of the HSA relies heavily on its control parameters, which govern the search process behavior and ultimately impact the quality of the obtained solutions. One of the important control parameters is the Harmony Memory Consideration Rate (HMCR), which determines whether new potential solutions are chosen from the memory or the search space. In the original HSA, the

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

evaluation process begins with the random initialization of a matrix called HM, followed by the generation of a new set of potential solutions. During the generation process, the HMCR value is tested for each decision variable. With a probability of HMCR, a new value is randomly selected from the HM, while with a probability of (1 - HMCR), an arbitrary value is chosen from the search space. The HMCR value ranges between 0 and 1. The significance of the HMCR in the search for optimal values is evident. Opting for a low HMCR value increases the likelihood of selecting new values from the search space, leading to slower convergence. Conversely, selecting a high HMCR value causes the HSA to predominantly choose values from the memory, potentially trapping it in local optima. Hence, the selection of the HMCR value is critical. In contrast to the standard version of HSA, where HMCR remains fixed, in this paper, the proposed modification of the harmony search algorithm, assigns different values to HMCR during different stages of the evaluation process. Initially, HMCR is assigned a small value and gradually increased. This approach takes into account that, in the initial stages, the fitness of the solutions stored in HM tends to be low as these solutions were randomly selected. At this point, both the values in HM and the search space are equally likely to be good candidates for the optimal value. However, as the search process progresses, the fitness of the values stored in HM improves due to an increasing probability of better values and fitness with each iteration. The gradual increase in HMCR value in the modified HSA helps strike a balance between exploration and exploitation. It promotes exploration in the early stages to avoid premature convergence and encourages exploitation as the search progresses and potentially better solutions are discovered. This adaptive approach can enhance the convergence towards optimal or near-optimal solutions in the optimization process. In this research study, HMCR is assigned values of 0.7, 0.8, or 0.9, respectively. HMCR is initialized with 0.7 for the first 33% of the maximum iteration number and gradually increased. Table (1) presents the HMCR values corresponding to the iteration numbers. The other control parameters of HSA remain fixed throughout the search process, as shown in Table (2).

Table (1):
The proposed values of HMCR

HMCR value	Iterations	Example (number of iterations is 1000)
0.7	1% - 33%	1-333
0.8	33%-66%	334-666
0.9	66%-100%	667-1000

Table (2):
The commonly used HS control parameters values [12]

Harmony Memory size HM	40
Pitch adjustment Rate PAR	0.7
Bandwidth BW	0.2
Maximum Iterations	1000

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

4. METHODOLOGY

The objective of this study is to examine the impact of dynamically assigning values to the Harmony Memory Consideration Rate (HMCR) on the performance of the Harmony Search Algorithm. To accomplish this objective, the HMCR value which commonly set between 0.7 and 0.9 [4, 12], was utilized in MATLAB simulations under two scenarios. In the first scenario, the Harmony Search Algorithm was executed ten times with a fixed HMCR value of 0.7, and the corresponding fitness of the optimization problem was recorded for each run. The remaining control parameters followed the values specified in Table (2). The same procedure was repeated using two different HMCR values, specifically 0.8 and 0.9, while keeping the other control parameters constant. In the second scenario, the same approach was employed, but unlike the previous scenario, the HMCR value was dynamically assigned based on the values provided in Table (1). Table (3) presents the HMCR values along with the corresponding fitness results for each run and for both scenarios.

To assess the effectiveness of the optimization algorithm in each run, the Rosenbrock function was employed as a test function. The Rosenbrock function is a well-known unimodal function that possesses a single global optimum and is commonly used to evaluate the performance and robustness of optimization algorithms [13, 14]. The Rosenbrock function is notable for its curved, elongated valley-shaped contour, with a global minimum. The function has a narrow, curved ridge that makes it difficult for optimization algorithms to navigate towards the global minimum. Figure (1) shows the properties of the Rosenbrock function.

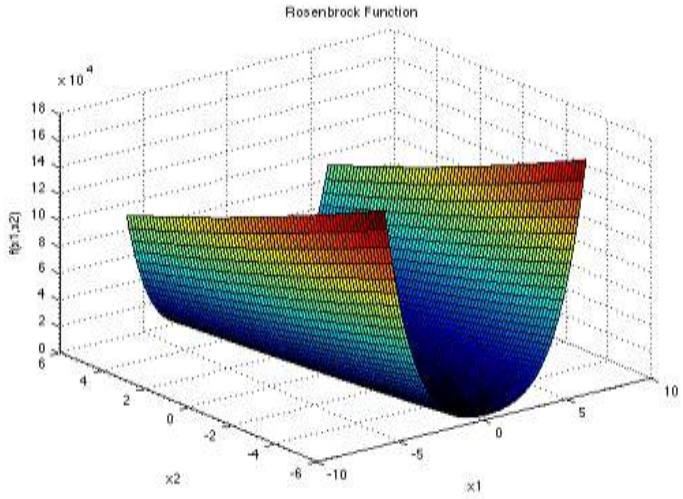

Figure (1): Rosenbrock Function [15].

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017
JOURNAL OF SCIENCES AND TECHNOLOGIES, VOL.(7), No.(2) Dec., 2023

Table (3):
The values of the fitness for HS when using fixed and dynamic HMCR

Run	Response (Fitness)			
	Fixed HMCR			Dynamic adjusted HMCR
	HMCR=0.7	HMCR=0.8	HMCR=0.9	HMCR = 0.7-0.9
1	0.0009700000	0.0005025000	0.0001069400	0.0002856900
2	0.0081000000	0.0012000000	0.0012000000	0.0007476700
3	0.0150000000	0.0016000000	0.0037000000	0.0008399300
4	0.0281000000	0.0037000000	0.0050000000	0.0024812000
5	0.0302000000	0.0066000000	0.0128000000	0.0035984000
6	0.0360000000	0.0222000000	0.0279000000	0.0260580000
7	0.0440000000	0.0307000000	0.0614000000	0.0314900000
8	0.0624000000	0.0736000000	0.0759000000	0.1149800000
9	0.0665000000	0.0952000000	0.0868000000	0.1155500000
10	0.1016000000	0.1204000000	0.1320000000	0.2112800000

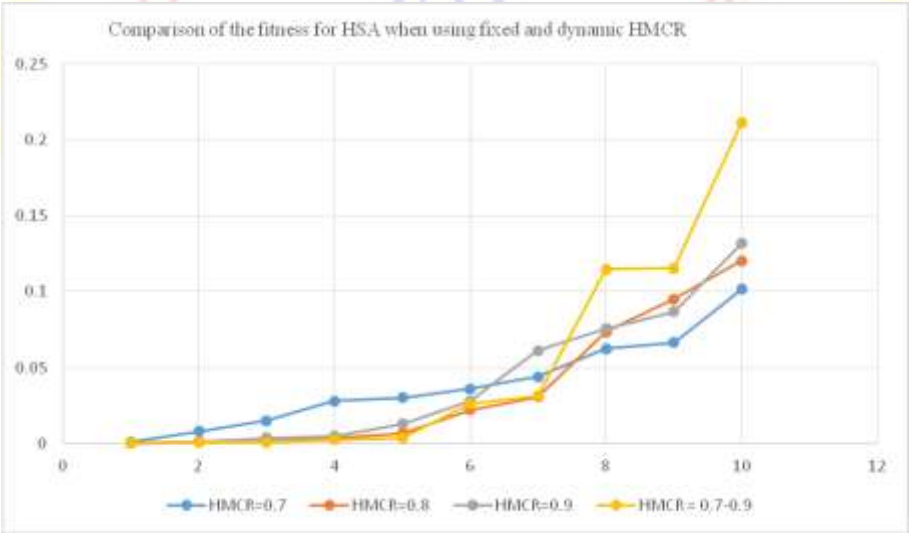

Figure (2): Comparison of the fitness for HS when using fixed and dynamic HMCR

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

5. RESULTS AND DISCUSSION

Table (3) and Figure (2) present the outcomes obtained through the application of the harmony search algorithm using fixed and dynamic values for the Harmony Memory Consideration Rate (HMCR). The results clearly indicate that utilizing an adaptable HMCR value leads to improved fitness in comparison to employing a fixed value. Specifically, when HMCR values of 0.7, 0.8, or 0.9 were assigned, the algorithm consistently converged to a single near-optimal solution in 1 out of 10 runs. These solutions were associated with the respective HMCR values of 0.7 (fitness: 0.0009700000), 0.8 (fitness: 0.0005025000), and 0.9 (fitness: 0.0001069400). Conversely, when the HMCR value was dynamically adjusted from 0.7 to 0.9 during the search process (scenario 2), the algorithm achieved three near-optimal solutions in 3 out of 10 runs. These solutions corresponded to HMCR values ranging from 0.7 to 0.9 and had fitness values of 0.0002856900, 0.0007476700, and 0.0008399300. The utilization of dynamic HMCR values facilitated more efficient exploration of the search space, striking a better balance between exploration and exploitation. Consequently, the algorithm displayed an enhanced capability to escape local optima and converge towards superior solutions. These findings underscore the effectiveness of employing dynamically adjusted HMCR values to enhance the harmony search algorithm's performance.

6. CONCLUSION

This study aimed to investigate the impact of utilizing a dynamic Harmony Memory Consideration Rate (HMCR) on the performance of the Harmony Search Algorithm. The objective was to assess whether the incorporation of an adaptable HMCR value could enhance the algorithm's efficiency. In contrast to employing a fixed HMCR value, the obtained results clearly indicate that the inclusion of adaptive HMCR significantly improves the performance of the Harmony Search Algorithm. These findings highlight the importance of considering the adoption of an adaptive HMCR approach rather than relying on fixed HMCR values. These results provide evidence that incorporating an adaptive HMCR value enhances the efficiency of the Harmony Search Algorithm by facilitating better exploration and exploitation of the search space, ultimately leading to improved solution quality. This research contributes to the existing knowledge of optimization algorithms by highlighting the effectiveness of parameter adaptability in achieving more efficient solution outcomes. Further investigations could focus on exploring the efficacy of utilizing the adaptive approach with other parameters of the Harmony Search Algorithm.

REFERENCES

1. Dubey, M., Kumar, V., Kaur, M., and Dao, T. P. (2021). A systematic review on harmony search algorithm: theory, literature, and applications. *Mathematical Problems in Engineering*, *2021*.
2. Jeong, Y. W., Park, S. M., Geem, Z. W., and Sim, K. B. (2020). Advanced parameter-setting-free harmony search algorithm. *Applied Sciences*, *10* (7), 2586.

ISSN: 2959 – 3298
received date: 20/07/2023
Acceptance date: 29/07/2023
LDN: 260/2017

3. Weyland, D. (2010). A rigorous analysis of the harmony search algorithm: How the research community can be misled by a "novel" methodology. *International Journal of Applied Meta-heuristic Computing (IJAMC)*, 1 (2), 50-60.
4. Theodossiou, N. P., and Kougias, I. P. (2012). Harmony search algorithm. *WIT Transactions on State-of-the-art in Science and Engineering*, *56*.
5. Tian, Z., and Zhang, C. (2018). An improved harmony search algorithm and its application in function optimization. *Journal of Information Processing Systems*, *14* (5), 1237-1253
6. .Mahdavi, M., Fesanghary, M., and Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, *188* (2), 1567-1579.
7. Khalili, M., Kharrat, R., Salahshoor, K., and Sefat, M. H. (2014). Global dynamic harmony search algorithm: GDHS. *Applied Mathematics and Computation*, *228*, 195-219.
8. Wang, Y., Guo, Z., and Wang, Y. (2017). Enhanced harmony search with dual strategies and adaptive parameters. Soft Computing, 21 (15), 4431-4445
9. Geem, Z. W., Kim, J. H., and Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, *76*(2), 60-68.
10. Botella Langa, A., Choi, Y. G., Kim, K. S., and Jang, D. W. (2022). Application of the Harmony Search Algorithm for Optimization of WDN and Assessment of Pipe Deterioration. *Applied Sciences*, *12* (7), 3550
11. Geem, Z. W. (2010). State-of-the-art in the structure of harmony search algorithm. In *Recent advances in harmony search algorithm* (pp. 1-10). Springer, Berlin, Heidelberg.
12. Geem, Z. W. (2006). Optimal cost design of water distribution networks using harmony search. Engineering Optimization, 38(3), 259-280.
13. Molga, M., and Smutnicki, C. (2005). Test functions for optimization needs. Test functions for optimization needs, 101, 48.
14. Picheny, V., Wagner, T., and Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. Structural and multidisciplinary optimization, 48, 607-626.
15. *Virtual Library of Simulation Experiments: Test Functions and Datasets*. (n.d.). *Virtual Library of Simulation Experiments: Test Functions and Datasets*. Retrieved July 17, 2023, from https://www.sfu.ca/~ssurjano/index.html