A Comparative Study of Improved Harmony Search Algorithm in Four Bar Mechanisms

LETICIA FLORES-PULIDO\textsuperscript{1}, EDGAR ALFREDO PORTILLA-FLORES\textsuperscript{2}, ERIC SANTIAGO-VALENTÍN\textsuperscript{2}, EDUARDO VEGA-ALVARADO\textsuperscript{2}, MARÍA BÁRBARA CALVA YÁÑEZ\textsuperscript{2}, AND PAOLA ANDREA NIÑO-SUÁREZ\textsuperscript{2}

\textsuperscript{1}Facultad de Ciencias Básicas, Ingeniería y Tecnología, Universidad Autónoma de Tlaxcala, Apizaco 90300, México
\textsuperscript{2}Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Ciudad de México 07738, México

Corresponding author: Leticia Flores-Pulido (aicitel.flores@gmail.com)

This work was supported in part by the Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDETEC-IPN), and in part by the Universidad Autónoma de Tlaxcala (UATx).

ABSTRACT There are problems that are difficult to solve through mathematical programming or by classical methods. These problems are called hard problems due to their high complexity or high dimension. On the other hand, metaheuristics intends to seek a better solution to a problem. The Improvement Harmony Search algorithm is proposed under modification of the bandwidth parameter increasing the quality of the exploitation of the solutions. That is why within the state of the art, are mentioned several versions of harmonic search. The state of the art supports the fact that the algorithm belongs to the category of those who make modifications to its parameters. This research demonstrates the ability of ImHS to solve a problem of high complexity focused on solving four-bar mechanism designs, whose solutions imply high dimension and which are also classified as hard problems. The two problems that are solved in this investigation, are problems very attacked within the state of the art by various metaheuristics. A comparison is then made against previous solutions with traditional metaheuristics and other versions of harmony search algorithm. Finally, the effectiveness of performance is demonstrated, where proposed algorithm it exceeded five metaheuristic algorithms and five harmony search versions. An optimum is provided in an easy and useful way, the parametric statistics are improved and the number of feasible solutions is exceeded in NP-hard problems as in the case of problems with four-bar mechanisms.

INDEX TERMS Algorithms performance, four-bar mechanism, improved harmony search, optimization problems, mechatronic.

I. INTRODUCTION
In the area of mechatronics, the four-bar mechanisms are widely applied [1]. Their application range goes from the health sector to the industrial context and they are used in tasks as diverse as welding, painting, robot trajectories, surgical needles and laser shots. On the other hand, the problems of metaheuristic computation have been widely tested and modified with the intention of achieving more efficiency in their application to real world problems, as is the case of mechatronics. One of these approaches is the Improved Harmony Search algorithm (ImHS) [2], where is proposed with the intention of increasing the quality of the exploitation of the solutions, based in a modification of bandwidth parameter.

The associate editor coordinating the review of this manuscript and approving it for publication was Seyedali Mirjalili.\textsuperscript{2}
most recent versions of the last ten years whose focus is on improving parameters, specifically on bandwidth and the generation of harmonic memory.

Regarding all the HS variations that existed in the last decade, there are 21 revised algorithms, of which 4 are hybridized with HS and 17 are versions that focus on improving parameters. The parameter improvements mainly in the bandwidth parameter, shows that the modification of said parameter is not something trivial within the recent works in the last decade.

II. STATE OF ART IN TRADITIONAL METAHEURISTICS

In this work, a variant of ImHS is applied for the dimensional synthesis of four-bar mechanisms. The objective is to evaluate the performance of the Improved Harmony Search algorithm (ImHS) [2], comparing it to four metaheuristic algorithms previously applied in the related literature for designing four-bar mechanisms [1]. Also, ImHS is compared versus five versions of HS. Two case studies are considered for the comparison. The algorithms used int this comparative analysis are ABC [4], [5], ACOR [6], [7], JADE [8], JAYA [9], [10], HS [11], IHS [12], ABHS [13], OHS [14] and MHS [15].

ABC is inspired by the behavior of the bees in a hive, with the purpose of giving solutions to optimization problems in a distributed way. This species is controlled under certain constraints to work as individuals or jointly. ACO [16] is another optimization algorithm based on a behavioral approach of species that collaborate socially. ACOR is a variant that is based on multi-agent systems, where each agent behave as an individual ant. It is an efficient algorithm, which has been applied to solve from the traveling salesmen agent problem to routing data problems of computer networks or the Internet [6].

JADE is a version of the Differential Evolution algorithm, proposed to improve its optimization performance. A novel mutation strategy is implemented with an optional external file, updating the control parameters in an adaptive way. Historical data is used to provide information to a search direction for the optimum. Both operations diversify the population and improve convergence [8]. Another algorithm within the state of the art is JAYA, proposed to solve constrained and unconstrained problems. The approach is based on guiding the search direction of the algorithm towards a better solution, avoiding the worst solutions. The algorithm was tested with a series of benchmarks [9], producing good solutions.

III. STATE OF ART IN HYBRIDIZED ALGORITHMS COMBINED WITH HS

Harmony Search (HS) [11] is a very simple optimization algorithm developed in 2001, that offers the possibility of solving diverse linear and non-linear engineering problems. HS is based on the musical composition process, and its implementation requires a low quantity of computational resources. The HS algorithm has been improved since its inception; and a few examples are mentioned in this section.

The development done in [17], the combination of HS and Optimum-Path Forest (HSOPF) is explored with the intention of accelerating the convergence of HS to discriminate those relevant characteristics in the search for the optimum of an objective function. In the context of an optimal forest path algorithm, each sample is linked to another by means of a distance. The proposed hybrid algorithm takes OPF as the fitness function to guide the HS algorithm. Basically, OPF classifier is trained for each harmony with the corresponding subset of features in a training set.

The work done in [18] it makes a comparison of seven heuristic and non-heuristic methods and the modification of HS called DHS, which is a method that combines HS with Differential Evolution. A hybrid is then achieved with the cross and mutation parameters. This hybridization is performed with the intention of avoiding the premature convergence of HS and improving the ability to explore differential evolution.

The research done in [19] It proposes a hybridization between HS and an aptitude function, with the intention of reducing the convergence time and accelerating the similarity between vectors that describe movement in video in real time. This approach is called harmonic search under the block matching approach (HS-BM). The parameter that is modified is the calculation of the harmonic memory, where by means of a kind of aptitude window, the search for better candidate vectors is performed to tie the video search in real time.

In [20] a self-adaptive harmony particle swarm optimization search algorithm, named (SHPSOS) is proposed. In this investigation, an efficient assignment of values to each of the HS parameters is proposed from the beginning, to ensure the quality of the solution. A self-adaptive adjustment scheme for bw and pitch is proposed, in order to balance the rapid convergence and high diversity during the improvisation step of the harmonic memory. Harmonic memory is combined with differential evolution, a Gaussian strategy is used for the mutation operator, and the global solution is evaluated by means of hidden Markov models.

IV. STATE OF THE ART IN HS VERSIONS FOCUSED ON PARAMETERS IMPROVEMENTS FOR THE LAST 10 YEARS

HS is an algorithm designed by [11], but tested in [12], where the objective is to accelerate convergence and improve accuracy. Specific issues are analyzed such as the strategy for parameter tuning and the influence of constant parameters. The effectiveness of the improvement was evaluated through a benchmark of functions. One of the most complete surveys about HS is in [13], where the authors analyze the development of HS until the recent improvements. Several investigators modified HS in its emergence, searching for an improvement from different points of view. In that work, an analysis of the functionality and performance of HS is made, and improvements are proposed from two points of view: the parameter tuning strategies, and the hybridization forms of the algorithm.
Another work in terms of HS improvements is the work done in order to raise the efficiency of harmonic search algorithms with intersection mutation operators. A hybridization is made regarding the use of a cellular automata local search for various optimization problems [15]. The relevance of this work is based on the improvement with respect to the harmonic memory, that is, it does not focus on the pitch strategy, but focuses on dividing the harmonic memory into two main parts. The first part consists of those solutions whose function of aptitude are not good, the second part of the harmonic memory consists of those solutions where the function of aptitude is good. The exploration occurs when the search space is wide in the first executions. The exploitation occurs when within a small area a refinement of the final solutions is carried out. Other works and improvements can be viewed in [21] and [22].

In the work carried out by [13], there is a chronological study regarding HS modifications from 2001 to 2010. Some of the modifications made to HS during the last 10 years (2011-2019) are mentioned in this section. Even the precise difference between these versions and the ImHS variant will be highlighted. The work done in [23], shows an approach which consists of two stages. In the first stage, HS explores the search space to find out the near-optimal cluster centers (HS-c means). The cluster centers are evaluated using reformulated c-means objective function. In the second stage, the best cluster centers found are used as the initial cluster centers for the c-means algorithms. These experiments show that HS can minimize the difficulty of choosing an initialization for the c-means clustering algorithms.

The research explained in [24], proposes a combination of harmonic search with neural networks (HS-ANN). The important elements of this variants are based on the design of experiments, analysis of variance and neural networks. The modification consists in testing with experimental data sets but the neural network adjusts the parameters of the algorithm with the intention of providing a better search area to achieve the optimum for each experiment. That is, the neural network adjusts and optimizes the search for the appropriate pitch of the HS algorithm.

In the research carried out in [25] two versions of HS are proposed, the harmonic search efficiency algorithm (EHS) and the harmonic search self-adaptive (SAHS). The modifications are of the tone parameter. EHS dynamically updates pitch and bandwidth during the search process. On the other hand, in the SAHS algorithm, the pitch is dynamically updated but the bandwidth is completely remove [26]. These algorithms mimics performance processes of the musical improvisation for finding the best succession of pitches in a melody, utilizing different harmonic memories.

The algorithm called Intelligent Tuned Harmony Search (ITHS) consists in the analysis of the sensitivity effects in two parameters: the size of the harmonic memory and the speed of consideration of the harmonic memory, surpassing eight more techniques and 17 benchmark functions [27].

The research done in [28] is a review of the different applications of HS is presented in different contexts. These contexts are: construction, energy, robotics, telecommunications, health and energy. In this review, it is indicated that 23 % of publications regarding HS focus on algorithm modifications, 31 % on engineering applications, 19 % on diverse applications and 14 % are applications to the energy sector. But this study is only carried out until the algorithmic modifications made until 2013.

The study of [29] proposes an algorithm for no restricted optimization in continuous problems. The algorithm is called harmonic search based on machine learning (LAHS). LAHS proposes a tuning of parameters, including tone and bandwidth or frequency (bw). The modification consists of two stages: the learning stage and the parameter selection stage. The essence of the selection of parameters and the learning stage are based on the roulette method, where values are chosen within certain ranges taking into account a previously assigned probability.

By the other hand in [30] and [31] it is proposed an algorithm called modified harmony search-based clustering (MHSC), where harmonic memory it should be dynamically changed during the improvisation step improving the optimization process.

In the work done by [32] an improved version of the harmonic search (EnHS) is proposed through an improvement based on a probabilistic selection within the process of creating harmonic memory. The competitiveness of the algorithm is tested against the original harmonic search version with a benchmark of four important problems within the context of steel frame balancing design.

A modified harmony search (MHS) algorithm is proposed in [33], which consists of a modification based on iterative convergence. Specifically, the parameters that are affected are harmonic memory, cross, and tone adjustment. By adjusting the calculations of these parameters, exploitation and exploration are balanced.

In the review carried out by [34] a statistic regarding HS applications is exposed, as well as the changes that the algorithm has undergone among 35 publications during the years close to 2015. In this analysis it is reported that 29 % of the publications refer to the use of HS in classification, 17 % about prediction, 21 % in feature selection and 33 % in terms of grouping. All of the above within data mining. The algorithms mentioned in terms of variations of HS are: HS self-adaptive (SGHS), HS with neural networks, HS-K means (HKA), HS hybridized with Bayesian criteria, HS hybridized with cloning, HS based on clustering, HS with search in forests and Differential HS. The applications vary from voice recognition, types of wines, irises, oils and detection of thyroid diseases.

Saparudin and Kurniawan in 2016 [35] propose a dynamic change of tone and a bandwidth adjustment for image compression. These parameters are dynamically loaded according to the generation and to the solution vectors. Comparisons are
made against original HS. In [36], the authors have intended as an intention to accelerate convergence which is achieved by considering the opposite solutions (OHS). Each of the solution vectors of the harmonic memory considers its decision rule, which also happens with the selection of the tone. In addition, there is a re initialization process which gives an optimal result corresponding to a minimum error within a space of mutual search.

In the work done by [2] a combination of two modified versions of HS is made: Improved HS (IHS) and the Adjustable Band-Width (ABHS) algorithm. Both versions are concentrated in a single algorithm called Improved Harmony Search (ImHS), whose performance is analyzed in the present document. ImHS consists of a modification to two parameters: the bandwidth according to IHS and the intelligent factor according to ABHS. Then the first proposed modification in this work is the reduction of band width as a exponential function of the number of iterations, in order to change the exploitation in the neighborhood of a potential optimum. The second proposed modification is about proper selection of intelligent factor whose is fundamental in order to avoid early convergence.

In the investigation [37] a random distribution is proposed in the creation of the HS Harmonic Memory (RDHS). The quality of the selection of the solution vectors of the harmonic memory is increased, the convergence is monitored and a new parameter is added which will maintain the values of the random distribution with which a new set of solutions is generated. Other work related proposes a dynamically adaptive parameter with fuzzy logic in [38]. Tests are carried out under three different fuzzy systems, thus achieving better intensification and better diversification. The resulting algorithms are: FHS (type 1), IT2FHS (type 2) and GT2FHS (generalized type 2). These methods use triangular membership functions that are applied to the benchmark functions.

A mutation-based harmony search (MBHS) algorithm is implemented in [39]. Where the idea is to modify the mutation parameter that can be combined and that supports different web services under the minimum number of defects (with the fewest number of restriction violations). MBHS is shown to be better than other heuristics and finds appropriate solutions that improve business planning in the application context.

In the research carried out by [40] is proposed a modification to the HS algorithm based on the probabilistic approach of the traveling agent (HS-ATSP). The modification is made in the selection of optimal solutions within the harmonic memory. The modified parameter is the pitch and is selected using the roulette method.

After the aforementioned works, it can be observed that all the modifications, improvements or tuning made to HS algorithm goes in two directions: (a) The first class consisting of hybridization of HS with other meta heuristics which are summarize in Table 1. (b) The second class consists of variants of the HS algorithm based on parameter tuning improvements as can be observed in Table 2 (Part 1) and Table 3 (Part 2).

This article is structured as follows: in the present section (Section I. Introduction) an explanation of the context of proposed algorithm is provided. The Section II. State of Art in Traditional Metaheuristics, shows the state of art of four algorithms under heuristic approaches to solve optimization problems. This four algorithms will be compared in performance metrics against ImHS taking the 2 study cases exposed by Sleesomgsom and Bureerat [1]. The Section III and IV explains the state of art of 22 versions of HS in the last ten years.

Section V. Problem Definition, details the optimization problem, the proposal of this research and the origins of the 2 cases of study in the context of mechatronic design. The corresponding objective functions, the limits of the variables, the minimum to be found and the restrictions are also shown. The diagrams of the local coordinates of both mechanisms are also exposed within the case studies that are implemented and which are to be optimized.
The Section VI. Improved Harmony Search Algorithm is explained, also the performance, advantages and disadvantages.

The Section VII. Results and Evaluation of ImHS - Part One-, explain the values of the parameters of tuning and the values of best mechanisms obtained in both cases of study and the comparative tables of the algorithmic performance of ImHS against the other 7 algorithms.

The Section VIII. Results and Evaluation of ImHS - Part Two-, shows the values of the parameters of tuning and the comparative tables of the algorithmic performance of ImHS against the other 5 variations of HS algorithms.

The feasibility and convergence graphs are exposed in Section IX. Feasibility and Convergence of ImHS Vs. Other HS Variations.

The analysis of results are discussed in Section X. Finally, in the Section XI. Conclusions are provided the findings and a complexity analysis of the algorithm.

V. PROBLEM DEFINITION

An optimization problem requires a mathematical model with a general structure, which is also known as a mathematical programming model and can be represented as follows:

Find $x$ to Maximize or Minimize

$$f(x)$$  \hspace{1cm} (1)$$

Subject to

$$g_i(x) \leq g_{bi}, \quad i = 1, \ldots, m$$  \hspace{1cm} (2)$$

$$h_j(x) = h_{bj}, \quad j = 1, \ldots, p$$  \hspace{1cm} (3)$$

$$x \geq 0$$  \hspace{1cm} (4)$$

where the objective function $f$ is a function of a single variable $x$, and the constraint functions $g_i$ and $h_j$ are general functions of the variable (otherwise) expressed as an unknown, decision variable or sometimes as a parameter) $x \in R^n$. The right hand sides, $g_{bi}$ and $h_{bj}$ are usually the known constants for deterministic problems. The non-negativity constraint, $x \geq 0$, is necessary for many practical problems (since many variables cannot be negative) and for many solution approaches (assumption by default). The above standard model may vary as follows: (1) contains upper and lower bounds of $x$ instead of a non-negativity constraint, (2) contains upper and lower bounds of $x$ instead of any other constraint, and (3) the above standard model, with or without (1) and (2) with multiple variables [41].

Let us assume $x^*$ represent a set of variables, where $x^* = (x_1, x_2, \ldots, x_3)$, then the above model can be rewritten for multiple variables as follows:

Find $x^*$ to Maximize or Minimize

$$f(x^*)$$  \hspace{1cm} (5)$$

The Section VI. Improved Harmony Search Algorithm is explained, also the performance, advantages and disadvantages.

The Section VII. Results and Evaluation of ImHS - Part One-, explain the values of the parameters of tuning and the values of best mechanisms obtained in both cases of study and the comparative tables of the algorithmic performance of ImHS against the other 7 algorithms.

The Section VIII. Results and Evaluation of ImHS - Part Two-, shows the values of the parameters of tuning and the comparative tables of the algorithmic performance of ImHS against the other 5 variations of HS algorithms.

The feasibility and convergence graphs are exposed in Section IX. Feasibility and Convergence of ImHS Vs. Other HS Variations.

The analysis of results are discussed in Section X. Finally, in the Section XI. Conclusions are provided the findings and a complexity analysis of the algorithm.

V. PROBLEM DEFINITION

An optimization problem requires a mathematical model with a general structure, which is also known as a mathematical programming model and can be represented as follows:

Find $x$ to Maximize or Minimize

$$f(x)$$  \hspace{1cm} (1)$$

Subject to

$$g_i(x) \leq g_{bi}, \quad i = 1, \ldots, m$$  \hspace{1cm} (2)$$

$$h_j(x) = h_{bj}, \quad j = 1, \ldots, p$$  \hspace{1cm} (3)$$

$$x \geq 0$$  \hspace{1cm} (4)$$

where the objective function $f$ is a function of a single variable $x$, and the constraint functions $g_i$ and $h_j$ are general functions of the variable (otherwise) expressed as an unknown, decision variable or sometimes as a parameter) $x \in R^n$. The right hand sides, $g_{bi}$ and $h_{bj}$ are usually the known constants for deterministic problems. The non-negativity constraint, $x \geq 0$, is necessary for many practical problems (since many variables cannot be negative) and for many solution approaches (assumption by default). The above standard model may vary as follows: (1) contains upper and lower bounds of $x$ instead of a non-negativity constraint, (2) contains upper and lower bounds of $x$ instead of any other constraint, and (3) the above standard model, with or without (1) and (2) with multiple variables [41].

Let us assume $x^*$ represent a set of variables, where $x^* = (x_1, x_2, \ldots, x_3)$, then the above model can be rewritten for multiple variables as follows:

Find $x^*$ to Maximize or Minimize

$$f(x^*)$$  \hspace{1cm} (5)$$

The Section VI. Improved Harmony Search Algorithm is explained, also the performance, advantages and disadvantages.

The Section VII. Results and Evaluation of ImHS - Part One-, explain the values of the parameters of tuning and the values of best mechanisms obtained in both cases of study and the comparative tables of the algorithmic performance of ImHS against the other 7 algorithms.

The Section VIII. Results and Evaluation of ImHS - Part Two-, shows the values of the parameters of tuning and the comparative tables of the algorithmic performance of ImHS against the other 5 variations of HS algorithms.

The feasibility and convergence graphs are exposed in Section IX. Feasibility and Convergence of ImHS Vs. Other HS Variations.

The analysis of results are discussed in Section X. Finally, in the Section XI. Conclusions are provided the findings and a complexity analysis of the algorithm.
Subject to
\[ g_i(x^*) \leq gb_i, \quad i = 1, \ldots, m \]  \hspace{1cm} (6)
\[ h_j(x^*) = hb_j, \quad j = 1, \ldots, p \]  \hspace{1cm} (7)
\[ x^* \geq 0 \]  \hspace{1cm} (8)

Once both case studies have been modeled, we proceed to solve them with the improved harmony search algorithm (ImHS). Upon reaching feasibility and satisfactory convergence, tests are performed to execute the algorithm with the search for 200 and 300 mechanisms. These tests are then configured as follows:

1) Case 1 with the variation \( M_1 = 200 \)
2) Case 1 with the variation \( M_2 = 300 \)
3) Case 2 with the variation \( M_1 = 200 \)
4) Case 2 with the variation \( M_2 = 300 \)

A comparison is made with nine meta heuristic methods, where convergence tables and feasible solution tables are analyzed within the optimal cases for harmonic memory. The parametric statistics of average, minimum, maximum, standard deviation, execution times, the number of successful executions for each case study and the value of \( M \) for each of the tests are also shown.
The Figure 1, shows the proposal of this investigation. Initially, a modeling of the global and local coordinate systems of the four-bar mechanism is carried out. As a second stage, two cases of study of the state of the art are implemented, which have already been taken as a standard in the evaluation of other heuristic algorithms. As a next phase, optimization problems are tuned in and provided as input to the ImHS algorithm and other five versions of harmony search. Finally a feasibility and convergence analysis is carried out.

This section provides a detailed description of the two case studies used in measuring performance between ImHS and the other 10 comparisons versus heuristic algorithms of the state of the art. Case Study 1 and Case Study 2 are used to evaluate the performance of 5 of the heuristic algorithms in [1]. Case studies are optimization problems that involve a four-bar mechanism in mechatronics. The application of these optimization problems are of vital importance in applications within the health sector, mechanical and industrial applications where monitoring of several trajectories is required below the desired points.

These problems make use of the same objective function, but the restrictions, limits and the number of points to travel through the final pin are different. The objective function, for both case 1 and case 2, are displayed in Equations (1) and (10) respectively. For both cases, the variables to be optimized are nine. Case 1 are stated as optimization problem in Equations from (1) to (9). For Case 2, the optimization problem are defined from Equation (10) to Equation (18). Variables, restrictions, desired points, objective function, limits and minimum desired values are shown below.

### A. CASE 1: CIRCLE AS PATH GENERATION WITHOUT PRESCRIBED TIMING

The objective function of the Study Case 1 in four bar mechanism is described in Equation (9):

\[
\min f(x) = \sum_{i=1}^{N} [(x_{d,i} - x_{P,i})^2 + (y_{d,i} - y_{P,i})^2]	ag{9}
\]

subject to two restrictions detailed in the Equations (10, 11):

\[
\min (r_1, r_2, r_3, r_4) = \text{crank}(r_2) \tag{10}
\]

\[
2\min (r_1, r_2, r_3, r_4) + 2\max (r_1, r_2, r_3, r_4) < (r_1, r_2, r_3, r_4) \tag{11}
\]

Thus, the design variables are shown in (12):

\[
x = \{r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0\}\tag{12}
\]

where \(r_1, r_2, r_3\) and \(r_4\) are the length of the four bars in the mechanism, \(r_{cx}\) and \(r_{cy}\) are the length of the final pin, \(\theta_0\) is the angle of the mechanism inclination, and \(x_0\) and \(y_0\) are the coordinates of the origin point where the mechanism must be fixed for its effective trajectory over the desired points. So, the six desired points are detailed in (13).

\[
r_{d} = \{(20, 20), (20, 25), (20, 30), (20, 35), (20, 40), (20, 45)\}\tag{13}
\]

Limitation of the nine design variables are stated for the lower and upper limits in the \(s\) (14), (15), (16) and (17).

\[
5 \leq r_1, r_2, r_3, r_4 \leq 60 \tag{14}
\]

\[
0 \leq r_{cx}, r_{cy} \leq 60 \tag{15}
\]
-60 \leq x_0, y_0 \leq 60 \quad (16)

0 \leq \theta_0 \leq 2\pi \quad (17)

B. CASE 2: LINE AS PATH GENERATION WITHOUT PRESCRIBED TIMING

The second problem of the four-bar mechanism, raises a case study of mechatronic design, using the same objective function shown in the Equation (9) for case study 1. For the convenience of the reader, the objective function is shown again in the Equation (18).

\[ \min f(x) = \sum_{i=1}^{N} [(x_{d,i} - x_{p,i})^2 + (y_{d,i} - y_{p,i})^2] \quad (18) \]

subject to the following constraints, detailed in Equations (19) and Equation (20):

\[ \min (r_1, r_2, r_3, r_4) = \text{crank}(r_2) \quad (19) \]

\[ 2\min (r_1, r_2, r_3, r_4) + 2\max (r_1, r_2, r_3, r_4) < (r_1, r_2, r_3, r_4) \quad (20) \]

The nine design variables to optimize, are listed in Equation (21):

\[ x = \{r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta^0, x_0, y_0\} \quad (21) \]

where \( r_1, r_2, r_3, r_4 \) are the lengths of the four bar in the optimum mechanism to find, \( r_{cx}, r_{cy} \) are the length of the pin in the mechanism, \( \theta^0 \) is the inclination angle of the mechanism at the ground, and \( x_0, y_0 \) are the origin coordinates where the mechanism must be placed in a coordinate system.

The eleven desired points to complete a perfect trajectory, can be observed in (22):

\[ r'_d = \{ (20, 20), (17.66, 15.142), (11.736, 17.878), (5, 16.928), (0.60307, 12.736), (0.60307, 7.2638), (5, 3.0718), (11.736, 2.1215), (17.66, 4.8577), (5, 3.0718), (20, 10) \} \quad (22) \]

The upper and lower limits of range for one of each design variables, are provided in (23), (24), (25) and (26).

\[ 5 \leq r_1, r_2, r_3, r_4 \leq 80 \quad (23) \]

\[ 0 \leq r_{cx}, r_{cy} \leq 80 \quad (24) \]

\[ -80 \leq x_0, y_0 \leq 80 \quad (25) \]

\[ 0 \leq \theta_0 \leq 2\pi \quad (26) \]

Once the optimization problems for case 1 and case 2 have been exposed, the local coordinate diagram for both designs can be seen in Figure 2. The global coordinate diagram for both mechanisms can be seen in Figure 3. The design of the four-bar mechanism labeled by \( r_1, r_2, r_3 \) and \( r_4 \), fixed points on the ground represented by \( r_{cx} \) and \( r_{cy} \) and inclination angle represented by \( \theta^0 \).

VI. IMPROVED HARMONY SEARCH ALGORITHM

The original harmonic search algorithm is based on the musical composition, put simply, on the harmonious combination of a set of musical notes. Harmonies can be formed in three ways according to musicians: a) can be achieved by combining harmonies previously used, b) may also be formed by adjusting the above harmony altering some notes, or c) using a harmony completely new with randomly selected notes [11].

We propose a novel algorithm previously exposed in [2] called Improved Harmony Search, and in the flow chart shown in Figure 4 are broadly exemplify the steps of the algorithm. Initially we have the stage that gives values to each of the variables of our algorithm: the pitch adjusting rate, the harmonic memory (HM) rate and the constant a. Then, the generation of first HM, is done and \( x_{\text{worst}} \) is computed form it.

On the other hand, the cycle of the number of generations begins, within which bandwidth is calculated, comparisons are made between a first random number \( r_1 \) and the accepting rate. If this is true, a second random number \( r_2 \) is generated and compared against the tone, if it is true, a \( x_{\text{new}} \) is generated and compared against the \( x_{\text{worst}} \) of the initial generation.

In case the previous comparisons against accepting rate are false, it is compared against a third random value \( r_3 \). If this false comparison results, then the tone is compared again. If true, \( x_{\text{new}} \) is generated as a second branch.

Finally, the one that is smaller is chosen between \( x_{\text{worst}} \) and \( x_{\text{new}} \). but also, Deb’s Rules are applied to know if the restrictions are met. In the event that they are not fulfilled,
the solution that meets the majority of them is taken and awarded as the winning solution. The algorithm ends.

The objective in this investigation is the evaluation of the performance of the improved harmony search algorithm (ImHS) [2] that has the following parameters: number of generations \((g)\), size of harmonic memory \((N)\), percentage of new solutions accepted \((r_{\text{accept}})\), harmonic memory pitch \((r_{\text{pa}})\) and intelligence factor (constant \(a\)) \((f_i)\). The original version of the harmony search algorithm is applied to optimization problems associating vectors of design variables \(X_j = (x_{j1}, x_{j2}, \ldots, x_{jN})\), which are called harmonic vectors. The set of vectors is known as the components of a harmonic memory (HM). The generation of this harmonic memory is governed by the lower limits \(L_i\) and higher limits \(U_i\). Therefore, harmony memory is defined in (27) as:

\[
HM = \begin{bmatrix}
X^1 & x_1^1 & x_1^2 & \ldots & x_1^N \\
X^2 & x_2^1 & x_2^2 & \ldots & x_2^N \\
X^3 & x_3^1 & x_3^2 & \ldots & x_3^N \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
X^N & x_N^1 & x_N^2 & \ldots & x_N^N 
\end{bmatrix}
\] (27)

Then, a new harmony value \(x_{i,\text{new}}\) indicated by the pitch as \(r_{\text{accept}} \in (0, 1)\) and \(r_{\text{pa}} \in (0, 1)\) is created, that allows a pitch
adjustment. The three principles where $x_{i}^{\text{new}}$ is generated are:

1. With a probability $r_{\text{accept}}$, a value already stored in the harmonic memory is randomly selected.
2. With a probability $r_{\text{accept}} \ast r_{\text{pa}}$ the Equation (28) is used, where $x_{i}^{\text{old}}$ is the original pitch and $bw$ is the width of band for pitch adjustment.
3. With a probability of $(1 - r_{\text{accept}})$, the randomization produces a completely new value in the Equation (29):

$$x_{i}^{\text{new}} = x_{i}^{\text{old}} + bw \ast \text{rand}(-1, 1) \quad (28)$$

$$x_{i}^{\text{new}} = \text{rand}(L_{i}, U_{i}) \quad (29)$$

One of the major strengths of HS original is its balance between diversification and intensification; $r_{\text{accept}}$ configures the global search stage while $r_{\text{pa}}$ and $bw$ corresponds to refinement of solutions, as the probability of local improvement and the search radius, respectively. If every $x_{i}^{\text{new}}$ produces a better value of the objective function then replaces the worst harmony $x_{\text{worst}}$, updating the HM. The process is repeated until a maximum number of iterations is reached in its original version.

In this sense, the modifications or improvements made to the ImHS are mentioned below. It must be taking into account that initially can be feasible areas and are delimited by a set of restrictions. So, the modification already made to HS detailed in [12] goes in that direction. It is important to mention that the feasibility rules of Deb are considered at all times. First: It must be considered that between two feasible vectors, only the one with the best value of the objective function is selected. Second: Between a feasible vector and an unfeasible vector, the feasible solution is selected. Third: Between two unfeasible vectors, the one with the least amount of violation of constraints is selected.

All constraints are treated as inequalities when applying the (30), with a previously established tolerance $\epsilon$ that tends to zero [2]:

$$|h_{k}(\vec{x})| - \epsilon \leq 0 \quad (30)$$

The Improved Harmony Search algorithm (ImHS) has a procedure known as the bandwidth adjustment, called in the algorithm as $bw$ which in musical terms means the change of frequency to generate a new tone slightly similar to the original. In the context of the algorithm, to avoid a process of slow convergence, it has been detected then that a low value for $r_{\text{pa}}$ together with a high value of $bw$ can limit the points of exploration [13]. The proposal then is to calculate $r_{\text{pa}}$ and $bw$ depending on the number of cycles and the current iteration, increasing the number of points of exploration [15] through a kind of soft step

The bandwidth starts at $bw_{\text{max}}$ and decreases exponentially to $bw_{\text{min}}$ at the end of the simulation. But this scheme has two disadvantages: the first is that there is only one bandwidth for all the variables, and the second is that it is necessary to know the number of iterations in advance.

On the other hand, the potential to reach the optimum within the harmonic memory depends on the number of cycles as $g$ increases. Meanwhile, the exploitation acquires a greater priority in front of the exploration. The improvement for ImHS lies in the reduction of $bw$ exponentially and as a function of the number of iterations [21]. So, the bandwidth is defined in the (31), where $a$ is a positive constant in the range of $0 < a \leq 1$ since a higher value could indicate a slow convergence. The behavior of $bw$ as a function of $g$ can be seen in Figure 5.

$$bw = \frac{(U_{i} - L_{i})}{g^{a}} \quad (31)$$

The pseudo code of the Improved Harmony Search is shown in the Algorithm 1.

Algorithm 1 ImHS

```plaintext
define objective function $f(x), x = (x_{1}, x_{2}, \ldots, x_{N})$
; define harmony memory accepting rate $r_{\text{accept}}$
; define pitch adjusting rate $r_{\text{pa}}$
; define constant $a$ as intelligent factor
; generate randomly first Harmony Memory HM
; while $g < \text{max number of iterations}$
while $i < N$
$$bw = \frac{(U_{i} - L_{i})}{g^{a}}; \quad \text{(Mod_1)}$$
else if $\text{rand} < r_{\text{accept}}$
index = $\text{rand}(1, k)$
else if $\text{rand} < r_{\text{pa}}$
$newX(i) = HM(\text{index}, i) + bw \ast \text{rand}(-1, 1)$
else if $\text{rand} < r_{ia}$
$newX(i) = HM(\text{best}, i)$
$newX(i) = HM(\text{index}, i)$
\text{(Mod_2)}
$newX(i) = \text{rand}(L_{i}, U_{i})$
accept $newX$ as new solution if better than worst harmony $X_{\text{worst}}$, applying Deb’s conditions;
```

The enhancement described in [2] is exposed here for comprehension of reader. The trade-off previously exposed in this section is not only about the wide band for search space of feasible solutions, but, an adjustment in $r_{ia}$ parameter of ImHS. This enhancement make a replacement of $x_{\text{new}}$ by
TABLE 4. Configuration of experiments for performance analysis of ImHS.

|   |   |   |   |
|---|---|---|---|
| 1 | 30 | Case 1 | 200 |
| 2 | 30 | Case 1 | 300 |
| 3 | 30 | Case 2 | 200 |
| 4 | 30 | Case 2 | 300 |

where the best solution is replaced with a probability described in Equation (32).

$$prob_{a} = r_{accept} \times (1 - r_{pa}) \times r_{ia}$$  (32)

where $r_{accept}$ and $r_{pa}$ are probability operators that will allow the passage of a new feasible solution. On the other hand, $r_{ia}$ is already known as an intelligent factor whose function is focused on redirecting the search for new solutions. The risk of a premature convergence, increases as $prob_{a}$ approaches to 1.

In the original harmonic search algorithm, the following ranges are recommended: for $r_{accept}$ of $(0.7 - 0.9)$; for the case of $r_{pa}$ the recommended range is $(0.1 - 0.5)$; in such a way that a correct choice of $r_{ia}$ is fundamental to avoid an early convergence.

This research considers for ImHS a accepting rate $r_{accept} = 0.2$, a pitch value $r_{pa} = 0.9$, and an intelligent factor $r_{ia} = 0.9$. Work with 50 harmonies and 50,000 iterations ($g$). Two values are considered for the number of mechanism posibles: $M$, where it is modified to $M_1 = 200$ in test for case 1 and $M_2 = 300$ in tests for case 2.

VII. RESULTS AND EVALUATION OF ImHS -PART ONE-

The analysis of performance on this article broads 9 algorithms in total of which 4 are traditional meta heuristics and 5 are variations of the original HS algorithm. A total of 900 runs were performed, of which 30 tests were selected to find the optimal solution for each case testing with 200 and 300 mechanisms. This produce 3,600 experiments in total. The tests considered are described in Table 4, with four cases: Case 1 with 200 mechanisms, Case 1 with 300 mechanisms, Case 2 with 200 mechanisms and Case 2 with 300 mechanisms.

This section is made up of 3 parts:

A. **Comparative analysis** is performed between ImHS and the 4 traditional meta heuristics, such as ABC, ACOR, JADE and JAYA.

B. The **tuning parameters** of each algorithm are provided.

C. The **optimum solutions** in tables of ImHS against the 4 HS variations are shown.

A. **COMPARATIVE ANALYSIS OF ImHS VS. TRADITIONAL METAHEURISTICS**

Once the ImHS tuning parameters have been established and the configuration of the experiments established, each of the values of mechanism variables is recorded, only for the case of the optimal solution for both Case 1 and Case 2. These variables are: $r_1$, $r_2$, $r_3$, $r_4$, $r_{cx}$, $r_{cy}$, $\theta_0$, $x_0$, $y_0$. In addition, parametric statistics such as average, maximum, minimum, standard deviation, successful runs of the algorithm and computational time expressed in seconds are also measured.

These variables are compared with the variables obtained in the state of the art of four metaheuristic algorithms against ImHS with $M = 200$ and against ImHS $M = 300$. These algorithms also solve the problems posed as Case 1 and Case 2 on this research. The algorithms against which ImHS is compared are: ABC, ACOR, JADE and JAYA. These comparatives are shown in Tables 5 and 6. Finally, the best mechanism achieved for **Case 1, with ImHS-200** is showed in Figure 6. By the other hand, the best mechanism achieved for **Case 2, with ImHS-200** is showed in Figure 7. The values of each variable optimized can be viewed also in Table 5 and Table 6.

In Table 5 it is observed that the minimum reached for the optimum of the objective function for Case 1 with ImHS with $M = 200$ is 0.0087591, resulting improving at JAYA, JADE, ACOR and ABC. On the other hand, ImHS with
TABLE 5. Comparative Results for Case-1: 5 Traditional Metaheuristics Vs. ImHS.

| Parameters | ABC       | ACOR      | JADE      | JAYA      | ImHS 200  | ImHS 300  |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|
| r1         | 56.2379   | 59.4200   | 39.8175   | 60.0000   | 49.334    | 56.2514   |
| r2         | 17.6238   | 14.0825   | 15.3898   | 24.3504   | 12.095    | 23.3435   |
| r3         | 55.1778   | 58.6327   | 17.9550   | 27.7573   | 35.234    | 29.5153   |
| r4         | 38.4697   | 57.1413   | 38.0480   | 60.0000   | 38.446    | 51.1769   |
| r_{ex}     | 60.0000   | 58.2705   | -42.0303  | 60.0000   | 36.866    | 44.2492   |
| r_{gy}     | 32.2899   | -49.3059  | -33.4336  | 60.0000   | 26.466    | 19.8608   |
| \theta_0   | 60.0000   | 60.0000   | -6.0344   | 60.0000   | 0.75883   | 0.316     |
| z_{0}      | -19.2212  | -3.1415   | 9.4786    | -0.6534   | 47.333    | 17.893    |
| y_{0}      | 65.5092   | 36.5289   | 39.5197   | 0         | 0.5213    | -20.4369  |
| mean       | 2.432742  | 1.576774  | 0.236152  | 32.03521  | 0.728746  | 0.6639    |
| max        | 10.23716  | 2.523341  | 0.348935  | 115.4545  | 2.766000  | 3.3862    |
| min        | 0.290407  | 0.752969  | 0.045676  | 7.523293  | 0.0087591 | 0.0391    |
| std        | 2.715589  | 0.575928  | 0.106758  | 43.25817  | 0.703190  | 0.8786    |
| Success    | 13        | 11        | 8         | 8         | 30        | 30        |
| Computational time (seconds) | 836 | 638 | 2664 | 672 | 2628 | 479.142 |

TABLE 6. Comparative Results for Case-2: 5 traditional metaheuristics Vs. ImHS.

| Parameters | ABC       | ACOR      | JADE      | JAYA      | ImHS 200  | ImHS 300  |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|
| r1         | 56.4909   | -         | -         | -         | 79.889    | 79.6286   |
| r2         | 30.6931   | -         | -         | -         | 9.9888    | 10.2795   |
| r3         | 71.8633   | -         | -         | -         | 54.388    | 79.9695   |
| r4         | 67.2668   | -         | -         | -         | 73.443    | 46.0852   |
| r_{ex}     | -49.9770  | -         | -         | -         | 10.938    | 19.9083   |
| r_{gy}     | -66.1093  | -         | -         | -         | 0.33462   | 0.475     |
| \theta_0   | 80.0000   | -         | -         | -         | 2.1654    | 2.7709    |
| z_{0}      | 20.8109   | -         | -         | -         | 20.781    | 29.4214   |
| y_{0}      | 185.0648  | -         | -         | -         | 11.612    | 14.6776   |
| mean       | 25.49955  | -         | -         | -         | 4.7537    | 5.3632    |
| max        | 44.18664  | -         | -         | -         | 12.392    | 75.1193   |
| min        | 4.525837  | -         | -         | -         | 0.072812  | 0.6937    |
| std        | 13.23306  | -         | -         | -         | 3.995540  | 41.5092   |
| Success    | 15        | 0         | 0         | 0         | 30        | 30        |
| Computational time (seconds) | 966 | 744 | 2046 | 82 | 2826 | 1436.148 |

$M = 300$ reaches an optimum of 0.0391, exceeding JAYA, JADE, ACOR and ABC. The number of successful executions with feasible solutions was 30, that is, it exceeds the 4 traditional metaheuristics in terms of feasibility and successful runs.

To evaluate the performance of the Harmony Search Improved algorithm (ImHS), it was executed 30 times to select 30 runs of the problem defined in Case 1, and Case 2. More specifically, 1,800 experiments were made where the number of mechanisms ($M$) was equal to 200. Another set of 1,800 experiments was also tested by varying the number of mechanisms of 300 for the $M$ parameter. Therefore, the total of experiments performed were 3,600.

On the other hand, the computational cost in terms of seconds, is 2628 seconds for ImHS 200 and 479, 142 for ImHS 300. As regards the experiments carried out for Case 2, in the Table 4, the reached targets of ImHS $M = 200$ is 0.072818, surpassing and ABC. As for the value of ImHS with $M = 300$, its optimal value is 0.6937, which exceeds ABC. It is also possible to observe that in terms of parametric statistics, the average of the optimal objective solutions is 41.5092, which is above the average solutions of the four traditional meta heuristic algorithms.

As for the execution time in seconds, a total of 1436.148 seconds is achieved, exceeding in time to JAYA and JADE. It is also noted that the standard deviation for ImHS 200 is 3.995540, which exceeds ABC.

As for the feasible solutions, all algorithms of the state of the art are surpassed, with 30 feasible executions.

The contribution of this section consists of having demonstrated the competitiveness capacity of ImHS against four traditional meta heuristic algorithms based on 2 case studies for four-bar mechanisms. This is demonstrated with previous numerical results.
Regarding parametric statistics, it is important to mention that in the case of the average, ImHS with $M = 200$ was 0.728746 and ImHS with $M = 300$ was 0.6639, which means that ImHS its located above three algorithms: JAYA, ACOR and ABC. This indicates that the stability for the search of the best solution was minimal with respect to these three algorithms, which indicates that it is also more efficient finding solutions within the exploitation process.

As for the statistics of the maxima, ImHS with $M = 200$, it was obtained a value of 2.7669 and for ImHS with $M = 300$ it was 2.3862, which indicates that ImHS only exceeds ABC. For the case of the standard deviation, it can be observed in Table 6 that ImHS with $M = 200$ is 0.7031 and the case of ImHS with $M = 300$ is 0.8786, above of JAYA and ABC. This indicates that ImHS is an important and competitive opponent in at least 2 cases of study, which also surpasses the performance of four algorithms (ABC, ACOR, JADE and JAYA) with 0.008759. The Improved Harmony Search algorithm also exceeds two more algorithms (ABC and JAYA) with a standard deviation of 0.703190, this imply that generation of solutions for ImHS algorithm has a better distribution than other algorithms.

In the same direction, for the case of study 2, ImHS, with an optimum in the objective function of 0.072812, exceeds the same four algorithms that for Case 1. So, ImHS exceeds two algorithms with an average of 4.7537. ImHS also exceeds three algorithms with a minimum of 12, 392. Regarding the maximum and standard deviation that exceeds three algorithms with 12, 392 and 3.955540 respectively.
TABLE 9. Comparative Results for Case-1: Three versions of HS for 200 and 300 Vs. ImHS.

| Parameters | OHS, M=200 | MHS, M=200 | ImHS, M=200 | OHS, M=300 | MHS, M=300 | ImHS, M=300 |
|------------|------------|------------|-------------|------------|------------|-------------|
| r1         | 76.2733    | 59.4711    | 49.334      | 55.4541    | 52.6481    | 56.2514     |
| r2         | 37.6367    | 14.7140    | 12.095      | 33.2641    | 23.1959    | 21.3435     |
| r3         | 75.5560    | 41.2373    | 35.234      | 43.2929    | 57.6955    | 29.5153     |
| r4         | 43.1809    | 51.2875    | 38.446      | 56.1518    | 45.5062    | 51.1769     |
| r_cx       | 64.4191    | 46.1803    | 36.866      | 60.9687    | 5.3542     | 44.2492     |
| r_cy       | 59.0761    | 30.1487    | 26.466      | 42.5767    | 51.1854    | 19.8605     |
| θ_0        | 0.8402     | -3.4767    | 0.75083     | 0.8535     | 3.6073     | 0.316       |
| π_0        | 64.8340    | 94.9614    | 47.333      | 75.1242    | -8.3540    | 17.893      |
| y_0        | -44.1788   | 55.0460    | 0.5213      | -50.1698   | 38.8193    | -20.4364    |
| mean       | 11.1318    | 0.7839     | 0.728746    | 2.8^4      | 0.4920     | 0.6639      |
| max        | 75.9558    | 1.0817     | 2.766900    | 5.7^4      | 0.6028     | 3.3862      |
| min        | 7.8828     | 0.1234     | 0.0087      | 22.9049    | 0.1130     | 0.0391      |
| std        | 48.9937    | 0.2408     | 0.7031      | 8.7^4      | 0.1057     | 0.8786      |
| Success    | 7          | 30         | 30          | 30         | 30         | 30          |
| Computational time (seconds) | 2152.782  | 2178.9     | 2628        | 2300.19    | 2663.448   | 479.142     |

This demonstrate that the Improved Harmony Search algorithm is competitive against ABC, ACOR, JADE and JAYA in at least five analysis points including parametric statistics, in 2 study cases of four-bar linkage mechanisms.

B. TUNING PARAMETERS OF TRADITIONAL METAHEURISTICS

The parameters settings of four meta heuristics are detailed as follows:

1) Artificial bee colony (ABC): The number of food sources for employed bees is set to be \( n_p/2 \). A trial counter to visit a food source is 100.
2) Real-code ant colony optimization (ACOR): The parameters settings are \( q = 0.2 \) and \( \xi = 1 \).
3) Adaptive differential evolution with optional external archive (JADE): The parameters are self-adapted during an optimization process.
4) Jaya algorithm (JAYA): Parameters settings are not required.

C. FEASIBILITY AND CONVERGENCE OF ImHS VS. TRADITIONAL METAHEURISTICS

Finally, we perform an evaluation of each case of study for each variant in the four tests: (1) The test with case 1 and 200 possible mechanisms, (2) the test with case 1 and 300 possible mechanisms, (3) the test with case 2 and 200 mechanisms, and (4) the test with case 2 and 300 mechanisms solved by the algorithm of Harmony Search Improved (ImHS). Feasible solutions were found faster than with other algorithms. It is then observed that the mechanism of the case study 2 with a quantity of 200 tested mechanisms, found more quickly a stability in its harmonic memory. This indicates that the search area was more promising than in the other tests.

Is important to mention that Test 1: case 1 with 200 mechanisms, reaches an optimal solution faster than the others tests.

VIII. RESULTS AND EVALUATION OF ImHS -PART TWO-

The number of experiments comprised 5 variations of the original HS. A total of 30 runs were carried out, of which 1,800 tests were obtained to find the optimal solution in Case 1 and 1,800 for case 2, for a total of 3,600 experiments, for each ImHS run. But if it take into account that the 5 HS variation algorithms were implemented, it is has a total of 18,000 tests.

This section is made up of 3 parts:

A. In the first part of the results, a comparative analysis is performed between ImHS and 5 variations of HS: HS, IHS, ABHS, MHS and OHS, in terms of variables, and objective function.
B. In the second part are shown the tuning parameters for the 5 HS variations.
C. In the third part the feasibility and convergence graphs of ImHS against of the 5 variations of HS.
The comparative analysis between OHS, MHS and ImHS can be observed in Table 9 and Table 10.

The tuning parameters of ImHS are detailed in the Table 11. The tuning parameters of the five harmonic versions are explained in Section VII.B. Tuning Parameters of Variations of HS.

In previous tables, 4 tests were analyzed, varying the study case and the number of mechanisms. As can be seen in the results of the Table 7, the first 4 columns show the optimal results of the 4 versions of HS, and it is shown that ImHS is victorious over the others in the calculation of $min = 0.008759$ when $M = 200$.

However, in the case of $M = 300$, the one that shows the best performance in terms of calculating the minimum is ABHS with $min = 0.0226$. Regarding the experiments shown in the 8, it is again demonstrated that ImHS with $M = 200$ obtains an optimum of $min = 0.0728$. In the case of $M = 300$, original ImHS obtains best result with $min = 0.6937$ even better than OHS and MHS.

### B. TUNING PARAMETERS OF VARIATIONS OF HS

Tuning parameters of the improve harmony search algorithm are the following: 50,000 generations (g), 50 harmonies (N), 0.9 of harmonic memory acceptance factor to the next generation ($r_{accept}$), 0.2 of pitch adjustment ($r_{pa}$) and 0.9 as smart adjustment factor ($f_i$). The performance evaluation of the Harmony Search Improved algorithm imply 30 execution runs both to test the problem defined in case 1 and to test the problem defined in case 2. The tuning parameters are detailed in the Table 11.

### TABLE 11. Tuned values for Harmony Search Improved algorithm.

| Description                  | Parameters | Values |
|------------------------------|------------|--------|
| Number of generations (g)    | N          | 50,000 |
| Size of harmonic memory (N)   | $r_{accept}$ | 0.9    |
| Percentage of new solutions accepted | $r_{pa}$   | 0.2    |
| Harmonic memory tone          | $f_i$      | 0.9    |

Regarding the tuning of the parameters of the ImHS algorithm, it is important to mention that each of them was chosen in function of performance of the algorithm, taking into account the set of 50 harmonic memories for each of the generations of search. That meaning, that for each run 50 times the value of the objective function was analyzed. If a nearest value of the objective function it arosed in any of the vectors of the harmonic memory, then the value of the generations are increased. If the value of the objective function did not appear, then the number of generations was increased. Generations was increased to give greater freedom to the exploration process of solutions within the search.
space. Once it was possible to obtain a relatively close value or similar to the objective function in a range of tenths, then the number of generations was set \( g = 50,000 \) and then the value of the tone was varied \( (r_{\text{par}}) \), which was established in the lowest value, that is, with a value of 0.2, in this way it was ensured that the algorithm did not jump abruptly within the solution space when it found a feasible solution, refining the exploitation process. From the beginning it was decided to have a value of \( r_{\text{accept}} = 0.9 \), since it was desired that the factor of new more suitable solutions, were accepted in a maximum of 90 \%.

The tuning values for each algorithm are the following:

- The values for HS, IHS, ABHS, MHS and ImHS are the same for all, and are shown in Table 11.
- The MHS (Modified Harmony Search) algorithm uses an additional parameter called influence factor. This value depends on the number of variables in the problem. The author del algorithmo en [15], [33] suggests that the value for the influence factor be assigned with level 1 (value of 0.1) for 10 variables and level 2 for 20 variables. However, for Case 1 and 2 of the 4-bar mechanisms, the number of variables of the problem fluctuate between 9 and 10. But tests were carried out where the value of 1 was not sufficient, so it was decided to assign the value of level 2 (it is say 0.2) to try to make the results as competitive as ImHS. This with the intention of obtaining the maximum performance from MHS.
- The values for OHS recommended by the [14], [36], to achieve balanced convergence are shown in Table 12. The parameters for OHS imply additional tuning, the jump value is of 0.8, the minimum adjustment value is of 0.45, the maximum adjustment value is 0.98, the minimum value of bandwidth is \( 6 \times 10^{-6} \), the maximum value is dynamically subject to Expression 33. An additional parameter is adjustment rate dynamically computed by Expression 35.

The computation value for bandwidth in the case of the OHS algorithm, is a static expression provided in Equation 33:

\[
bw_{\text{max}} = \frac{1}{20(par_{\text{max}} - par_{\text{min}})}
\]  

(33)

where the value computed by \( bw_{\text{max}} \) is taken to compute the dynamical value of bandwidth by generation as is showed in Expression 34:

\[
bw(\text{gen}) = bw_{\text{min}} + e \left( \frac{bw_{\text{min}}}{bw_{\text{max}}} \right)^{\ast \text{gen}}
\]  

(34)

where \( NI \) is the number of individuals or variables in the four bar mechanism problem and \( \text{gen} \) is the number of generations.

In the case of the Expression to compute the dynamical value of \( par_{\text{gen}} \), the Equation 35 is employed:

\[
par_{\text{gen}} = par_{\text{min}} + \frac{par_{\text{max}} - par_{\text{min}}}{NI} \ast \text{gen}
\]  

(35)

where \( par_{\text{min}}, par_{\text{max}} \), and \( \text{gen} \) are already stated in Table 12.

It is interesting to mention that in the case of OHS and MHS, these are algorithms that consist of more recent versions of HS whose modifications are based on improvements or modifications to the parameters. Specifically to improvements in the bandwidth \( bw \).

The programming of the HS, IHS, ABHS, MHS and OHS algorithms was carried out under the following conditions:

1. They are given uniformity in programming with the intention of:
   - 2) Increase convergence capabilities
   - 3) Optimal values are assigned according to the authors
   - 4) All optimal solutions are selected through Deb Rules
   - 5) A tone limit check is added to all

### IX. FEASIBILITY AND CONVERGENCE OF ImHS VS. OTHER VARIATIONS OF HS

In the Figure 8 (a) and (b), a feasibility and convergence graph of the observed performance of the ImHS algorithm is displayed during a cumulative count from the first generation up to the generation 50,000. This graph shows the 4 tests carried out: for case 1 with 200 mechanisms (blue line), for case 1 with 300 mechanisms (cyan line), case 2 with 200 mechanisms (red line) and case 2 with 300 mechanisms (pink line).

Similarly, the convergence for the 4 cases for ImHS, HS, IHS and ABHS is shown in the Figure 8 (c) and (d). This convergence graph shows the value of each of the optimal solutions that are found during the 50,000 generations iterations. It is observed that from the first generations optimal solutions are obtained until practically the convergence becomes very close to 0.

In Figure 8 (e) and (f), is shown the feasibility and convergence of ImHS, OHS and MHS. The feasibility of ImHS is outstanding versus the other 2 versions of HS. The case considered for comparison, is case 1 of the 4-bar mechanisms. The convergence of ImHS is faster than OHS and MHS.

The third row of Figure 8 shows the feasibility and convergence of the latest tested algorithms: OHS, MHS and ImHS. In the feasibility graph Figure 8 (e) it can be seen how OHS

---

### TABLE 12. Tuned values for Opposite Harmony Search Algorithm.

| Description                  | Parameters | Values     |
|------------------------------|------------|------------|
| Number of generations        | \( g \)    | 50,000     |
| Size of harmonic memory      | \( N \)    | 50         |
| Percentage of new            | \( r_{\text{accept}} \) | 0.9        |
| solutions accepted           | \( j \)    | 0.8        |
| Jump value                   | \( par_{\min} \) | 0.45       |
| Minimum Adjustment           | \( par_{\max} \) | 0.98       |
| Maximum Adjustment           | \( bw_{\min} \) | \( 6 \times 10^{-6} \) |
| \( bw \) Minimum Value       | \( bw_{\max} \) | \( \ast \text{see (33)} \) |
| Adjustement rate             | \( par_{\text{gen}} \) | \( \ast \text{see (35)} \) |
hardly achieves feasible solutions during the 50,000 iterations. On the other hand, MHS achieves a feasibility that is even less than that obtained by ImHS. The convergence graph between OHS, MHS and ImHS shows that OHS hardly converges, unlike MHS. But it is also observed that ImHS shows the best convergence of the three (see Figure 8 (f)).
In the feasibility graph Figure 8 (e) it can be seen how OHS hardly achieves feasible solutions during the 50,000 iterations. On the other hand, MHS achieves a feasibility that is even less than that obtained by ImHS. But it is also observed that ImHS shows the best feasibility and convergence of the three (see Figure 8 (f)).

**X. ANALYSIS OF RESULTS**

The best mechanisms obtained by ImHS are shown below. The Figure 9 (a) shows the optimal mechanism obtained for Case 1 for problems with 4 bars, where the mechanism must make a circular journey around nine points, with 9 variables to solve objective function. In the Figure 9 (b) the second mechanism of the Case 2 in four-bar problem is shown, where the end gripper must make a route of practically one line. This problem is not trivial to solve by traditional techniques, and was even a challenge for the other versions of ImHS. Both problems were solved with an error very close to zero with the following values: $f_{obj, case1} = 0.0087$ and $f_{obj, case2} = 0.0728$.

In Figure 9 (c) and (d) is shown with colored circles, the relationship between performance and feasible solutions on a general Cartesian axis represented graphically. This figure shows the position that ImHS has compared to traditional algorithms and the rest of HS versions. That is, their performance is represented in a better position in terms of feasibility. Where the red circles represents null solutions.
or high errors regarding the optimum found. The yellow circles indicate average performance and solutions with a lower error, but with unsatisfactory solutions. In the case of the green circles imply those algorithms with high values of feasible solutions and very small errors in their optimal solutions very close but not exceeding ImHS for study case 1 in mechanism of four bars.

In Figure 9 (d) the performance of the ten algorithms (4 analyzed and 6 implemented) in this research is observed. In this scheme, a general relationship is displayed between all algorithms, the x-axis shows the performance with respect to the algorithm and the y-axis shows a scale of the optimum found for the problem of case 2 on mechanisms four-bar. In this figure, the red circles include algorithms those low performance and lower solutions were obtained. In the yellow circle, the algorithms with mean solutions were included with many feasible solutions, but its errors are still in the order of units or tens. The green circles show the algorithms that observed the best performances. The best solutions are obtained with very small errors for both cases.

In Table 13, it is possible to observe a summary of the optimal solutions reached by each of the ten algorithms analyzed in this investigation. In case 1 with 200 mechanisms and with 300 mechanisms ImHS is victorious with a minimum of 0.0087591 and 0.0728 respectively. In case 2 with

**TABLE 13. Optimal solutions of ten meta heuristic algorithms.**

| Algorithm | Case 1 M=200 | Case 1 M=300 | Case 2 M=200 | Case 2 M=300 |
|-----------|-------------|-------------|-------------|-------------|
| ABC       | 0.290407*   | -*          | 4.525837*   | -*          |
| ACOR      | 0.752969*   | -*          | -*          | -*          |
| JADE      | 0.045676*   | -*          | -*          | -*          |
| JAYA      | 7.523293*   | -*          | -*          | -*          |
| HS        | 0.0503      | 0.0399      | 2.8040      | 0.1618      |
| IHS       | 0.0409      | 0.0340      | 13.4655     | 17.3623     |
| ABHS      | 0.0371      | 0.0226      | 16.7777     | 27.6333     |
| OHS       | 2.8040      | 22.9049     | 107.9662    | 123.7637    |
| MHS       | 13.4655     | 0.1130      | 26.9900     | 22.6565     |
| ImHS      | **0.0087591** | 0.0391      | **0.0728**  | 0.6937      |

In Table 13, it is possible to observe a summary of the optimal solutions reached by each of the ten algorithms analyzed in this investigation. In case 1 with 200 mechanisms and with 300 mechanisms ImHS is victorious with a minimum of 0.0087591 and 0.0728 respectively. In case 2 with
300 mechanisms, it is observed that the best algorithms were ABHS and HS with 0.0226 and 0.1618 respectively. That is, the proposed algorithm is an effective solution in case 1 against ten algorithms, but for case 2, it is not possible to establish an absolute winner.

Finally we show a summary of the objective functions graphically to complement the previous Table in Figure 10.

### XI. CONCLUSION

It is well known that the problem of the mechanism of four bar handled in this investigation with two cases of study, is one of the problems particularly complicated and particularly proven within the state of the art. The case studies handled involve a high complexity due to the number of variables, however, a simplification has been handled in the case of the angles; which in turn simplifies the optima of the entire mechanism in both case 1 and case 2. Although in each study case 9 variables are handled and evaluated within an objective function. The objective function was iterated in 50,000 generations, each of the 50 harmonic memoirs, along 30 executions, and where each execution takes 87 seconds approximately, it has a complexity like the one shown in (36).

\[
O(n) = n\text{var} \times g \times \text{memory} \times \text{runs} \quad (36)
\]

where \(n\text{var}\) is the number of variables, \(g\) is the number of generations, \(\text{memory}\) is the size of the harmonic memory and \(\text{runs}\) is the number of executions. That is, the number of evaluations of the objective function is 6,758, which was executed in Matlab R2014a (8.3.532).

Regarding the ImHS algorithm, it is important to emphasize that the comparison made in this research it obtains that ImHS surpass 9 algorithms en Case 1 and 9 algorithms in case 2 in terms of improving the optimum. On the other hand, it can be seen that convergence and feasibility is above of most algorithms.

However, in the feasibility and convergence graphs, ImHS remains the winner, since the feasibility is reached faster and convergence is also achieved in fewer generations with respect to HS, IHS or ABHS for case 1.

From all of the above, we can say that ImHS is victorious with respect to 9 algorithms of the state of art in terms of reaching the optimal solution as is browsed in Figure 10.

### REFERENCES

[1] S. Sriessongsom and S. Bureerat, “Four-bar linkage path generation through self-adaptive population size teaching-learning based optimization,” Knowl.-Based Syst., vol. 135, pp. 180–191, Nov. 2017.

[2] E. A. Portilla-Flores, A. Sanchez-Marquez, L. Flores-Pulido, E. Vega-Alvarado, M. B. Calva Yanez, J. A. Aponte-Rodríguez, and P. A. Nino-Suarez, “Enhancing the harmony search algorithm performance on constrained numerical optimization,” IEEE Access, vol. 5, pp. 25759–25780, 2017.

[3] A. LaTorre, S. Muelas, and J.-M. Pena, “Large scale global optimization: Experimental results with MOS-based hybrid algorithms,” in Proc. IEEE Congr. Evol. Comput., Jun. 2013, pp. 2742–2749.

[4] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Eng. Faculty, Erciyes Univ., Kayseri, Turkey, Tech. Rep. TR06, Oct. 2005.

[5] E. Mezura-Montes and O. Cetina-Domínguez, “Empirical analysis of a modified artificial bee colony for constrained numerical optimization,” Appl. Math. Comput., vol. 218, no. 22, pp. 10943–10973, Jul. 2012.

[6] M. Close, G. Dorigo, and D. Caro, “The ant colony optimization meta-heuristic,” in New Ideas in Optimization. New York, NY, USA: McGraw-Hill, 1999, pp. 11–32.

[7] A. Kaveh and S. Talatabari, “An improved ant colony optimization of planar steel frames,” Eng. Struct., vol. 32, pp. 864–873, Oct. 2010.

[8] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 945–958, Oct. 2009.

[9] R. Venkata Rao, “Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems,” Int. J. Ind. Comp. Manag., vol. 5, no. 3, pp. 19–34, 2014.

[10] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. N. Suganthan, “Flexible job-shop rescheduling for new job insertion by using discrete jaya algorithm,” IEEE Trans. Syst. Man, Cybern., vol. 49, no. 5, pp. 1944–1955, May 2019.

[11] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: Harmony search,” Simulation, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[12] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving problems,” Appl. Math. Comput., vol. 188, no. 2, pp. 1567–1579, 2007.

[13] O. M. Alia and R. Mandava, “The variants of the harmony search algorithm: An overview,” Artif. Intell. Rev., vol. 36, no. 1, pp. 49–68, Jun. 2011.

[14] A. Banerjee, V. Mukherjee, and S. P. Ghosal, “An opposition-based harmony search algorithm for engineering optimization problems,” IET Syst. Trans. Eng., vol. 5, no. 85, pp. 100–101, 2014, doi: 10.1016/j.ijepes.2012.08.021.

[15] J. Yi, L. Gao, X. Li, and J. Gao, “An efficient modified harmony search algorithm with intersect mutation operator and cellular local search for continuous function optimization problems,” Int. J. Speech Technol., vol. 44, no. 3, pp. 725–755, Apr. 2016.

[16] C. V. Camp, B. J. Bichon, and S. P. Stovall, “Design of steel frames using ant colony optimization,” J. Struct. Eng., vol. 131, no. 3, pp. 369–379, Mar. 2005.

[17] C. C. Ramos, A. N. Souza, G. Chiachia, A. X. Falcao, and J. P. Papa, “A novel algorithm for feature selection using harmony search and its application for non-technical losses detection,” Comput. Electr. Eng., vol. 37, no. 6, pp. 886–894, 2011.

[18] L. Wang and L.-P. Li, “An effective differential harmony search algorithm for solving the non-convex economic load dispatch problems,” Int. J. Electr. Power Energy Syst., vol. 44, no. 1, pp. 832–843, Jan. 2013, doi: 10.1016/j.ijepes.2012.08.021.

[19] E. Cuevas, “Block-matching algorithm based on harmony search optimization for motion estimation,” Int. J. Speech Technol., vol. 39, no. 1, pp. 165–183, Jul. 2013, doi: 10.1007/s10149-012-0403-7.

[20] F. Zhao, Y. Liu, C. Zhang, and J. Wang, “A self-adaptive harmony PSO search algorithm and its performance analysis,” Expert Syst. Appl., vol. 42, no. 2, pp. 7436–7442, Nov. 2015, doi: 10.1016/j.eswa.2015.05.035.

[21] W. L. Xiang, M. Q. An, Y. Z. Li, R. C. He, and J. F. Zhang, “An improved global best harmony search algorithm for faster optimization,” Expert Syst. Appl., vol. 41, no. 31, pp. 5788–5803, 2014.

[22] E. Valian, S. Tavakoli, and S. Mohanna, “An intelligent global harmony search approach to continuous optimization problems,” Appl. Math. Comput., vol. 232, pp. 670–684, Apr. 2014.

[23] M. O. Alia, M. A. Al-Betar, R. Mandava, and A. T. Khader, “Data clustering using harmony search algorithm,” in Swarm, Evolutionary, and Memetic Computing Berlin, Germany: Springer, 2011, pp. 79–88.

[24] M. R. Razfar, R. Farshbaf Zinati, and M. Haghshenas, “Optimum surface roughness prediction in face milling by using neural network and harmony search algorithm,” Int. J. Adv. Manuf. Technol., vol. 52, nos. 5–8, pp. 487–495, Feb. 2011.

[25] S. O. Jegede, “Improved harmony search algorithms for sizing optimization of truss structures,” Comput. Struct., vols. 92–93, pp. 229–241, Feb. 2012.

[26] S. M. Ashrafi and A. B. Dariane, “A novel and effective algorithm for numerical optimization: Melody Search (MS),” in Proc. 11th Int. Conf. Hybrid Intell. Syst. (HIS), Melaka, Malaysia, Dec. 2011, pp. 109–114, doi: 10.1109/HIS.2011.6122089.

[27] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, “An intelligent tuned harmony search algorithm for optimization,” Inf. Sci., vol. 196, no. 1, pp. 42–47, 2012, doi: 10.1016/j.ins.2011.12.035.
EDGAR ALFREDO PORTILLA-FLORES was born in Tlaxcala, Mexico, in 1968. He received the B.Sc. degree in electronics engineering from Universidad Autónoma Metropolitana, Mexico, in 1992, the M.Sc. degree in mechanical engineering from the Instituto Tecnológico de Puebla, Mexico, in 2002, and the Ph.D. degree in electrical engineering from the Centro de Investigación y Estudios Avanzados, Mexico, in 2006. He completed the Postdoctoral Residency at the Universidad de Estadual de Campinas, Brazil, in 2012. He is currently a full-time Research Professor with the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City. His research interests include optimum design of mechatronic systems and the application of bio-inspired algorithms to the solution of engineering problems. He is a member of the National System of Researchers of Mexico.

ERIC SANTIAGO-VALENTÍN was born in Tlaxcala, Mexico, in 1990. He received the B.Sc. degree in computer systems from the Instituto Tecnológico de Tlaxaco, Mexico, in 2013, and the M.Sc. degree in computer technology from the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City, Mexico, in 2016, where he is currently pursuing the Ph.D. degree in robotic and mechatronic systems engineering. His research interests include bio-inspired algorithms and their applications to engineering problems.

EDUARDO VEGA-ALVARADO was born in Mexico City, Mexico, in 1965. He received the B.Sc. degree in communications and electronics engineering from the Escuela Superior de Ingeniería Mecánica y Eléctrica, Mexico, in 1992, the M.Sc. degree in digital systems from the Centro de Investigación Tecnológica en Computación, Mexico, in 1996, and the Ph.D. degree in computational and electronic systems from the Universidad Autónoma de Tlaxcala, Mexico, in 2017. He is currently a full-time Researcher and a Professor with the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City. His research interests include optimization with metaheuristics, and the development of hybrid algorithms and its application on engineering problems.

leticia-flores-pulido was born in Puebla, Mexico. She received the Ph.D. degree in computer sciences from the Universidad de las Américas Puebla, with the research titled: Modeling of Data Segmentation for Visual Information Retrieval Systems. She has been a Research Professor with Level “C” within the Facultad de Ingeniería y Tecnología, Universidad Autónoma de Tlaxcala, since 2001. Together with colleagues and students, she has researched in artificial intelligence, neural networks, pattern recognition, image processing, and heuristic algorithms.
MARÍA BÁRBARA CALVA YÁÑEZ was born in Tlaxcala, Mexico, in 1983. She received the B.Sc. degree in electronic systems engineering from the Universidad Autónoma de Tlaxcala, Mexico, in 2010, the M.Sc. degree in computational technologies from the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico, in 2013, and the Ph.D. degree in robotic and mechatronic systems engineering from the Escuela Superior de Ingeniería Mecánica y Eléctrica, Azcapotzalco of the Instituto Politécnico Nacional, in 2018. Her research interests include optimum design of mechatronic systems and the application of bio-inspired algorithms to engineering problems.

PAOLA ANDREA NIÑO-SUÁREZ received the B.Sc. degree in electronics engineering from Universidad Antonio Nariño, Colombia, in 1995, the M.Sc. degree in electrical engineering biomedical specialty from the Universidad de los Andes, Colombia, in 1998, and the Ph.D. degree in electrical engineering from the Centro de Investigación y Estudios Avanzados, Instituto Politécnico Nacional, Mexico, in 2006. She is currently a Research Professor with the Graduate Studies and Research Section, School of Mechanical and Electrical Engineering-Azcapotzalco, Instituto Politécnico Nacional. Her research interests include mechatronics and mobile robotics.