

A Dynamic Quality-Based Harmony Search Algorithm for Solving Constrained Engineering Optimization Problems

Ali Kattan

Department of Information Technology, Ishik University, Erbil, Iraq
kattan@ieee.org

Abstract—Meta-heuristic algorithms overcome the computational drawbacks of the existing numerical methods where they are commonly used to solve constrained engineering problems and find an optimum solution. Solving this kind of problems is considered of high value in many engineering and manufacturing processes. The author have recently published a dynamic and self-adaptive meta-heuristic that is based on the harmony search algorithm. The method has the advantage of dynamically setting optimization parameters based on quality measures that are computed during the optimization process. Testing showed superiority in solving continuous optimization problems with high dimensionality. In this work, the same method is applied with minor modifications to solve constrained engineering problems whereby the feasible search-space is shrunk due to the existing constraints making the problem more difficult. The results obtained are close to those achieved by some other recent meta-heuristic methods. However, there is a room for improvement.

Keywords—component; meta-heuristic; optimization; constrained problems; harmony search

I. INTRODUCTION

Solving constrained optimization problems is considered of high value in engineering and manufacturing processes. The constraints shrink the feasible search-space and, instead of making the problem easier, it makes it more difficult because feasible solutions can be hard to find [1]. Numerical methods, such as gradient-based search, may become difficult and unstable when the objective function and constraints have multiple or sharp peaks [2]. Stochastic algorithms such as evolutionary algorithms (EA) have shown to be effective optimization techniques since no conditions or mathematical properties of the objective function are required [3].

To solve constrained problems and find global optimum solutions, many different meta-heuristics that are population-based have been proposed. These meta-heuristics deploy stochastic techniques such as differential evolution (DE) [1, 4], particle swarm optimization (PSO) [5, 6] and harmony search (HS) [2, 7]. Of particular interest is the recent HS approach. The authors have recently published a dynamic and self-adaptive harmony search algorithm, referred to as dynamic self-adaptive harmony search (DSHS), for solving continuous optimization problems [8]. The DSHS algorithm utilized two quality measures to drive the optimization process and does not require any manual setting to be investigated before starting optimization. In addition, and in

contrast to many HS-based approaches, the DSHS optimization parameters are completely decoupled from the iteration count and changes dynamically based on the quality of the current solutions. In this paper, the author have adopted the DSHS algorithm with minor modifications and applied it on a commonly used constrained engineering optimization problem. Results indicate that the method have successfully obtained results that are similar to those obtained by other recent techniques.

The paper is organized as follows: Section 2 is the related works in which we present some of the recent meta-heuristics for solving constraint optimization engineering problems inclusive of some HS-based methods. Section 3 presents the proposed method. Section 4 presents a numerical example of an engineering problem along with the experimentation details, the results obtained and the discussion. Finally, in Section 5 some conclusions are drawn.

II. RELATED WORKS

Optimization for several engineering design problems involves finding \mathbf{x} which optimizes a $f(\mathbf{x})$. This is considered as a general nonlinear programming problem where $f(\mathbf{x})$ is the objective function and $\mathbf{x} \in \mathfrak{R}^n$ is an n -dimensional vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ with mixed variable such as integer, discrete and continuous. Each x_k , $k=1, \dots, n$ is bounded by a lower and upper limits $L_k \leq x_k \leq U_k$. The problem must adhere to a number of equality and inequality constraints, $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$. In both cases, constraints could be linear or nonlinear. The optimization problem is formalized in equation (1) where m and p represents the total number of equality and inequality constraints respectively.

Optimize:

$$f(x) \text{ subject to } \begin{cases} h_i(\bar{x}) = 0 & i = 1, 2, \dots, m \\ g_j(\bar{x}) \leq 0 & j = 1, 2, \dots, p \end{cases} \quad (1)$$

Several meta-heuristics have been proposed to solve such constrained problems in order to find global optimum solutions. A common technique for constraint handling is the adoption of penalties. A penalty function is applied to unfeasible solutions to generate a poor function value. If the solution is feasible then the penalty is 0. In this way, the unfeasible solutions can be removed from the population in the next iteration of the algorithm. Using this approach, the constrained problem is treated as an unconstrained one [1, 2, 4, 6, 7]. Equation (2) describes a common penalty function used for minimization problems. In equation (2) $F(x)$ represents penalty function, $f(x)$ is the objective function, $h(x)$

and $g(x)$ are the equality and inequality constraints as described in (1). Notice that since it is hard to find feasible solutions that satisfy the equality constraint $h_i(x) = 0$ exactly, it is necessary to convert it into the inequality constraint $|h_i(x)| - \varepsilon \leq 0$. Here, ε is a small positive constant, and it represents the tolerated violation. In addition, λ is a large positive constant, which imposes penalty on unfeasible solutions, and it is defined as penalty coefficient [4].

$$\min F(x) = f(x) + \lambda \left[\sum_{j=1}^p \max\{0, g_j(x)\}^2 + \sum_{i=1}^m \max\{0, |h_i(x)| - \varepsilon\}^2 \right] \quad (2)$$

The DE technique, originally introduced in 1995 [9], was adopted for solving constrained engineering problems [1, 4]. DE is a floating-point encoding populational meta-heuristic similar to traditional EAs. However, the generation of candidate solutions is made by a greedy selection scheme, after choosing more than two individuals of the population [1].

A more recent DE-based approach, referred to as novel modified differential evolution (NMDE) algorithm is proposed in [4]. The NMDE algorithm introduces some self-adaptive capabilities to set some of the algorithm parameters, namely the scale factor and crossover rate, in order to get rid of stagnation. Each solution in the proposed technique has its own scale factor and crossover rate instead of using pre-fixed value for all solutions before starting the optimization process.

The PSO, originally introduced in 1995 [10], is yet another meta-heuristic that was adopted for solving constrained engineering problems. In the classical PSO algorithm each population member is seen as a particle, and each particle is a potential solution to the problem under analysis. Each particle in PSO has a randomized velocity associated to it, which moves through the space of the problem. The quantum-behaved PSO method (QPSO) is a more recent variant of the classical PSO [6]. QPSO combines an approach inspired from quantum mechanics theories whereby it applies Gaussian mutation operators instead of random sequences as in the classical PSO. The method performs better as it prevents premature convergence to local minima.

The HS algorithm is more recent meta-heuristic originally introduced in 2001 [11]. It is a relatively young meta-heuristic that was inspired from the improvisation process of musicians and was later used for engineering optimization problems with continuous design variables. A complete description of the algorithm can be found in [12]. The HS algorithm is an evolutionary stochastic global optimization method that is similar in concept to other methods such as DE and PSO in terms of combining the rules of randomness to imitate the process that inspired it [8].

HS performance and convergence to a good solution is bound to the settings of HS optimization parameter prior to starting the optimization process [3]. These optimization parameters include the pitch-adjustment rate (PAR) and the bandwidth (BW). Many self-adaptive HS algorithms were proposed to automatically tune and find the best settings for these optimization parameters in order to achieve better

results [2, 3, 7, 13]. However, many of these variants have also introduced new additional parameters, which in some cases would require an additional effort to set their initialization values manually depending on the problem considered prior to starting the optimization process. In addition, many of these variants have forced a relationship between the value of these parameters and the current optimization process iteration count imposing a linear monotonic change that is bounded by a maximum integer iteration (improvisation) value (MAXIMP). This value is usually selected subjectively by the user to indicate termination. Consequently, the selection of such integer value would affect the whole optimization process [8].

III. PROPOSED METHOD

The DSHS algorithm was recently proposed by the author, where a detailed description of the method concept can be found [8]. In this work, the DSHS algorithm is adopted and modified a bit for solving constrained engineering problems. The DSHS method utilizes a quality measure namely the best-to-worst ratio (BtW). The BtW ratio is computed for the current population values represented by the algorithm's harmony memory (HM). At any time the HM would contain a number of solutions including a best solution and a worst solution in terms of their stored quality measures, i.e. fitness function values. With minimization problems in mind, the BtW value is a value in the range [0,1] and as given by the ratio of the current best harmony fitness value to the current worst harmony fitness value in HM. This is expressed in equation (3) where a higher BtW value in this case indicates that the quality of current HM solutions is approaching that of the current best.

$$\text{BtW} = \frac{f(\bar{x}_{\text{best}})}{f(\bar{x}_{\text{worst}})} \quad (3)$$

The PAR value in the proposed method is adjusted dynamically based on the value of the current BtW ratio rather than the value of the current iteration count. This is shown in Fig. 1 where as search progresses the BtW value would eventually decrease owing to having better quality solutions in HM. This behavior is expressed in equation (4) and (5).

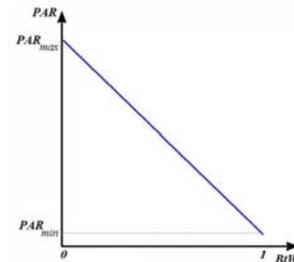


Figure 1. The dynamic setting of PAR in the proposed method [8]

$$\text{slop} = \frac{\text{PAR}_{\text{max}} - \text{PAR}_{\text{min}}}{0 - 1} = \text{PAR}_{\text{min}} - \text{PAR}_{\text{max}} \quad (4)$$

$$\text{PAR} = \text{slop} \cdot \text{BtW} + \text{PAR}_{\text{max}} \quad (5)$$

PAR_{max} is set at the value of 1.0, as in many similar approaches [3, 14], and PAR_{min} is set at a small value that is greater than zero. The pitch-adjusting process is accomplished by using a dynamic bandwidth value (DBW) that considers each harmony vector component (dimension variable) separately and as given in (6) and (7).

$$ActiveBW(i) = C \cdot StdDev(x_{HM}^i) \quad (6)$$

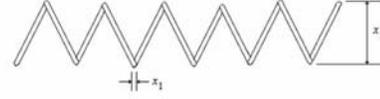
$$DBW(i) = rnd(-ActiveBW(i), ActiveBW(i)) \quad (7)$$

The ActiveBW for the considered dimension variable is computed by calculating the standard deviation of the respective HM column (with index i). The DBW of the dimension variable is a random value confined within the positive and negative range of the ActiveBW for that dimension variable. Based on a number of experiments, the constant C in (6) is set to a fixed value ($C=2.0$). This is unlike the technique used in [8] where C is set dynamically based on the improvisation acceptance rate. Such decision was considered since the dimensionality of the problems (number of variables in \mathbf{x}) discussed in [8] is much larger than what is considered in this work.

IV. EXPERIMENTAL RESULTS & DISCUSSION

One of the common constrained engineering problems is the minimization of the weight of a tension/compression spring [1, 2, 4-7]. The spring to be designed for minimum weight subject to constraints on minimum deflection, shear stress, surge frequency, limits on the outside diameter, and on design variables. The design variables are the wire diameter (x_1), the mean coil diameter (x_2), and also the number of active coils (x_3). Fig. 2 shows the tension/compression spring problem function along with its constraints [6]. This problem does not have equality constraints.

The DSHS algorithm was slightly modified to accommodate the changes in the problem's nature and to use the penalty function introduced earlier in equation (2). The parameter settings are given in Table (1), which are basically the same as in [8] apart from the smaller MAXIMP value. HM was initialized randomly using values from the permissible range. The proposed method was run for 30 times and the optimal solution is represented by the best in terms of fitness function value. Table (2) shows the results of the proposed method in comparison to the results of some other recent methods.



$$\min f(\mathbf{X}) = (x_3 + 2)x_1x_2^2$$

$$g_1(\mathbf{X}) = 1 - \frac{x_1^3x_3}{71,785x_2^4} \leq 0$$

$$g_2(\mathbf{X}) = \frac{4x_1^2 - x_1x_2}{12,566(x_1x_2^3 - x_2^4)} + \frac{1}{5108x_2^2} - 1 \leq 0$$

$$g_3(\mathbf{X}) = 1 - \frac{140.45x_2}{x_1^2x_3} \leq 0$$

$$g_4(\mathbf{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$0.25 \leq x_1 \leq 1.3, \quad 0.05 \leq x_2 \leq 2.0, \quad \text{and } 2 \leq x_3 \leq 15$$

Figure 2. Tension/compression spring constrained optimization problem

TABLE I. PARAMETER SETTINGS FOR DSHS

HMS	HMCR	PAR _{min}	PAR _{max}	C	MAXIMP
50	0.99	0.1	1.0	2.0	200,000

TABLE II. RESULTS FOR TENSION/COMPRESSION SPRING CONSTRAINED OPTIMIZATION PROBLEM

V	IHS [7]	QPSO [6]	NMDE [4]	Proposed Method
x_1	3.49871E-01	3.52529E-01	3.56723E-01	3.89935E-01
x_2	5.11544E-02	5.15150E-02	5.16893E-02	5.30342E-02
x_3	1.20764E+01	1.15389E+01	1.12886E+01	9.57936E+00
g_1	-5.21995E-02	4.83000E-05	-1.78000E-11	-1.25000E-04
g_2	1.36707E-02	-3.58000E-05	-7.55000E-13	-3.25000E-05
g_3	-3.86015E+00	-4.04548E+00	-4.05380E+00	-4.11396E+00
g_4	-7.32650E-01	-7.30637E-01	-7.27725E-01	-7.04687E-01
$f(\mathbf{x})$	1.28874E-02	1.26661E-02	1.26652E-02	1.26996E-02

The obtained results are within the same range of those obtained by other methods. Unlike many other meta-heuristic optimization techniques, the proposed DSHS algorithm does not require any manual setting for its main optimization parameters, namely PAR or BW. In addition, unlike many HS-based methods [2, 7, 12, 13] these two parameters are completely decoupled from the iteration count and changes in accordance with BtW, the newly introduced quality measure. In this case PAR change is not monotonic and is not bound by a maximum iteration count value, i.e. MAXIMP. PAR is set to be inversely proportional to BtW and increases or decreases in response to the quality of current solutions in HM represented by the computed BtW ratio. However, the obtained results could be improved further. In this work the author have adopted the previously

developed DSHS algorithm targeting continuous optimization problems [8] with minimum modification. The problem dimensionality for these continuous optimization problems was comparatively higher (30 to 100) in comparison to the dimensionality of the constrained problem considered in this work. In addition constrained optimization problems are considered to be more difficult because feasible solutions can be hard to find [1]. Due to this probably a more thorough investigation is required to tweak the DSHS method to better suite such problems.

V. CONCLUSIONS

This work proposed using a new meta-heuristic algorithm, referred to as the DSHS algorithm, to solve constrained engineering optimization problems. The DSHS method was previously proposed by the author targeting continuous optimization problems having relatively high dimensionality. The algorithm was slightly modified to suite constrained optimization problems typically with smaller dimensionality. The DSHS algorithm is an HS-based method with dynamic and self-adaptive features. The proposed technique utilizes new quality measure, namely BtW, which based on the current best and worst fitness values to drive the optimization process. Two of the algorithm's main optimization parameters, namely the PAR and the BW are dynamically adjusted during the optimization process. Testing is implemented using a common constrained engineering optimization problem. The results obtained are close to those achieved by some recent meta-heuristic methods. However, a more thorough investigation of the proposed method is required in order to better tweak the algorithm and achieve more superior results.

REFERENCES

- [1] V. c. V. d. Melo and G. L. C. Carosio, "Evaluating differential evolution with penalty function to solve constrained engineering problems," *Expert Systems with Applications*, vol. 39, pp. 7860–7863, 2012.
- [2] M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems," *Commun Nonlinear Sci Numer Simulat*, vol. 15, pp. 3316–3331, 2010.
- [3] C.-M. Wang and Y.-F. Huang, "Self-adaptive harmony search algorithm for optimization," *Expert Systems with Applications*, vol. 37, pp. 2826–2837, 2010.
- [4] D. Zou, H. Liu, L. Gao, and S. Li, "A novel modified differential evolution algorithm for constrained optimization problems," *Computers and Mathematics with Applications*, vol. 61, pp. 1608–1623, 2011.
- [5] E. Zahara and Y.-T. Kao, "Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, pp. 3880–3886, 2009.
- [6] L. d. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, pp. 1676–1683, 2010.
- [7] M. Mahdavi, M. Fesanghary, and E. Damangir, "An Improved Harmony Search Algorithm for Solving Optimization Problems," *Applied Mathematics and Computation*, vol. 188, pp. 1567–1579, 2007.
- [8] A. Kattan and R. Abdullah, "A dynamic self-adaptive harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 219, pp. 8542–8567, 2013.
- [9] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, USA1995.
- [10] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39–43, 1995.
- [11] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 72, pp. 60–68, 2001.
- [12] K. S. Lee and Z. W. Geem, "A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, pp. 3902–3933, 2005.
- [13] M. G. H. Omran and M. Mahdavi, "Global-Best Harmony Search," *Applied Mathematics and Computation*, vol. 198, pp. 643–656, 2008.
- [14] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory Power of the Harmony Search Algorithm: Analysis and Improvements for Global Numerical Optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, pp. 89–106, 2011.