

Harmony Search with Differential Mutation Based Pitch Adjustment

A. K. Qin

INRIA Grenoble Rhone-Alpes
655 avenue de l'Europe, Montbonnot
38334 Saint Ismier Cedex, France
kai.qin@inrialpes.fr

Florence Forbes

INRIA Grenoble Rhone-Alpes
655 avenue de l'Europe, Montbonnot
38334 Saint Ismier Cedex, France
florence.forbes@inrialpes.fr

ABSTRACT

Harmony search (HS), as an emerging metaheuristic technique mimicking the improvisation behavior of musicians, has demonstrated strong efficacy in solving various numerical and real-world optimization problems. This work presents a harmony search with differential mutation based pitch adjustment (HSDM) algorithm, which improves the original pitch adjustment operator of HS using the self-referential differential mutation scheme that features differential evolution – another celebrated metaheuristic algorithm. In HSDM, the differential mutation based pitch adjustment can dynamically adapt the properties of the landscapes being explored at different searching stages. Meanwhile, the pitch adjustment operator's execution probability is allowed to vary randomly between 0 and 1, which can maintain both wild and fine exploitation throughout the searching course. HSDM has been evaluated and compared to the original HS and two recent HS variants using 16 numerical test problems of various searching landscape complexities at 10 and 30 dimensions. HSDM almost always demonstrates superiority on all test problems.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – *heuristic methods*; G.1.6 [Numerical Analysis]: Optimization – *global optimization*

General Terms

Algorithms, Experimentation, Performance

Keywords

DE, Differential evolution, DE, Differential mutation, Harmony search, Metaheuristic

1. INTRODUCTION

Metaheuristic [1] is a generic computational technique aiming at efficiently solving optimization problems that arise in diverse scientific and engineering fields. Recent years have seen

remarkable advances in metaheuristic methods inspired by different kinds of natural and behavioral phenomena, such as genetic algorithm [2, 3], evolution strategy [4, 5], artificial immune system [6], particle swarm optimization [7-10], ant colony optimization [11], and so on. These methods have demonstrated significant efficacy in numerous real-world applications.

Harmony search (HS) [10, 12-21], as an emerging metaheuristic algorithm, mimics the musicians' improvisation behavior. In HS, a candidate solution of an optimization problem corresponds to a musical harmony composed of notes played by a group of musicians. Each decision variable in a candidate solution is analogous to a musician with its value range analogized by the pitch range within which the corresponding musician plays the note. The quality of candidate solutions corresponds to the euphoniousness of musical harmonies. By simulating how a group of musicians keep enriching their experiences to collaboratively seek for the most euphonious harmony in the improvising procedure, HS searches for global optima using harmony improvisation operators to iteratively update the harmony memory (HM) that contains promising candidate solutions.

HS has succeeded in a wide range of applications [13-15], which thus attracts much research attention devoted to further improve its performance. Such research effort is mainly made in terms of the HM initialization, the harmony improvisation, the HM updating and hybridizing HS with other metaheuristic methods. This work focuses on enhancing the pitch adjustment operator of the original HS using the self-referential differential mutation scheme that features another well-known metaheuristic algorithm – differential evolution (DE) [22].

The original pitch adjustment operator of HS intrinsically performs the mutation of a fixed step size with a pre-specified execution probability on those decision variable values generated by the HM consideration operator. Although empirical experiences induce some generic guidelines on how to set up the mutation step size and operator execution probability, these two parameters are closely related to specific problems and searching stages:

- For the mutation step size, larger sizes facilitate jumping out of undesirable local optima while smaller sizes help to refine the already found promising solutions. Problems of different searching landscape's properties (e.g., narrow or wide attraction basins) may prefer distinct step sizes. Moreover, at different searching stages when candidate solutions spread

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Copyright 2011 ACM 978-1-4503-0557-0/11/07...\$10.00.

over different sub-regions of the solution space, the mutation step size is expected to adapt the landscape of the corresponding sub-region. Therefore, the fixed-step-size mutation is not a desirable choice.

- For the operator's execution probability, it is not reasonable to freeze it throughout the searching course because both wild exploitation (when the large probability is applied) and fine exploitation (when the small probability is applied) around those decision variable values generated by the HM consideration operator could be expected during the search.

To address the above issues, we propose a harmony search with differential mutation based pitch adjustment (HSDM) algorithm. HSDM incorporates differential mutation, the essence of the DE algorithm, into the pitch adjustment operator of HS to replace the original fixed-step-size mutation with the DE-style self-referential mutation. Consequently, the mutation step size can be adjusted according to the distribution of promising candidate solutions at any searching stage, which characterizes the landscape of the corresponding sub-region being explored. Moreover, we allow the operator's execution probability to vary randomly between 0 and 1 during the search such that both large and small probabilities can be applied throughout the searching course to enable both wild and fine exploitation around those decision variable values generated by the HM consideration operator.

Experiments on 16 numerical test problems at both 10 and 30 dimensions demonstrate the superiority of HSDM over the original HS and two recent HS variants.

The remaining paper is organized as follows. Section 2 and 3 reviews the original HS and DE algorithms, respectively. HSDM is detailed in Section 4 followed by experiments in Section 5. Section 6 concludes the paper with some future plans.

2. HARMONY SEARCH

HS has received considerable attention since its invention [12], and already developed into an independent research branch of metaheuristic. It evolves a HM as shown in (1), composed of HMS (i.e., harmony memory size) candidate solutions with D decision variables $\mathbf{x}_i = [x_i(1), \dots, x_i(D)]$, $i \in \{1, \dots, HMS\}$, towards global optima using three harmony improvisation operators, i.e., \mathbf{O}_1 : HM consideration operator, \mathbf{O}_2 : random selection operator and \mathbf{O}_3 : pitch adjustment operator, as well as the greedy replacement based HM updating scheme. The objective function $f(\cdot)$ in (1) measures the solution quality. This work only considers single-objective optimization problems where $f(\cdot)$ is a scalar function indicating better quality when its value is smaller (larger) in the case of minimization (maximization).

$$HM = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(D) & | & f(\mathbf{x}_1) \\ x_2(1) & x_2(2) & \cdots & x_2(D) & | & f(\mathbf{x}_2) \\ \vdots & \vdots & \dots & \vdots & | & \vdots \\ x_{HMS}(1) & x_{HMS}(2) & \cdots & x_{HMS}(D) & | & f(\mathbf{x}_{HMS}) \end{bmatrix} \quad (1)$$

Firstly, a HM of fixed size HMS is randomly initialized within the solution space. Then, a new harmony \mathbf{x}_{new} is created by applying

either \mathbf{O}_1 or \mathbf{O}_2 with probabilities $HMCR$ and $1-HMCR$ respectively to determine the value of each decision variable $x_{new}(d)$, $d \in \{1, \dots, D\}$, and subsequently applying \mathbf{O}_3 with probability PAR to refine the values of those decision variables produced by \mathbf{O}_1 . $HMCR$ and PAR denotes the HM consideration rate and the pitch adjustment rate (i.e., the operator execution probability), respectively. The parameter BW associated with \mathbf{O}_3 represents the bandwidth, which determines the maximum value range for the refining (i.e., the mutation step size). The newly generated harmony will replace the worst harmony in the current HM if it has better quality in comparison. This harmony creation and replacement process is repeated until certain termination criterion is met (e.g., the maximum number of function evaluations $maxFEvals$ is reached).

The following describes the pseudo-code of the original HS algorithm for solving minimization problems where $U(0, 1)$ denotes a random number uniformly distributed between 0 and 1. $x_U(d)$ and $x_L(d)$ represent the upper and lower bounds of the solution space with respect to the d^{th} decision variable.

Step 1 Set HS parameters: HMS , $HMCR$, PAR and BW

Step 2 Initialize the HM randomly

```
For ( $i = 1$  to  $HMS$ )
  For ( $d = 1$  to  $D$ )
     $x_i(d) = x_L(d) + U(0, 1) \times (x_U(d) - x_L(d))$ 
  End
  Evaluate  $f(\mathbf{x}_i)$ 
End
```

Step 3 Create a new harmony $\mathbf{x}_{new} = [x_{new}(1), \dots, x_{new}(D)]$ using three harmony improvisation operators \mathbf{O}_1 , \mathbf{O}_2 and \mathbf{O}_3

```
For ( $d = 1$  to  $D$ )
  If ( $U(0, 1) \leq HMCR$ )
     $x_{new}(d) = x_r(d)$ ,  $r$  is random from  $\{1, \dots, HMS\}$  //  $\mathbf{O}_1$ 
  If ( $U(0, 1) \leq PAR$ )
    If ( $U(0, 1) \leq 0.5$ ) //  $\mathbf{O}_3$ 
       $x_{new}(d) = x_{new}(d) + U(0, 1) \times BW$ 
    Else
       $x_{new}(d) = x_{new}(d) - U(0, 1) \times BW$ 
    End
  End
Else //  $\mathbf{O}_2$ 
   $x_{new}(d) = x_L(d) + U(0, 1) \times (x_U(d) - x_L(d))$ 
End
End
Evaluate  $f(\mathbf{x}_{new})$ 
```

Step 4 Update the HM with \mathbf{x}_{new} using the greedy replacement

```
worst =  $\arg \max_i (f(\mathbf{x}_i))$ 
 $\mathbf{x}_{worst} = \mathbf{x}_{new}$ , if  $f(\mathbf{x}_{new}) < f(\mathbf{x}_{worst})$ 
```

Step 5 If any termination criterion is met, return the best harmony found so far, otherwise go to **Step 3**.

Many improved HS algorithms have been developed recently. For example, the improved HS [16] dynamically adjusts the values of *PAR* and *BW* at different searching stages according to certain rules. The global HS [17] creates new harmonies using the global-best harmony in the HM. The local HS [18] periodically regroups the HM, applies HS independently within each group, and uses the group-best harmony to create new harmonies for each group. The self-adaptive HS [19] utilizes the harmony distribution information to perform the pitch adjustment and thus eliminates the parameter *BW*. The self-adaptive global best HS [21] was derived from the global HS with its parameters self-adapted according to previously accumulated successful experiences. HS has also been successfully hybridized with other metaheuristic methods such as particle swarm optimization [10] and DE [14, 20] to collaboratively boost the optimization performance.

3. DIFFERENTIAL EVOLUTION

DE, as a simple yet powerful population-based stochastic search method, has demonstrated outstanding performances in various optimization scenarios [22, 23]. It evolves a population of candidate solutions towards global optima using three operators: mutation, crossover and replacement. The quality of candidate solutions is evaluated using certain objective function.

Firstly, a fixed-size population of candidate solutions is randomly initialized within the solution space. Then, each candidate solution in the population, so-called target vector, respectively undergoes the following three operations in sequence:

- *Mutation*: a base vector is first generated using population members, which determines the reference point of the mutation. Then, the vector difference of randomly sampled population members excluding the target vector under consideration is scaled and added to the basis vector to produce a mutant vector. There exist different ways to create the base vector and vector difference, which correspond to different mutation strategies [22-24].
- *Crossover*: the crossover is applied with certain probability between the above-generated mutant vector and the target vector under consideration to generate a trial vector. Two typical crossover schemes used in DE include binomial (uniform) crossover and exponential (circular two-point) crossover.
- *Replacement*: if the trial vector has better quality than the target vector under consideration, it will replace the target vector and enter the population of the next generation. Otherwise, the target vector will remain in the population of the next generation.

The population is iteratively updated by applying these three operations until certain termination criterion is met (e.g., the maximum number of function evaluations *maxFEvals* is reached).

The success of DE is mainly attributed to its unique differential mutation scheme in the mutation operation, which distinguishes DE from other existing metaheuristic methods, and accordingly coins its name. Two widely used differential mutation schemes using one and two pairs of randomly sampled population members are as follows:

$$\text{One-pair differential mutation: } F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (2)$$

$$\text{Two-pair differential mutation: } F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) + F \cdot (\mathbf{x}_{r_3} - \mathbf{x}_{r_4}) \quad (3)$$

where r_1, r_2, r_3 and r_4 are mutually exclusive indices of the randomly chosen population members. The parameter F is a scaled factor controlling the mutation step size. Scaled difference vectors with respect to all possible pairs of population members adapt the property of the landscape currently explored, which can thus provide promising mutation directions with adjustable step sizes balancing between local and global search. Specifically, at the initial searching stage, population members spread over the entire solution space. Accordingly, lengths of difference vectors are large to favor the global search. As the evolution goes on, population members gradually converge to a small sub-region of the solution space. Consequently, the local search is advocated by small lengths of differential vectors. The adjustable scale factor F can preserve both exploration and exploitation capabilities throughout the searching course.

Recent years have seen numerous DE variants, which improve the original algorithm mainly in terms of the population initialization methods, the mutation strategies, the crossover strategies and the replacement schemes. DE with strategy and parameter adaptation [24] helps to avoid the time-consuming trail-and-error algorithmic configuration and allows fitter strategies and parameters to adapt different searching stages. DE based on specific neighborhood topologies [25] prevents the premature convergence to improve the exploration capability. Hybridizing DE with other metaheuristic algorithms such as genetic algorithm [3], evolution strategy [5] and particle swarm optimization [9] promotes the optimization performance to surpass any individual components.

4. HARMONY SEARCH WITH DIFFERENTIAL MUTATION

As discussed in Section 1, the pitch adjustment operator of the original HS algorithm intrinsically performs the fixed-step-size mutation with a pre-specified execution probability, which cannot adapt the searching landscapes of different problems at different searching stages. On the other hand, the differential mutation scheme in DE provides a spontaneous self-adaptability to the searching landscape. Therefore, it is very desirable to incorporate differential mutation into the pitch adjustment operator of HS.

4.1 HSDM

In this work, we propose a harmony search with differential mutation based pitch adjustment (HSDM) algorithm. HSDM replaces the fixed-step-size mutation in the pitch adjustment operator \mathbf{O}_3 of the original HS with the two-pair differential mutation in (3) and leads to a new pitch adjustment operator $\mathbf{O}_3^{\text{new}}$.

The scale factor F is randomly sampled from a normal distribution with mean value 0.5 and standard deviation 0.3. The execution probability of $\mathbf{O}_3^{\text{new}}$ is randomly chosen during the search from the set $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ that evenly quantizes the range between 0 and 1.

Two-pair differential mutation may yield better perturbation than one-pair differential mutation. It was claimed in [26] that according to the central limit theorem, the random variation of two-pair difference vectors is shifted slightly into the Gaussian direction, which has constituted many competent mutation operators. The advantages of using two-pair difference vectors

were also discussed in the context of particle swarm optimization [9]. Empirical results demonstrate that the statistical distribution of the summation of all possible one-pair difference vectors in the population has a triangle shape while the summation of all possible two-pair difference vectors leads to a bell-shaped distribution that is commonly regarded as a better perturbation mode.

As suggested in [24], the scale factor sampled from a normal distribution with mean value 0.5 and standard deviation 0.3 can guarantee its values between -0.4 and 1.4 with probability 0.99. As such, exploration and exploitation can be simultaneously maintained throughout the searching course.

The varying $\mathbf{O}_3^{\text{new}}$ execution probabilities maintain the capabilities of both wild and fine exploitation, when large and small probabilities are applied respectively, around those decision variable values generated by the HM consideration operator \mathbf{O}_1 throughout the searching course.

The following summarizes the pseudo-code of the HSDM algorithm for solving minimization problems where $U(0, 1)$ and $N(0.5, 0.3)$ denote a random number uniformly distributed between 0 and 1 and a random number normally distributed with mean value 0.5 and standard deviation 0.3, respectively. $x_U(d)$ and $x_L(d)$ represent the upper and lower bounds of the solution space with respect to the d^{th} decision variable.

Step 1 Set HSDM parameters: *HMS* and *HMCR*

Step 2 Initialize the HM randomly

```

For ( $i = 1$  to HMS)
  For ( $d = 1$  to  $D$ )
     $x_i(d) = x_L(d) + U(0, 1) \times (x_U(d) - x_L(d))$ 
  End
  Evaluate  $f(\mathbf{x}_i)$ 
End

```

Step 3 Create a new harmony $\mathbf{x}_{\text{new}} = [x_{\text{new}}(1), \dots, x_{\text{new}}(D)]$ using three harmony improvisation operators \mathbf{O}_1 , \mathbf{O}_2 and $\mathbf{O}_3^{\text{new}}$

Create a two-pair differential mutation vector as $\mathbf{x}_{\text{mut}} = N(0.5, 0.3) \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2} + \mathbf{x}_{r_3} - \mathbf{x}_{r_4})$ where r_1, r_2, r_3 and r_4 are random from $\{1, \dots, \text{HMS}\}$ and mutually exclusive.

Choose *PAR* randomly from the set $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

```

For ( $d = 1$  to  $D$ )
  If ( $U(0, 1) \leq \text{HMCR}$ )
     $x_{\text{new}}(d) = x_r(d)$ ,  $r$  is random from  $\{1, \dots, \text{HMS}\}$  //  $\mathbf{O}_1$ 
  If ( $U(0, 1) \leq \text{PAR}$ ) //  $\mathbf{O}_3^{\text{new}}$ 
     $x_{\text{new}}(d) = x_{\text{new}}(d) + x_{\text{mut}}(d)$ 
  End
  Else
     $x_{\text{new}}(d) = x_L(d) + U(0, 1) \times (x_U(d) - x_L(d))$  //  $\mathbf{O}_2$ 
  End
End

```

Evaluate $f(\mathbf{x}_{\text{new}})$

Step 4 Update the HM with \mathbf{x}_{new} using the greedy replacement

```

worst = arg max $i$  ( $f(\mathbf{x}_i)$ )
 $\mathbf{x}_{\text{worst}} = \mathbf{x}_{\text{new}}$ , if  $f(\mathbf{x}_{\text{new}}) < f(\mathbf{x}_{\text{worst}})$ 

```

Step 5 If any termination criterion is met, return the best harmony found so far, otherwise go to **Step 3**.

4.2 Related Work

Among the existing HS variants, two methods are related to the proposed HSDM algorithm and deserve more details.

Self-adaptive harmony search (SaHS) algorithm [19] modifies the original pitch adjustment operator \mathbf{O}_3 of HS using the distribution of harmonies in the HM. Suppose $x_{\text{max}}(d)$ and $x_{\text{min}}(d)$ respectively denote the maximum and minimum values of the d^{th} decision variable in the current HM. SaHS employs the following pitch adjustment operator to replace \mathbf{O}_3 :

```

If ( $U(0, 1) \leq 0.5$ )
   $x_{\text{new}}(d) = x_{\text{new}}(d) + U(0, 1) \cdot (x_{\text{max}}(d) - x_{\text{new}}(d))$ 
Else
   $x_{\text{new}}(d) = x_{\text{new}}(d) - U(0, 1) \cdot (x_{\text{new}}(d) - x_{\text{min}}(d))$ 
End

```

Such an operator can also adapt the searching landscapes at different searching stages. However, using the value range in the HM cannot properly characterize the landscape's properties especially when the landscape is rotated, and therefore may not provide promising mutation directions.

SaHS follows the rules in [19] to dynamically change the pitch adjustment operator's execution probability *PAR* as:

$$PAR(t) = PAR_{\text{min}} + \frac{PAR_{\text{max}} - PAR_{\text{min}}}{\text{maxFEvals}} \cdot t \quad (4)$$

where t , $PAR(t)$, PAR_{max} and PAR_{min} represent the current number of function evaluations, the *PAR* value at the current number of function evaluations, the maximum and minimum *PAR* values, respectively.

Differential harmony search (DHS) algorithm [20] incorporates differential mutation into HS in a different manner from HSDM. Specifically, DHS skips the original pitch adjustment operator \mathbf{O}_3 of HS and applies the one-pair differential mutation in (2) on those decision variable values generated using the HM consideration operator \mathbf{O}_1 as well as the random selection operator \mathbf{O}_2 . The scale factor F is randomly sampled from a uniform distribution between 0 and 1. The differential mutation's execution probability takes a pre-specified value.

In fact, it may not be reasonable to apply differential mutation on those decision variable values generated using the random selection operator \mathbf{O}_2 . Differential mutation can only provide promising mutation directions within the sub-region explored by the current HM. However, the decision variable values generated by \mathbf{O}_2 are often outside such a sub-region.

5. EXPERIMENTS

The performance of HSDM is evaluated and compared to the original HS, SaHS and DHS using 16 numerical test problems [8] of different searching landscape complexities at 10 and 30 dimensions (i.e., 10D and 30D).

5.1 Test Problems

The 16 numerical test problems defined in [8], including one unimodal functions and 15 multimodal functions, are employed in this work. All problems are tested at 10D and 30D. According to their properties, these 16 test functions are categorized into four groups as follows.

Group A: *Two unimodal and simple multimodal problems*

F₁: Sphere function

F₂: Rosenbrock's function

Group B: *Six unrotated multimodal problems*

F₃: Ackley's function

F₄: Griewanks's function

F₅: Weierstrass function

F₆: Rastrigin's function

F₇: Noncontinuous Rastrigin's function

F₈: Schwefel's function

Group C: *Six rotated multimodal problems*

F₉: Rotated Ackley's function

F₁₀: Rotated Griewanks's function

F₁₁: Rotated Weierstrass function

F₁₂: Rotated Rastrigin's function

F₁₃: Rotated noncontinuous Rastrigin's function

F₁₄: Rotated Schwefel's function

Group D: *Two composition problems*

F₁₅: Composition function 1 (CF1)

F₁₆: Composition function 5 (CF5)

The function definition, global optima and their corresponding objective function values, solution space ranges and initialization ranges are all detailed in Section IV in [8]. We do not reiterate them here due to page limitation. Note that unsymmetrical initialization ranges are used for those functions whose global optima are at the center of the solution space.

5.2 Experimental Setup

The parameters of HS are set as per empirical guidelines [12, 13] as: $HMS = 50$, $HMCR = 0.98$, $PAR = 0.3$, $BW = 0.01$. As suggested in [19], the parameters of SaHS are set as: $HMS = 50$, $HMCR = 0.98$, $PAR_{min} = 0.0$, $PAR_{max} = 1.0$. The parameters of DHS are set according to [20] as: $HMS = 50$, $HMCR = 0.98$, $PAR = 0.3$. The parameters of HSDM are set as: $HMS = 50$, $HMCR = 0.98$. Note that common parameters among HS, SaHS, DHS and HSDM are set same for the fair comparison.

All algorithms are implemented in MATLAB, and performed on a Linux PC with the Intel Xeon E5520 CPU at 2.27 GHz.

For each test problem, each algorithm to be compared is executed 25 times starting from different random seeds while all of four algorithms (HS, SaHS, DHS and HSDM) share the same random seed with respect to any individual run.

Two stopping criteria are applied: (1) the maximum number of function evaluations ($maxFEvals$) is reached. Here, the $maxFEvals$ is set to 10^4 times the problem dimension, which means 10^5 for 10D problems and 3×10^5 for 30D problems; (2) The difference of objective function values between the best solution found so far and the global optimal solution (i.e., error function value (EFV)) is smaller than 10^{-8} . In such a case, the EFV is negligible and set to zero.

The optimization performance is quantitatively measured by (1) the mean value and standard deviation of the best EFVs achieved when an algorithm terminates over 25 runs and (2) the success rate (SR) over 25 runs. An optimization algorithm is regarded as successfully solving the problem once it achieves an EFV smaller than the pre-specified accuracy level. In this work, the accuracy levels of all 16 functions are set to 10^{-8} .

Practical optimization tasks are often subjected to the strict requirement on the computation speed of the algorithm applied to solve them, which is proportional to the executed number of function evaluations. To inspect an optimization algorithm's efficacy with respect to various computation budgets (i.e., the maximally allowed number of function evaluations), the empirical cumulative distribution function (ECDF) [27] with respect to the number of function evaluations at success (i.e., the number of function evaluations when the EFV just goes below the pre-specified accuracy level) over 25 runs on all 16 test functions is illustrated.

5.3 Results

Tables 1 and 2 report and compare, with respect to each of 16 test functions at 10D and 30D respectively, the optimization performances of HS, SaHS, DHS and HSDM in terms of the mean value and standard deviation of the best EFVs over 25 runs as well as the SR under the accuracy level 10^{-8} over 25 runs. For each function, bold fonts show the largest SR (if not zero) and the optimal best EFVs (i.e., with the smallest mean value) as well as those best EFVs indiscernible from the optimal based on the Wilcoxon's signed-rank test [28] at the significance level of 0.05. This nonparametric statistical hypothesis test assesses whether the medians of two sets of the best EFVs achieved by two algorithms over 25 runs are statistically significantly different.

It is observed that, at both 10D and 30D, HSDM consistently demonstrates superior performances compared to HS, SaHS and DHS on all test functions except for F₂. F₂ is the Rosenbrock's function, which has a long, narrow and parabolic shaped flat valley from the perceived local optima to the global optimum. The differential mutation in HSDM is not competent to adapt this valley. Moreover, the random scale factor and execution probability of the pitch adjustment operator in HSDM may even distract the evolution path from the valley. Therefore, HSDM is beaten by the original HS, which exploits the valley using the mutation of a small fixed step size with a small fixed execution probability.

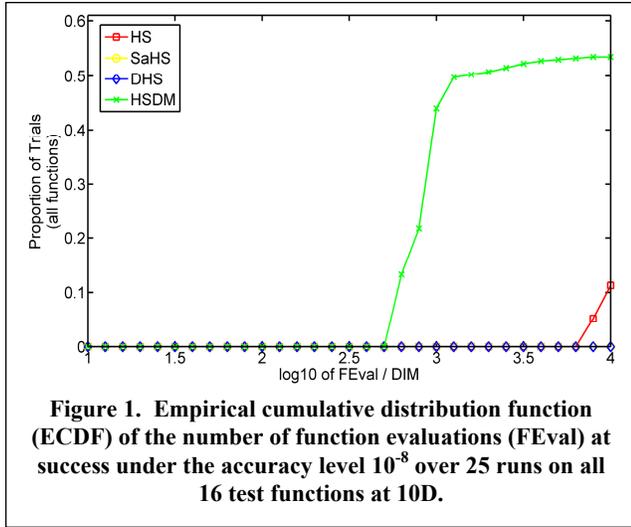


Figure 1. Empirical cumulative distribution function (ECDF) of the number of function evaluations (FEval) at success under the accuracy level 10^{-8} over 25 runs on all 16 test functions at 10D.

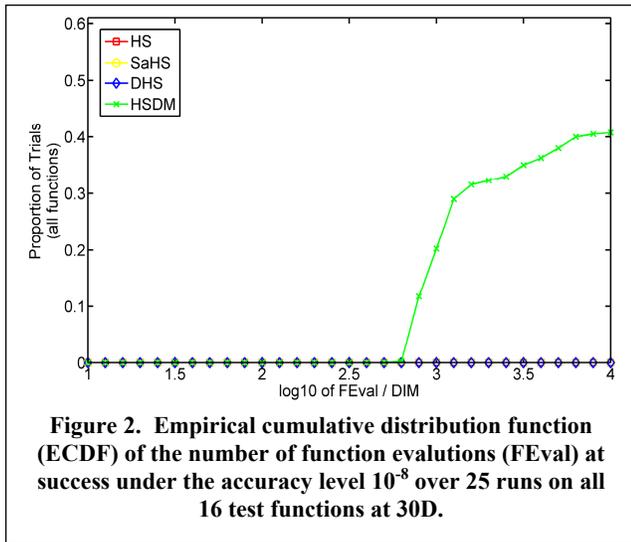


Figure 2. Empirical cumulative distribution function (ECDF) of the number of function evaluations (FEval) at success under the accuracy level 10^{-8} over 25 runs on all 16 test functions at 30D.

For 10D problems, although the original HS compares favorably with HSDM in terms of the best EFVs on four functions F_6 , F_7 , F_8 and F_{14} , it underperforms HSDM from the aspect of the SRs on these functions. In fact, regarding functions F_6 , F_7 , F_{14} , the mean value of the best EFVs over 25 runs for HSDM are augmented by a few failing runs.

Another interesting observation is that compared to the original HS, both SaHS and DHS do not demonstrate prominent performances on many test functions at 10D and 30D.

Figures 1 and 2 illustrate, at 10D and 30D respectively, the ECDFs with respect to the number of function evaluations at success under the accuracy level 10^{-8} over 25 runs on all 16 test functions. It is observed that HSDM always outperforms the other three algorithms in comparison after around 5000 (10D) and 19000 (30D) function evaluations. Before that, all four algorithms are equally incompetent.

Table 1. Performances of HS, SaHS, DHS and HSDM in terms of the mean value and standard deviation (*italic* below the mean value) of the best error function values achieved when the algorithm terminates as well as the success rate under the accuracy level 10^{-8} over 25 runs with respect to each of 16 test functions at 10D. Bold fonts show the optimal value as well as those indiscernible from the optimal based upon Wilcoxon's signed-rank test at the significance level of 0.05.

		HS	SaHS	DHS	HSDM
F_1	Best	3.519E-09	1.903E-02	8.125E-02	0.000E+00
	EFV	<i>6.747E-09</i>	<i>1.953E-02</i>	<i>5.208E-02</i>	<i>0.000E+00</i>
	SR	0.76	0.00	0.00	1.00
F_2	Best	1.048E+00	5.661E+00	6.026E+00	6.681E+00
	EFV	<i>4.971E-01</i>	<i>2.584E+00</i>	<i>1.648E+00</i>	<i>9.942E-01</i>
	SR	0.00	0.00	0.00	0.00
F_3	Best	9.562E-05	5.823E-02	1.472E-01	0.000E+00
	EFV	<i>2.681E-05</i>	<i>4.897E-02</i>	<i>6.702E-02</i>	<i>0.000E+00</i>
	SR	0.00	0.00	0.00	1.00
F_4	Best	5.914E-02	8.419E-02	1.572E-01	1.141E-02
	EFV	<i>3.372E-02</i>	<i>3.666E-02</i>	<i>4.996E-02</i>	<i>8.883E-03</i>
	SR	0.04	0.00	0.00	0.24
F_5	Best	4.144E-02	1.308E-01	3.103E-01	0.000E+00
	EFV	<i>7.480E-03</i>	<i>5.040E-02</i>	<i>8.284E-02</i>	<i>0.000E+00</i>
	SR	0.00	0.00	0.00	1.00
F_6	Best	8.885E-07	1.394E-02	3.482E-02	2.356E-04
	EFV	<i>6.041E-07</i>	<i>1.387E-02</i>	<i>2.312E-02</i>	<i>8.800E-04</i>
	SR	0.00	0.00	0.00	0.80
F_7	Best	8.986E-07	9.955E-03	3.292E-02	9.472E-04
	EFV	<i>5.871E-07</i>	<i>1.100E-02</i>	<i>2.042E-02</i>	<i>2.386E-03</i>
	SR	0.00	0.00	0.00	0.60
F_8	Best	1.108E-09	6.354E-02	2.034E-01	0.000E+00
	EFV	<i>5.540E-09</i>	<i>8.271E-02</i>	<i>9.216E-02</i>	<i>0.000E+00</i>
	SR	0.96	0.00	0.00	1.00
F_9	Best	1.776E+00	1.313E+00	1.176E+00	0.000E+00
	EFV	<i>9.399E-01</i>	<i>9.651E-01</i>	<i>8.993E-01</i>	<i>0.000E+00</i>
	SR	0.00	0.00	0.00	1.00
F_{10}	Best	1.702E-01	2.529E-01	3.145E-01	1.038E-02
	EFV	<i>1.120E-01</i>	<i>1.272E-01</i>	<i>1.151E-01</i>	<i>9.165E-03</i>
	SR	0.00	0.00	0.00	0.20
F_{11}	Best	7.348E-01	1.790E+00	1.475E+00	2.201E-02
	EFV	<i>6.203E-01</i>	<i>6.721E-01</i>	<i>7.114E-01</i>	<i>1.037E-01</i>
	SR	0.00	0.00	0.00	0.76
F_{12}	Best	7.960E+00	8.205E+00	7.920E+00	3.781E+00
	EFV	<i>2.829E+00</i>	<i>3.590E+00</i>	<i>3.773E+00</i>	<i>1.770E+00</i>
	SR	0.00	0.00	0.00	0.00
F_{13}	Best	6.840E+00	7.121E+00	7.009E+00	5.325E+00
	EFV	<i>2.561E+00</i>	<i>2.385E+00</i>	<i>1.976E+00</i>	<i>1.387E+00</i>
	SR	0.00	0.00	0.00	0.00
F_{14}	Best	5.868E+02	7.095E+02	5.567E+02	4.385E+02
	EFV	<i>3.486E+02</i>	<i>2.942E+02</i>	<i>3.306E+02</i>	<i>2.722E+02</i>

	SR	0.04	0.00	0.00	0.12
F ₁₅	Best	2.400E+01	3.601E+01	3.602E+01	2.400E+01
	EFV	<i>4.359E+01</i>	<i>4.899E+01</i>	<i>4.899E+01</i>	<i>4.359E+01</i>
	SR	0.00	0.00	0.00	0.76
F ₁₆	Best	1.600E+02	1.334E+02	1.251E+02	8.455E+01
	EFV	<i>8.428E+01</i>	<i>9.489E+01</i>	<i>8.134E+01</i>	<i>3.682E+01</i>
	SR	0.00	0.00	0.00	0.08

Table 2. Performances of HS, SaHS, DHS and HSDM in terms of the mean value and standard deviation (*italic* below the mean value) of the best error function values achieved when the algorithm terminates as well as the success rate under the accuracy level 10^{-8} over 25 runs with respect to each of 16 test functions at 30D. Bold fonts show the optimal value as well as those indiscernible from the optimal based upon Wilcoxon's signed-rank test at the significance level of 0.05.

		HS	SaHS	DHS	HSDM
F ₁	Best	2.920E-05	1.100E-02	5.650E-02	8.055E-06
	EFV	<i>5.389E-06</i>	<i>7.020E-03</i>	<i>2.645E-02</i>	<i>3.508E-05</i>
	SR	0.00	0.00	0.00	0.80
F ₂	Best	2.458E+01	4.538E+01	4.460E+01	2.629E+01
	EFV	<i>1.759E+01</i>	<i>2.917E+01</i>	<i>2.790E+01</i>	<i>8.400E-01</i>
	SR	0.00	0.00	0.00	0.00
F ₃	Best	4.001E-03	3.145E-02	6.169E-02	4.395E-05
	EFV	<i>2.954E-04</i>	<i>1.657E-02</i>	<i>1.326E-02</i>	<i>1.552E-04</i>
	SR	0.00	0.00	0.00	0.52
F ₄	Best	1.406E-02	5.934E-02	1.205E-01	7.186E-04
	EFV	<i>1.366E-02</i>	<i>3.994E-02</i>	<i>3.284E-02</i>	<i>2.078E-03</i>
	SR	0.00	0.00	0.00	0.64
F ₅	Best	8.216E-01	2.570E-01	6.142E-01	2.295E-04
	EFV	<i>8.927E-02</i>	<i>5.796E-02</i>	<i>6.688E-02</i>	<i>8.220E-04</i>
	SR	0.00	0.00	0.00	0.84
F ₆	Best	5.391E-03	5.900E-03	3.059E-02	7.079E-05
	EFV	<i>9.596E-04</i>	<i>5.019E-03</i>	<i>1.298E-02</i>	<i>2.315E-04</i>
	SR	0.00	0.00	0.00	0.80
F ₇	Best	5.535E-03	7.053E-03	2.773E-02	4.908E-05
	EFV	<i>7.296E-04</i>	<i>5.838E-03</i>	<i>1.039E-02</i>	<i>1.234E-04</i>
	SR	0.00	0.00	0.00	0.76
F ₈	Best	4.117E-06	3.583E-02	1.587E-01	1.429E-08
	EFV	<i>5.359E-07</i>	<i>2.079E-02</i>	<i>5.545E-02</i>	<i>7.143E-08</i>
	SR	0.00	0.00	0.00	0.96
F ₉	Best	2.344E+00	1.980E+00	1.233E+00	6.187E-04
	EFV	<i>7.473E-01</i>	<i>5.194E-01</i>	<i>7.737E-01</i>	<i>1.232E-03</i>
	SR	0.00	0.00	0.00	0.36
F ₁₀	Best	1.761E-02	1.345E-01	1.823E-01	3.704E-03
	EFV	<i>1.633E-02</i>	<i>6.672E-02</i>	<i>5.729E-02</i>	<i>9.660E-03</i>
	SR	0.00	0.00	0.00	0.00
F ₁₁	Best	4.686E+00	5.718E+00	5.602E+00	3.487E-01
	EFV	<i>1.456E+00</i>	<i>1.573E+00</i>	<i>1.342E+00</i>	<i>4.952E-01</i>
	SR	0.00	0.00	0.00	0.00

F ₁₂	Best	2.914E+01	2.540E+01	2.593E+01	1.437E+01
	EFV	<i>6.001E+00</i>	<i>6.108E+00</i>	<i>5.581E+00</i>	<i>4.866E+00</i>
	SR	0.00	0.00	0.00	0.00
F ₁₃	Best	2.256E+01	2.308E+01	2.234E+01	1.886E+01
	EFV	<i>5.066E+00</i>	<i>5.439E+00</i>	<i>6.001E+00</i>	<i>1.090E+01</i>
	SR	0.00	0.00	0.00	0.00
F ₁₄	Best	2.283E+03	2.234E+03	2.340E+03	1.699E+03
	EFV	<i>5.489E+02</i>	<i>5.016E+02</i>	<i>6.995E+02</i>	<i>4.148E+02</i>
	SR	0.00	0.00	0.00	0.00
F ₁₅	Best	8.001E+00	7.800E-04	4.003E+00	3.646E-06
	EFV	<i>4.000E+01</i>	<i>6.583E-04</i>	<i>2.000E+01</i>	<i>1.375E-05</i>
	SR	0.00	0.00	0.00	0.84
F ₁₆	Best	2.846E+01	2.039E+01	2.106E+01	4.162E+00
	EFV	<i>1.880E+01</i>	<i>1.877E+01</i>	<i>1.899E+01</i>	<i>1.649E+00</i>
	SR	0.00	0.00	0.00	0.00

6. CONCLUSIONS AND FUTURE WORK

This work presents a harmony search with differential mutation based pitch adjustment (HSDM) algorithm, which incorporates the DE-style two-pair differential mutation scheme with the pitch adjustment operator of HS. Such introduced self-referential mutation can help HSDM to dynamically adapt the properties of landscapes being explored at different searching stages. Moreover, the execution probability of the pitch adjustment operator in HSDM varies randomly between 0 and 1, which can maintain both wild and fine exploitation on those values generated by the HM consideration operator throughout the searching course. HSDM has been compared to the original HS and two recent HS variants (SaHS and DHS) using 16 numerical test problems at 10D and 30D. HSDM almost always outperforms its competitors on all test problems at both 10D and 30D.

Ongoing and planned research agendas include: investigating the parameter self-adaptation in HSDM based on the accumulated evolution information, improving HSDM from more aspects such as the HM initialization, the HM topology and the replacement scheme, and evaluating HSDM extensively on more numerical and real-world optimization problems.

7. ACKNOWLEDGMENTS

Experiments presented in this paper were carried out using the Grid5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>). This work was partially supported by NSFC under Grant No. 61005051 and SRFDP under Grant No. 20100092120027.

8. REFERENCES

- [1] Glover, F. W., and Kochenberger, G. A. *Handbook of Metaheuristics, 1st edition*. Springer, 2003.
- [2] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers, 1989.
- [3] Zhang, M., Geng, H. T., Luo, W. J., Huang, L. F., and Wang, X. F. A hybrid of differential evolution and genetic algorithm for constrained multiobjective optimization

- problems. In *Proceeds of the 6th International Conference on Simulated Evolution and Learning (SEAL'06)*, 2006, 318-327.
- [4] Hansen, N., and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9, 2, 2001, 159-195.
- [5] Kämpf, J. H., and Robinson, D. A hybrid CMA-ES and HDE optimization algorithm with application to solar energy potential. *Applied Soft Computing*, 9, 2, 2009, 738-745.
- [6] De Castro, L. N., and Timmis, J. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, 2002.
- [7] Kennedy, J., and Eberhart, R. Particle swarm optimization. In *Proceeds of the 1995 IEEE International Conference on Neural Networks (IJCNN'95)*, 1995, 1942-1948.
- [8] Liang, J. J., Qin, A. K., Suganthan, P. N., and Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10, 3, 2006, 281-295.
- [9] Zhang, W. T., and Xie, X. F. DEPSO: hybrid particle swarm with differential evolution operator. In *Proceeds of the 2003 IEEE International Conference on Systems Man and Cybernetics (SMC'03)*, 2003, 3816-3821.
- [10] Zhao, S. Z., Suganthan, P. N., Pan, Q. K., and Tasgetiren, M. F. Dynamic multi-swarm particle swarm optimizer with harmony search. *Expert Systems with Application*, 38, 4, 2011, 3735-3742.
- [11] Dorigo, M., and Stützle, T. *Ant Colony Optimization*. IT Press, Cambridge, MA, 2004.
- [12] Geem, Z. W., Kim, J. H., and Loganathan, G. V. A new heuristic optimization algorithm: harmony search. *Simulation*, 76, 2, 2001, 60-68.
- [13] Geem, Z. W. *Music-Inspired Harmony Search Algorithm: Theory and Applications, 1st edition*. Springer, 2009.
- [14] Liao, T. W. Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, 10, 4, 2010, 1188-1199.
- [15] Geem, Z. W., and Williams, J. C. Ecological optimization using harmony search. In *Proceeds of American Conference on Applied Mathematics*, 2008.
- [16] Mahdavi, M., Fesanghary, M., and Damangir, E. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188, 2, 2007, 1567-1579.
- [17] Omran, M. G. H., and Mahdavi, M. Global-best harmony search. *Applied Mathematics and Computation*, 198, 2, 2008, 643-656.
- [18] Pan, Q. K., Suganthan, P. N., Liang, J. J., Tasgetiren, M. F. A local-best harmony search algorithm with dynamic subpopulations. *Engineering Optimization*, 42, 2, 2010, 101-117.
- [19] Wang, C.-M., and Huang, Y.-F. Self-adaptive harmony search algorithm for optimization. *Expert Systems with Applications*, 37, 4, 2010, 2826-2837.
- [20] Chakraborty, P., Roy, G. G., Das, S., Jain, D., and Abraham, A. An improved harmony search algorithm with differential mutation operator. *Fundamenta Informaticae*, 95, 2009, 1-26.
- [21] Pan, Q. K., Suganthan, P. N., Tasgetiren, M. F., Liang, J. J. A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation*, 216, 3, 2010, 830-848.
- [22] Price, K., Storn, R., and Lampinen, J. *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, Berlin, Germany, 2005.
- [23] Das, S., and Suganthan, P. N. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 14, 6, 2010, 1-28.
- [24] Qin, A. K., Huang, V. L., and Suganthan, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13, 2, 2009, 398-417.
- [25] Das, S., Abraham, A., Chakraborty, U. K., and Konar, A. Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, 13, 3, 2009, 526-553.
- [26] Storn, R. On the usage of differential evolution for function optimization. In *Proceedings of Biennial Conference of North America Fuzzy Information Process*, 1996, 519-523.
- [27] Hansen, N., Auger, A., Finck, S., and Ros, R. *Real-parameter black-box optimization benchmarking 2010: experimental setup*. Technical Report RR-7215, INRIA, France, 2010.
- [28] Siegel, S. *Non-Parametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York, 1956.