Hybrid evolutionary search method for complex function optimisation problems

H. Binol, I. Guvenc, E. Bulut and K. Akkaya

In this Letter, harmony search (HS) technique hybridised with genetic algorithm (GA) is proposed. This technique mainly takes HS direction estimation mechanism and genetic operators in GA, which significantly increase the convergence of the HS algorithm. Specifically, the authors propose to incorporate main operators of GA into the HS algorithm to avoid some inherent drawbacks of the HS. For example, crossover is incorporated into HS to deal with low accuracy problem, while mutation is incorporated to escape from the local optimum solutions. In addition, elitism is introduced into the HS, to precipitate the performance and prevent the loss of favourable individuals found during the search process. The authors compare the performance of the GA, HS, and other popular HS variants on several benchmark functions. Numerical results show that the proposed hybridisation exhibits a superior performance in comparison to other algorithms.

Introduction: The harmony search (HS) [1] is a population-based stochastic technique inspired from the music *improvisation* process. A group of candidate solutions called harmony vectors are first created randomly in the feasible variable space and accumulated in the harmony memory (HM). A new candidate harmony vector is then improvised in HS by implementing three main actions: usage of HM, pitch adjustment, and randomisation. If the new vector is better than at least one of the harmony in the HM, the worst harmony vector is replaced by the new one, and thus the population is restored. Genetic algorithms (GAs) [2], on the other hand, are population-based stochastic global search methods that imitate the mechanisms of natural selection. Building on Darwin's evolution theory, the GA mimics the progress of a population of candidate solutions to solve quantitative problems. The candidates in the GA acclimate to an objective function over a development process utilising biology-like operators such as the chromosome crossovers and gene mutations. The GA picks out two individuals from the existing mating pool to create two offspring for the next generation at each step.

In this Letter, we propose a novel search strategy based on hybridisation of HS and GA techniques to achieve some improvements in optimisation problems which have challenging objective functions. In particular, we exploit crossover operator in the memory consideration phase of HS to increase solution quality and mutation scheme to take an enhanced exploratory role. Moreover, rank-based elitism is utilised in the reproduction process of the proposed algorithm to preserve exceptional characteristics over generations. Furthermore, we dynamically adjust the key parameters of the algorithm over iterations to keep it adaptable and robust. Numerical results obtained with the proposed hybrid approach were compared with the results with both HS and GA, and also other several optimisation algorithms in the literature. The results clearly show that the proposed solution is superior to not only the HS and the GA, but also other recent variants.

Harmony search: Initially, the HM is constructed with randomly generated elements, as many as HM size (S_{HM}), as follows:

$$x_i^j = (u_i - l_i) \cdot \mathrm{rand}() + l_i, \quad i = 1, 2, \ldots, d, \tag{1}$$

where x_i^j is the ith element of jth harmony; l_i and u_i are the lower/upper bounds of the ith variable; rand() is a uniformly-distributed real random number in [0, 1]; and d is the dimension of the problem.

In the process of iteration after generation of initial HM, improvisation process operates based on two control parameters – HM consideration rate (R_{HMC}), pitch adjustment rate (p)

$$x_i' = \begin{cases} x_i^{\mathrm{rand}}, & \mathrm{rand}() < R_{\mathrm{HMC}} \\ (u_i - l_i) \cdot \mathrm{rand}() + l_i, & \mathrm{else} \end{cases} \tag{2}$$

Here, x_i^{rand} is the ith element of a random harmony which selected in the inclusive range from 1 to S_{HM}.

Thereafter, each part obtained by the memory concern rule is shifted to its local space, which yields

$$x_i' = \begin{cases} x_i' \pm \mathrm{rand}().f_i, & \mathrm{rand}() < p, \\ x_i', & \mathrm{else} \end{cases} \tag{3}$$

where f_i is the fret width which is the amount of maximum change in pitch adjusting. The improvisation will persist until the termination conditions are satisfied.

Elitist hybrid algorithm: Given the overview of the HS in previous section, this section introduces an elitist HS algorithm based on GA (eHSGA) to deal with the drawbacks of original HS and GA. The main motivations of HS and GA to reach a feasible solution have different aspects considering the harmony and HM phenomena in HS, and chromosome and population in GA. It is thus appealing to hybridise HS with genetic procedures in order to acquire some potential enhancement in performance. The main procedure of the proposed algorithm is to begin with initiation of the parameters including the S_{HM}, R_{HMC}, p, maximum number of improvisation (N_{IMP}), and f.

In classical HS, all harmonies in HM are with the same probability of utilising in improvisation. In the proposed elitist scheme, the HM is organised to a sorted list with the descending order of the good-fitting values. The sorting ensures that current best harmony is always in the head of HM. As described in Algorithm 1, pitch adjustment is adopted on the best-so-far harmony. The crossover is applied between the best member and the current harmony with probability $(1 - p)$. The usage of elitism ensures sufficient harmony is employed as the local explorer of search process.

Algorithm 1: The Elitist HS-GA (eHSGA)

```
initialise the algorithm parameters; S_HM, p, R_HMC, f, and d
create the initial population randomly
evaluate the fitness value of each member
while not stopping condition do
  for k = 1 : S_HM do
    if rand() < R_HMC then
      pick the best value (x_best) from HM
      if rand() < p then
        pitch adjustment on x_best
        x_rand^new = x_rand^best + (2.rand() − 1).f_rand
      else
        crossover between x^k and a random member (x^rand)
      end
    else
      random selection
      x_i^new = (u_i − l_i).rand() + l_i
    end
  end
  f ← f × 99%
end
return the best solution in the HM
```

The two fundamental parameters, i.e. R_{HMC} and p, of HS have naturally great impact on the speed of the convergence of the search algorithm. In order to assure that the hybrid search method can expeditiously detect its way to local optima in the early process and the solutions reached in the later are diverse, eHSGA updates the R_{HMC} as linearly decreasing with the iteration $R_{\mathrm{HMC}}^{(t)} \leftarrow R_{\mathrm{HMC}}^{\max} - ((R_{\mathrm{HMC}}^{\max} - R_{\mathrm{HMC}}^{\min}).t/N_{\mathrm{IMP}})$, where $R_{\mathrm{HMC}}^{(t)}$ is the HM consideration rate for generation t; N_{IMP} is the maximum number of the improvisations; $R_{\mathrm{HMC}}^{\max}$ and $R_{\mathrm{HMC}}^{\min}$ denote the predefined maximum and minimum HM considering rates, respectively. Dynamic change strategy for pitch adjustment is also integrated into the algorithm and p finally adopted in eHSGA is adaptive and linearly increasing as $p(t) \leftarrow p_{\min} + ((p_{\max} - p_{\min}).t/N_{\mathrm{IMP}})$, where $p(t)$ denotes the pitch adjustment rate for the tth iteration.

We utilised a merge rule to update the HM that determines which individuals will survive to the succeeding generation. In that method, good harmonies are chosen as many as S_{HM} from old (HM^{old}) and new memories (HM^{new}) which include sorted harmonies in a descending order of fitness values after each iteration. We also employed the mutation operator in Algorithm 2 which replaces the long-term survival harmony with a new generated one if the new one has a better fitness.

Algorithm 2: Mutation operator on long-term survival harmonies

```
for t = 1 : N_IMP do
  for k = 1 : S_HM do
    flag(x_k) ← flag(x_k) + 1
    if flag(x_k) > threshold then
      x_rand^mut = x_rand^k + (2.rand() − 1).f_rand^t
      if f(x^mut) < f(x^k) then
        replace x^k with x^mut
      end
    end
    if x_k ∉ HM^update then
      flag(x_k) ← 0
    end
  end
end
```

Simulations and discussions: To evaluate the effectiveness of the proposed eHSGA, we chose several challenging benchmark functions from [3], i.e. the sixth Bukin function (*Bukin N.6*; it has many local minima), the *Ackley* which has many regularly distributed local minima, the *Griewank* has many regularly distributed widespread local minima, the *Rastrigin* function which is highly multimodal and has several local minima, the modified Himmelblau (*MHB*) which is multimodal and has four local minima, the *Drop-Wave* function which is multimodal and highly complex, and the *Easom* function which is unimodal and has several local minima. Performance of the proposed algorithm is compared with HS from [4], GA from [5], particle swarm optimisation (PSO) from [6], improved HS from [7], and a recent HS–GA hybridisation (HGHS) from [8]. In HGHS, after the initialisation of HM, the GA operators, e.g. selection, crossover, and mutation, are employed to create next generation. The generated population is held as the HM, and the classical HS procedure is pursued in the forthcoming steps. In contrast, we consider to modify the exploration strategy of HS by integrating a crossover step into pitch adjustment, and mutating generated solutions remain same for a long time period, as discussed earlier. While the crossover operation in our eHSGA algorithm results in some complexity increase, it also helps the solution to avoid being stuck in local minima and hence it reaches a solution closer to the global best.

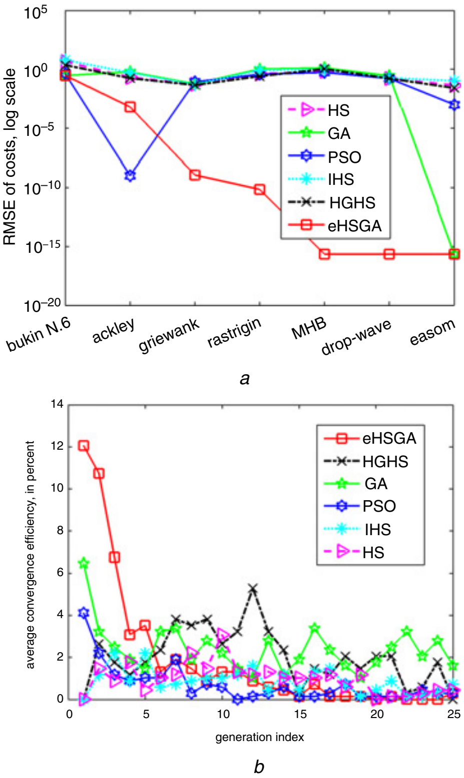

Fig. 1 *Performances of eHSGA and five other techniques on seven challenging benchmark functions*

a Performance comparison of methods combining all the 50 test cases generated in terms of RMSE of costs

b Convergence efficiencies corresponding to the ratio of good solutions in each generation of metaheuristic algorithms for Drop-Wave function

The population size (S_{HM}) is fixed at 40 and $p_{\max} = 0.8$, $p_{\min} = 0.4$, $R_{\mathrm{HMC}}^{\max} = 0.8$, $R_{\mathrm{HMC}}^{\min} = 0.2$, and initial f is set to 1. All search methods are set to terminate after 1000 iterations. Threshold value for the mutation is selected as 10. Each algorithm is tested 50 times for each function and root mean square error (RMSE) of fitness values, which is the sample standard deviation of the differences between real and observed values, is also calculated for all the test cases. Search domain is taken as $-50 \leq x_d \leq 50$, $d = 1, 2$. Our simulation results in Fig. 1 show that eHSGA outperforms any other method except the Ackley function on recorded statistic measures. The eHSGA converges to exact global minima for three of benchmark functions, i.e. MHB, Drop-Wave, and Easom, in all runs. The second best HS variant is HGHS which reaches a better point than the original HS on four of the problems. PSO which is a powerful group-based stochastic optimisation technique is superior to the proposed algorithm for only Ackley function in terms of RMSE. We plot the convergence efficiencies of all algorithms for the first 25 generations. Fig. 1*b* shows the recorded average values (in percent) of the ratio of the number of good members that outperform the previous best solution to the overall population. In other words, it shows how fast a specific algorithm will tend to converge to the solution. Starting with the early generations, eHSGA provides the highest percentage improvement on the Drop-Wave function, significantly better than other approaches. Overall, our results indicate that eHSGA has very strong stability and robustness.

Conclusion: In this Letter, we present an elitist hybrid metaheuristic approach based on HS and GA. In the proposed algorithm, fixed-point crossover procedure is applied at generation reproduction mechanism of HS. Moreover, mutation operator is adopted to prevent getting stuck in a locally optimal solution in search process. The elitist way is incorporated in both reproduction and HM updating in the proposed technique. To judge the performance of the algorithm, numerical experiments have been conducted on challenging test functions to compare with other methods. We showed that the hybrid approach can effectively improve the convergence of the both HS and GA. A possible future work will concentrate on the extending of the proposed algorithm to deal with multi-objective optimisation problems. Our approach can be applied to various optimisation and machine learning problems in engineering such as in telecommunications, computer vision, and automation.

Acknowledgments: This publication was made possible by #NPRP9-257-1-056 from the QNRF (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. The MATLAB codes used to generate the results in this letter are available at https://research.ece.ncsu.edu/mpact/data-management/.

Submitted: *2 August 2018* E-first: *18 October 2018*
doi: 10.1049/el.2018.6506
One or more of the Figures in this Letter are available in colour online.

H. Binol and K. Akkaya (*Department of ECE, Florida International University, Miami, FL, USA*)

✉ E-mail: hbinol@fiu.edu

I. Guvenc (*Department of ECE, NC State University, Raleigh, NC, USA*)

E. Bulut (*Department of CS, Virginia Commonwealth University, Richmond, VA, USA*)

References

1 Lee, K.S., and Geem, Z.W.: 'A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice', *Comput. Methods Appl. Mech. Eng.*, 2005, **194**, (36-38), pp. 3902–3933
2 Whitley, D.: 'A genetic algorithm tutorial', *Stat. Comput.*, 1994, **4**, (2), pp. 65–85
3 Surjanovic, S., and Bingham, D.: 'Virtual library of simulation experiments: test functions and datasets'. (http://www.sfu.ca/ ssurjano), online, accessed 29 May 2018
4 Alia, O.M., and Mandava, R.: 'The variants of the harmony search algorithm: an overview', *Artif. Intell. Rev.*, 2011, **36**, (1), pp. 49–68
5 Malhotra, R., Singh, N., and Singh, Y.: 'Genetic algorithms: concepts, design for optimization of process controllers', *Comput. Inf. Sci.*, 2011, **4**, (2), p. 39
6 Marini, F., and Walczak, B.: 'Particle swarm optimization (PSO). A tutorial', *Chemometr. Intell. Lab Syst.*, 2015, **149**, pp. 153–165
7 dos Santos-Coelho, L., and Mariani, V.C.: 'An improved harmony search algorithm for power economic load dispatch', *Energy Convers. Manage.*, 2009, **50**, (10), pp. 2522–2526
8 Manzoor, A., Asif, K., Zunaira, N., *et al.*: 'A hybrid genetic based on harmony search method to schedule electric tasks in smart home'. Proc. Int. Conf. P2P, Parallel, Grid, Cloud and Internet Computing, Barcelona, Spain, November 2017, pp. 154–166