A Hybrid Algorithm based on Differential Evolution, Particle Swarm Optimization and Harmony Search Algorithms

Ezgi Deniz Ülker
Computer Engineering Department,
Girne American University, University Drive, Karaoglanoglu, Girne, Mersin 10, Turkey
Email: ezgideniz@gau.edu.tr

Ali Haydar
Computer Engineering Department,
Girne American University, University Drive, Karaoglanoglu, Girne, Mersin 10, Turkey
Email: ahaydar@gau.edu.tr

Abstract—**Evolutionary optimization algorithms and their hybrid forms have become popular for solving multimodal complex problems which are very difficult to solve by traditional methods in the recent years. In the literature, many hybrid algorithms are proposed in order to achieve a better performance than the well-known evolutionary optimization methods being used alone by combining their features for balancing the exploration and exploitation goals of the optimization algorithms. This paper proposes a novel hybrid algorithm composed of Differential Evolution algorithm, Particle Swarm Optimization algorithm and Harmony Search algorithm which is called HDPH. The proposed algorithm is compared with these three algorithms on the basis of solution quality and robustness. Numerical results based on several well-studied benchmark functions have shown that HDPH has a good solution quality with high robustness. Also, in HDPH all parameters are randomized which prevents the disadvantage of selecting all possible combination of parameter values in the selected ranges and of finding the best value set by parameter tuning.**

I. Introduction

In recent years, many different optimization techniques have been proposed for solving the complex, multimodal functions in several fields [1-4]. Some of the well-known optimization algorithms are the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, Ant Colony Optimization (ACO) algorithm, Differential Evolution (DE) algorithm, and Harmony Search (HS) algorithm. These algorithms are used in various fields by many researchers to obtain the optimum value of the problems [5-10]. Each optimization algorithm uses different properties to keep a balance between the exploration and exploitation goals which can be a key for the success of an algorithm. Exploration attribute of an algorithm enables the algorithm to test several areas in the search space. On the other hand, exploitation attribute makes the algorithm focus the search around the possible candidates. Although the optimization algorithms have positive characteristics, it is shown that these algorithms do not always perform as well as it is desired [11]. Because of this, hybrid algorithms are growing area of interest since their solution quality can be made better than the algorithms that form them by combining their desirable features. Hybridization is simply the combination of two or more techniques in order to outperform their performances by the use of their good properties together. Hybridization has been done in several different ways in the literature and it is observed that the new hybridization techniques are very efficient and effective for optimization [11-16].

A novel hybrid algorithm proposed in this paper is called HDPH and it is a combination of three well known evolutionary algorithms, namely Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, and Harmony Search (HS) algorithm. It merges the general operators of each algorithm recursively. This achieves both good exploration and exploitation in HDPH without altering their individual properties.

HDPH is compared with the three algorithms that form it on the basis of the solution quality and the robustness on random initialization of a solution set. The set of well studied benchmark functions which are Multimodal (M)/Separable (S) or Multimodal (M)/Non-separable(N) are used for the evaluation.

The rest of the paper is organized as follows; Section II describes the HDPH algorithm that is proposed in detail. Section III presents the performances of the hybrid algorithm and the algorithms that generate it together and also our discussions. In the last section, Section IV, the concluding remarks of the paper are given.

HDPH Algorithm

In the literature, many different ways of combining the well-known algorithms are performed to obtain more powerful optimization algorithms [11-16]. The main aim of the hybridization is to use different properties of different algorithms to improve the solution quality.

Among the well-known algorithms, DE, PSO and HS algorithms are the three algorithms that are used in many fields by researchers and these algorithms are proven to be very powerful optimization tools [5-8]. Each algorithm has different strong features. As an example, DE usually requires less computational time and also has better approxi-

978-1-4673-4471-5/$25.00

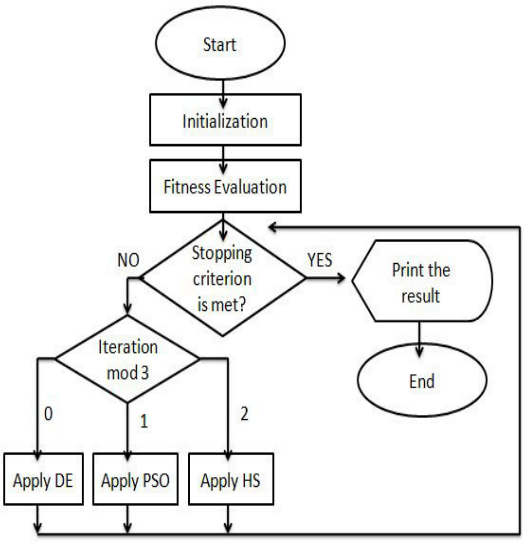

Fig. 1.Flowchart of HDPH

mation of solutions for most of the problems. PSO generally avoids the solution from trapping into local minima by using its diversity. HS on the other hand, is an efficient algorithm that has a very good performance on different applications.

HDPH uses the operators of these three algorithms with randomly selected parameters consecutively and by not altering their properties. The new candidate set, obtained by each algorithm, is used as a new solution set for the other algorithm. Fig.1 shows the HDPH algorithm in the form of a flowchart which demonstrates the main steps of the process.

The summarized steps of HDPH can be given as follows:

Step 1. *Generation of the candidate population with given dimensions:* Initialize the candidate population $X_{i,j}$ in a given range.

Step 2. *Crossover and mutation operators of DE:* The mutation and crossover operators are applied to find the better approximation to a solution by using (1), (2), and (3).

The mutant vector V_{ij} is calculated as corresponding to each member in population using (1) where *a, b,* and *c* are distinct numbers. Mutant vector V_{ij} is crossoverred with X_{ij} and trial vector U_{ij} is generated by using (2) where r_j is a uniformly distributed number for each j^{th} parameter of X_i. Also, *F* and *CR* are the main control parameters of DE.

$$V_i = X_a + F(X_b - X_c) \tag{1}$$

$$U_{ij} = \begin{cases} V_{ij} & if \;\; r_j \le CR \\ X_{ij} & otherwise \end{cases} \tag{2}$$

$$X_i = \begin{cases} U_i & if \;\; f(U_i) < f(X_i) \\ X_i & otherwise \end{cases} \tag{3}$$

.

Selection process determines U_{ij} to survive to the next generation by using (3).

TABLE I
MULTIMODAL-SEPARABLE AND MULTIMODAL-NON-SEPARABLE BENCHMARK FUNCTIONS

D	**Function**	**Formula**	f_{min}
2	Booth	$\left[(x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \right]$	0
30	Rastrigin	$\sum_{i=1}^{n} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$	0
30	Schwefel	$\left[\sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right) \right]$	-418.9*D
10	Michalewicz 10	$\left[-\sum_{i=1}^{n} \sin(x_i) * \left(\sin\left(\frac{i x_i^2}{\pi} \right) \right)^{2m} \right]$ $m = 10$	-9.6602
2	Schaffer	$\left[0.5 + \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2} \right]$	0
2	Six Hump Camel Back	$\left[4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4 \right]$	-1.03163
2	Shubert	$\left[\left(\sum_{i=1}^{5} i \cos((i+1)x_1 + i) \right) * \left(\sum_{i=1}^{5} i \cos((i+1)x_2 + i) \right) \right]$	-186.73
30	Griewank	$\left[\frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 \right]$	0
30	Ackley	$\left[-20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e \right]$	0
30	Penalized	$\left[\frac{\pi}{n} \left\{ 10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) \right]$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - a)^m, x_i < a \end{cases}$	0

Step 3. *Particle movement by PSO:* The randomly selected parameters are applied on the velocities by using (4). When a better solution is being discovered, all particles improve their positions by using (5). This movement avoids the particles to be trapped to the local minima by increasing the diversity of solution. V_{ij} refers to the velocity values and for each row is calculated according to the control parameters c_1, c_2, and w by using (4). $global_{best}$ is the best position obtained by any particle and P_{best} is the personal best of a particle. X_{ij} refers to current positions of a particle and can be updated by using (5) for each row.

$$V_i = w * V_i + c_1 * (P_{best} - X_i) + c_2 * (global_{best} - X_i) \quad (4)$$

$$X_i = X_i + V_i \quad (5)$$

Step 4. *Choosing a neighboring value by HS:* HS can search in different zones of the search space by using the control parameters that are *hmcr, par* and *fw*. With a given probability of *hmcr*, a value is selected from the candidate population. With a given probability of 1-*hmcr*, a random candidate is generated in the given range. The population can have non-updated candidates to keep the diversity in the population with a given probability of 1-*par*. With a given probability of *par,* the candidates are updated by applying (6) where *rand()* is a random number ∈ (-1,1).

$$X_i = X_i + rand() * fw \quad (6)$$

Step 5. Consecutively Step 2, Step 3, and Step 4 are applied.

The algorithm is performed until the termination criterion is not satisfied. Elitism is included in HDPH by keeping the best solution at the end of each iteration.

III. Numerical Results and Discussions

The proposed hybrid algorithm HDPH is tested using 10 well known benchmark functions with different characteristics and is compared for the solution quality and robustness for random initialization of the population with the three algorithms used to form it. The benchmark functions are selected as Multimodal (M)/Separable (S) and Multimodal (M)/Non-separable (N). These benchmark functions are presented in Table I. The population size for the functions is fixed to 100 for all algorithms.

The two control parameters of DE algorithm which are *F* and *CR* are selected from the sets given as follows; $F \in \{0.3, 0.5, 0.7, 0.8, 0.9, 1.2, 1.4\}$ and $CR \in \{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$. The three control parameters that are used in PSO algorithm are selected from the sets as given; *c1* and *c2* $\in \{0.3, 0.6, 0.9, 1.2, 1.5, 1.8\}$, and $w \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For the HS algorithm, the control parameters called *hmcr* and *par* are selected from the sets as follows; $hmcr \in \{0.7, 0.8, 0.9, 0.93, 0.96, 0.98\}$ and $par \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$. The control parameter *fw* is adjusted as 0.01, 0.05, 0.1, and 0.2 times the upper bound of each function in HS. Each possible combination of control parameters is selected and each selection is run for 20 times for each algorithm. The selected parameter ranges are chosen similar to the commonly used ranges in the literature. For each function, the control parameter values that are closest to the optimum solution are selected and the function is further evaluated around these selected control parameters. By doing this, we try to achieve a good parameter tuning.

For the HDPH model, instead of selecting the eight control parameters discretely from the sets used for three algorithms, they are selected randomly from the parameter ranges that are formed by selecting the minimum and maximum elements of each parameter set as the lower and upper bounds of the ranges for these parameters respectively. This is done because it would have been very difficult to test all possible combination of parameter values otherwise. The results are obtained only by running the hybrid model for 20 times.

In Tables II and III, the performances of these algorithms over 10000 function evaluations are shown. For each algorithm, the best value (*BestVal*), the average (*Avg*) of the 20 runs for the selected best value parameters and the standard deviations (*Stdev*) are shown. In case that there are more than one control parameter values that give the best value, the one that has a closer average to the optimal value and smaller standard deviation is chosen.

The results that are obtained for the selected MS functions are shown in Table II. The best values obtained using HDPH, except Rastrigin function, are either better or similar to the best values obtained by the other three algorithms. The standard deviations and the averages of HDPH for Schwefel and Michalewicz10 functions are substantially better than the other three algorithms. However, for the Rastrigin function, the standard deviation and the average of HS algorithm are better than HDPH and for the Booth function, all three algorithms have a better standard deviation values compared to HDPH.

In Table III, the results for MN functions are tabulated. For Griewank, Ackley, and Penalized functions, the best values obtained using HDPH outperform the other three algorithms. For these three functions, when both the average and standard deviation values are taken into consideration, the HDPH gives better results than DE and PSO algorithms. When it is compared by the HS algorithm, except Ackley function which gives similar results, HDPH is again better than HS algorithm. For the Schaffer, Six Hump Camel Back and Shubert functions, both the best values and standard deviations are comparable for all four algorithms.

It can be seen from the results that HDPH generally worked as good as or sometimes better than other three algorithms in terms of solution quality and robustness. This is achieved by running the HDPH algorithm only 20 times. For the other three algorithms, the tabulated results are obtained by running the programs 20 times for all possible combinations of parameters, finding the parameter set that gives the best performance, making a parameter tuning around those values and using those parameters that has achieved the best performance. This point is a verification of the good performance of HDPH algorithm.

TABLE II
RESULTS FOR MULTIMODAL-SEPARABLE FUNCTIONS

Function MS	Values	HDPH	DE	PSO	HS
Booth	Avg	0.0001	1.37E-25	0	3.36E-07
	Stdev	0.0007	1.80E-25	0	5.02E-07
	BestVal	0	2.41E-27	0	1.27E-08
Rastrigin	Avg	36.18	137.899	46.3655	18.1256
	Stdev	14.203	6.68121	17.416	3.41769
	BestVal	21.82	126.013	19.0816	12.7443
Schwefel	Avg	-12567.6	-7485.74	-8531.08	-12554.6
	Stdev	2.5759	270.62	949.247	28.8299
	BestVal	-12569.5	-8128.58	-10353.9	-12566.1
Micha10	Avg	-9.65918	-9.13592	-8.00576	-9.6111
	Stdev	0.0025	0.11902	0.93836	0.05591
	BestVal	-9.66015	-9.31606	-9.65524	-9.66004

TABLE III
RESULTS FOR MULTIMODAL-NON-SEPARABLE FUNCTIONS

Function MN	Values	HDPH	DE	PSO	HS
Schaffer	Avg	0.007923	0.00107	0.00250	0.00923
	Stdev	0.003701	0.00088	0.00428	0.00217
	BestVal	0	7.80E-05	0	1.85E-06
Six Hump Camel Back	Avg	-1.03163	-1.03163	-1.03163	-1.03163
	Stdev	0	0	0	3.66E-06
	BestVal	-1.03163	-1.03163	-1.03163	-1.03163
Shubert	Avg	-186.722	-185.624	-186.729	-186.727
	Stdev	0.026154	1.40940	0.00959	0.00438
	BestVal	-186.731	-186.703	-186.731	-186.731
Griewank	Avg	0.045248	1.53190	0.39352	1.04977
	Stdev	0.071979	0.19544	0.31861	0.0222
	BestVal	0.000208	1.29684	0.05316	1.00414
Ackley	Avg	0.885152	16.7758	2.86698	1.09805
	Stdev	0.594669	0.75719	1.01934	0.29879
	BestVal	0.007775	15.1746	0.65150	0.56317
Penalized	Avg	0.031359	5.08107	4.36306	0.29210
	Stdev	0.056563	2.13136	2.94708	0.24432
	BestVal	3.59E-06	2.79334	0.37862	0.04269

IV. CONCLUSION

In this work, the new hybrid algorithm, called HDPH, is proposed to achieve a robust algorithm with a good solution quality by combining the three well-known algorithms, DE, PSO and HS. The performances of chosen algorithms are based on the parameter selection. Therefore, all combination of parameter values are tested for each function for all three algorithms and the results that are tabulated are selected as the best values obtained through all possible trials. However, in the HDPH algorithm the parameters are chosen randomly in the given ranges which make the algorithm easier to implement. Even with this kind of simplification in HDPH algorithm, the good performance is verified. Also, the experimental results have shown that, when both solution quality and robustness of an algorithm are taken into consideration, in most of the test functions, HDPH is more robust than the other three algorithms. At the same time, HDPH, for many functions analyzed, has similar or even better solution quality than the three algorithms that composes it. Hence, the proposed hybrid algorithm, HDPH, makes use of the features of the three algorithms and has similar or better solution quality with high robustness to random initialization of the population.

REFERENCES

[1] Z.W. Geem, J.H. Kim, and G.V. Loganathan, “A New Heuristic Optimization Algorithm: Harmony Search”, *Simul., Trans. of the Soc. for Model. and Simul. Int*., pp. 60-68, 2001.

[2] J. Kennedy, R. Eberhart, “Particle Swarm Optimization. Piscataway”, *Proc. of IEEE Int. Conf. on Neural Netw. IV, NJ: IEEE Press*, pp. 1942-1948, 1995.

[3] R. Storn, K. Price, “Differential Evolution; A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, *J. of Glob. Optim*., vol.11, pp. 341-359, 1997.

[4] M. Dorigo, G. Dicaro, “Ant colony optimization: A new meta-heuristic”, *Evol. Comput. CEC 99. Proc. of the 1999 Congr. on.*, vol. 2, pp. 1470-1477, 1999.

[5] F. Wenlong, M. Johnston, and M.Zhang, “Soft edge maps from edge detectors evolved by genetic programming”, *Evol. Comput. Proc. IEEE,* pp.1-8, 2012.

[6] R. Storn, “Differential Evolution Design of an IIR-filter”, *Evol. Comput. Proc. IEEE,* pp. 268-273, 1996.

[7] Z.W. Geem, J.H. Kim, and G.V. Loganathan, “Harmony Search optimization, Application to pipe network design”, *Int. J. of Model.&Simul.,* vol. 22, pp. 125-133, 2002.

[8] Z.W. Geem, C. Tseng, and Y. Park, “Harmony Search for Generalized Orienteering Problem: Best touring in China”, *Springer Lect. Notes in Comput. Sci.,* vol. 3412, pp. 741-750, 2005.

[9] I. Hitoshi, “Using genetic programming to predict financial data”, *Evol. Comput. CEC 99. Proc. of the 1999 Congr. on.*, vol.1, pp.244-251, 1999.

[10] R.S. Parpinelli, H.S. Lopes, and A.A. Freitas, “Data mining with an Ant Colony Optimization Algorithm”, *IEEE Trans. on Evol. Comput.,* vol.6, pp. 312-332, 2002.

[11] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, “Particle Swarm Optimization: Hybridization perspectives and experimental illustrations”, *Appl. Math. and Comput.,* vol. 217, pp. 5208-5226, 2011.

[12] X.H. Shi, Y.C. Liang, and L.M. Wang, “An improved GA and novel PSO-GA-based hybrid algorithm”, *Inf. Process. Lett.,* vol. 93, pp. 255-261, 2005.

[13] N. Holden, A.A. Freitas, “A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data.” *Swarm Intell. Symp. SIS 2005,* pp.100-107, 2005.

[14] A.A.A. Esmin, G.T. Torres, and G.B. Alvarenga, “Hybrid Evolutionary Algorithm Based on PSO and GA mutation”, *In proc. of the Sixth Int. Conf. on Hybrid Intell. Syst*., pp.57, 2006.

[15] H. Li, and H. Li, “A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems”, *The Int. Conf. on Intell. Pervasive Comput.*, pp. 94-97, 2007.

[16] I. Cionei, E. Kyriakides, “Hybrid Ant Colony-Genetic Algorithm (GAAPI) for Global Continuous Optimization”, *IEEE Trans. on Syst., Man, and Cybern. - Part B: Cybern.,* vol.42, pp. 234-245, 2012.