Hybrid-Adaptive Differential Evolution with Decay Function (HyDE-DF) Applied to the 100-Digit Challenge Competition on Single Objective Numerical Optimization

Fernando Lezama, João Soares, Ricardo Faia, and Zita Vale
GECAD-Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development
Polytechnic of Porto, Portugal

ABSTRACT

In this paper, a hybrid-adaptive differential evolution with a decay function (HyDE-DF)[1] is proposed for numerical function optimization. The proposed HyDE-DF is applied to the 100-Digit Challenge in a set of 10 benchmark functions. Results show that HyDE-DF can achieve a 93/100 score, proving its effectiveness for numerical optimization.

CCS CONCEPTS

• **Computing methodologies** → *Search methodologies*; • **Applied computing** → **Engineering**.

KEYWORDS

Evolutionary computation, differential evolution, numerical optimization

1 INTRODUCTION

Classical optimization approaches have been struggling to match the growing complexity presented in different scientific and industry domains. In such a context, alternative methods, such as evolutionary computation (EC), represent a viable option for solving complex problems. EC is a branch of computational intelligence that encompasses a set of algorithms for global optimization mostly inspired by biological and evolutionary processes [1].

In this paper, we propose a hybrid-adaptive differential evolution with a decay function (HyDE-DF) for numerical optimization. The proposed algorithms is a modified version of HyDE [2], including a decay function and an extra re-initialization step. HyDE-DF integrates ideas from different EAs and has already demonstrated good performance in solving a real-world application in the energy domain [2].

2 BENCHMARK FUNCTIONS

In this study, we use the set of ten benchmark functions from the 100-digit challenge [3]. Functions are listed in Table 1. The complete details of the used functions are given in [3].

[1] Algorithm implementation available at: https://fernandolezama.github.io./publication

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

Table 1: The 100-Digit Challenge Basic Test Functions.

No.	Function	Min	D	Range
F1	Storn's Chebyshev	1	9	[-8192,8192]
F2	Inverse Hilbert	1	16	[-16384,16384]
F3	Lennard-Jones	1	18	[-4,4]
F4	Rastrigin's Function	1	10	[-100,100]
F5	Griewangk's Function	1	10	[-100,100]
F6	Weierstrass Function	1	10	[-100,100]
F7	Modified Schwefel's	1	10	[-100,100]
F8	Expanded Schaffer's F6	1	10	[-100,100]
F9	Happy Cat	1	10	[-100,100]
F10	Ackley Function	1	10	[-100,100]

3 HYDE WITH DECAY FUNCTION AND RE-INITIALIZATION STEP

HyDE-DF combines ideas from different evolutionary algorithms being an improved version of the HyDE algorithm proposed in [2]. HyDE-DF uses the so-called "$DE/target-to-perturbed\_best/1$" mutation strategy of HyDE with a decay factor δ_G, which is a function that decreases gradually from 1 to 0 in a period of DF_t iterations. The operator is as follows:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + \delta_G \cdot [F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})] + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{1}$$

where $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ are two random individuals from the population (Pop), mutually different and also different from the current target vector $\vec{x}_{i,G}$, while $\vec{x}_{best}$ is the best found solution. F_i^1, F_i^2, and F_i^3, are scale factors in the range [0,1] independent for each individual i, and updated at each iteration following the self-adaptive parameter mechanism of jDE algorithm (see [4]). $\epsilon = \mathcal{N}(F_i^3, 1)$ is a random perturbation factor taken from a normal distribution with mean F_i^3 and standard deviation 1. The factor δ_G is used to gradually decrease the influence of the term $F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})$ responsible for the fast convergence towards the best individual in the population.

The HyDE-DF also incorporates a reinitialization mechanism that is activated if $R_N = 1e4$ successive iterations show no improvement in the objective function. In such case, the original population is replaced by generating new individuals around a given number of best solutions found so far (we used 10 in our experiments). The new individuals are generated using random numbers that follow a normal distribution with mean of those best solutions and standard deviation of 10e-4. The best individual in the population after reinitialization is kept to preserve memory. A pseudocode of the algorithm is given in Algorithm 1.

Algorithm 1 HyDE-DF pseudocode

Set the control parameters $F_i^1, F_i^2, F_i^3, Cr_i = 0.5$ and NP.
Generate the initial population Pop.
Evaluate fitness of every individual.
Save best fitness individual x_{best}
for G=1:GEN **do**
  Calculate decay factor δ_G (linearly decreasing factor)
  Generate F_i^1, F_i^2, F_i^3, and Cr_i $\forall i \in Pop$
  for i=1:NP **do**
    Select two individuals: $x_{r1,G} \neq x_{r2,G}$ different from $x_{i,G}$.
    Apply mutation operator Eq. (1)
    Apply recombination (same as standard DE).
    Verify boundary constraints.
    Apply selection operator (same as standard DE) and update Pop.
  end for
  Update F_i^1, F_i^2, F_i^3, and Cr_i $\forall i \in Pop$ (as jDE)
  Update best individual x_{best}
  If DF_t iterations passed, $\delta_G = 1$, o.w., decrease $\delta_G \longrightarrow 0$
  Apply reinitialization of population if in R_N successive iterations there is no objective value improvement
end for

4 RESULTS

We applied HyDE-DF to the set of 10 benchmark functions provided in [3]. HyDE-DF has few control parameters, since a self-adaptive mechanism is employed (i.e., the jDE adaptive mechanism). Table 2 presents the initial parameters used in common for all the functions. The algorithm stop if the 10 digits of precision are achieved, or a maximum budget of 8e6 iterations is reached. Regarding tuning of parameters, Table 4 Shows the specific values of NP (size of population) used for the functions. We mainly use $NP = 50$ for functions 1 to 6, and function 9, $NP = 100$ for functions 7 and 9, and $NP = 200$ for function 8. The experiments were implemented in MATLAB 2014b in a computer with Intel Xeon(R) E5-2620v2@2.1 GHz processor with 16GB of RAM running Windows 10[2].

Table 2: Default values of parameters in HyDE-DF

Parameter	Value
Max Iterations	8.00E+06
Cr	0.5
F	0.5
DF_t	1.00E+05

Table 3 presents the number of correct digits found for each of the functions in 50 trials. Column "Score" is the average number of correct digits in the best 25 trials, i.e. if 50% or more of the trials find all 10 digits, then the score for that function is a perfect 10. We can see that HyDE-DF was able to obtain a final score of 93. In fact, HyDE-DF obtained a score in 9 out of 10 functions. Only in the Function 9 (Happy Cat Function), HyDE-DF was not able of getting more than 3 digits of accuracy.

[2]Algorithm implementation available at: https://fernandolezama.github.io./publication

Table 3: Fifty runs sorted by the number of correct digits

F	Number of correct digits											Score
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
1	0	0	0	0	0	0	0	0	0	0	50	10
2	0	0	0	0	0	0	0	0	0	0	50	10
3	0	7	0	0	0	0	0	0	0	0	43	10
4	0	0	0	0	0	0	0	0	0	0	50	10
5	0	0	0	0	0	0	0	0	0	0	50	10
6	0	0	0	0	0	0	0	0	0	0	50	10
7	0	1	16	0	0	0	0	0	0	0	33	10
8	0	0	4	0	0	0	0	0	0	0	46	10
9	0	0	4	46	0	0	0	0	0	0	0	3
10	0	0	0	0	0	0	0	0	0	0	50	10
											Total:	**93**

Table 4: Tuned parameter values

Function	NP
1-6 and **10**	50
7 and **9**	100
8	200

5 CONCLUSIONS

In this paper, we applied a new HyDE-DF algorithm to the 100-digit challenge. HyDE-DF is an improved version of HyDE, a self-adaptive DE algorithm that also incorporates a perturbation of the best individual and a decay function in its main operator. Being a self-adaptive version of DE, HyDE-DF eliminates the tedious tuning of parameters and yet achieved a good performance in the tested functions. Results showed that HyDE-DF can achieve a final score of 93, struggling only with the happy cat function in which only 3 digits of precision were found. Future work will analyze and compare the performance of HyDE-DF in a more complete and complex set of benchmark functions.

ACKNOWLEDGMENTS

This research has received funding from FEDER funds through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under Project POCI-01-0145-FEDER-028983; by National Funds through the FCT Portuguese Foundation for Science and Technology, under Projects PTDC/EEI-EEE/28983/2017 (CENERGETIC).

REFERENCES

[1] Joao Soares, Tiago Pinto, Fernando Lezama, and Hugo Morais. Survey on complex optimization and simulation for the new power systems paradigm. *Complexity*, 214(1), 2018.
[2] F. Lezama, J. Soares, R. Faia, T. Pinto, and Z. Vale. A new hybrid-adaptive differential evolution for a smart grid application under uncertainty. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, July 2018.
[3] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan. Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. *Technical Report Nanyang Technological University, Singapore*, 2018.
[4] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec, and V. Zumer. Dynamic optimization using self-adaptive differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 415–422, May 2009.