A Winner-Take-All Methodology: Finding the Best Evolutionary Algorithm for the Global Optimization of Functions

Tiantian Zhang
University of Central Florida
Department of EECS
Orlando, FL 32816
zhangtt@knights.ucf.edu

Giselle Borrero
University of Central Florida
Department of EECS
Orlando, FL 32816
gaborrero@knights.ucf.edu

Michael Georgiopoulos
University of Central Florida
Department of EECS
Orlando, FL 32816
michaelg@ucf.edu

ABSTRACT

The problem of effectively and efficiently finding the global optimum of a function by using evolutionary algorithms is current and pertinent, and two of the evolutionary techniques that have received significant attention in the literature are Particle Swarm Optimization (PSO), and Differential Evolution (DE), as well as their numerous variants. One way of taking advantage of the many good PSO and DE variant algorithms that have appeared in the literature is to run them all for a particular optimization problem and choose the best answer provided. This approach, referred to as the Naive Approach (NA) is time consuming. In this paper, we are using the naive approach with a suite of algorithms for each function minimization problem and we run the algorithms, in the suite, for a specific number of function evaluations (typically much smaller than the number of function evaluations needed for each specific algorithm to converge to a solution) and decide, using appropriate performance measures of merit, which one of these algorithms will continue running until convergence; the rest of the algorithms, deemed as algorithms that will eventually produce inferior solutions, are aborted. We refer to our methodology, for obvious reasons, as the Winner-Take-All (WTA) methodology. Using this methodology we introduce WTA algorithmic variants that are efficient and effective solvers of global optimization problems. In this paper, we report results on one of these variants, called WTA1, and show that it is very competitive compared to the constituent algorithms in the suite used for its design, and efficient compared to NA.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global Optimization*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Experimentation

Keywords

Particle Swarm Optimization, Differential Evolution, Single-objective Optimization

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

1. INTRODUCTION

Since PSO's and DE's introduction many of their variants that solve optimization problems have appeared in the literature. The first trend in designing PSO and DE variants is the *modified algorithm approach*, where a deficiency of an existing evolutionary algorithm is recognized and remedies to address this deficiency are proposed. The second trend in designing PSO and DE variants is the *hybrid algorithm approach*, where a few select algorithms, with differing strengths and weaknesses, are chosen and combined to create a *superior algorithm* that does better than its constituent counterparts in finding the function's global minimum. One trend that is typically neglected due to its high inefficiency, is the *naive approach (NA)*, where a number of existing PSO and DE variants are applied to the optimization problem of interest and the best result, produced by any variant, is retained.

In this paper, we introduce the *Winner-Take-All (WTA)* methodology, succinctly explained in the abstract, that makes NA more efficient. In particular, one WTA variant, called WTA1 is discussed and experimentally compared with NA and the constituent algorithms used in its design. Of note is that the WTA methodology frees the practitioner from the sometimes unrealistic demand of the modified and hybrid algorithm approaches that require knowledge of the evolutionary algorithms in order to modify them or combine them.

2. ALGORITHM DESIGN

The WTA algorithm designer is faced with a number of choices, such as the algorithms to consider in the suite of evolutionary algorithms that will be concurrently run and evaluated, the measures of merit used in their evaluation, the time instance at which such an algorithmic evaluation will take place, and the number of algorithms from the suite that will be retained at such a time instance. In the WTA1 design we used the following suite of algorithms: AIW-PSO[8], CPSO-S[4], CLPSO[2], DE, JADE[6] and DEGL/SAW[1]. The *Completion Time* of the retained algorithms equals to $N_p \cdot 5,000$ NFEs, where N_p stands for the population size and NFE stands for Number of Function Evaluations. An algorithm that is not retained is stopped at a time instance proportional to the completion time, which is chosen to be $\frac{1}{10} \cdot CT$. The *gbest* value at the designated stopping time is used as the *Measure of Merit* to retain an algorithm, after the stopping time elapses. In WTA1 only one algorithm is retained after the stopping time, in particular the algorithm that has the smallest *gbest* value at the designated stopping time. In summary, with WTA1, we run 6 algorithms for $N_p \cdot 500$ NFEs, and only the algorithm with the smallest *gbest* value at the designated stopping time was allowed to run up to $N_p \cdot 5,000$ NFEs. The WTA methodology is fully explained in [7] where other WTA algorithmic variants are introduced and discussed.

3. EXPERIMENTAL RESULTS

In this section, we assess the performance of WTA1 used the set of benchmark functions considered in [2]. In the experiments, the dimensionality of the function D is either 10 or 30 and the initial population size varies from D,$2.5D$,$5D$ to $10D$. We compared the mean and standard deviation of the *gbest* value that WTA1 and the constituent algorithms produce over a number of runs, corresponding to different initializations of the population. In all the experiments conducted, with respect to the mean *gbest* value, WTA1 outperformed any constituent algorithm 77.6% of the cases (i.e., the mean values of WTA1 are smaller than the mean of constituent algorithms), while in 58.9% of the cases WTA1 significantly outperformed any constituent algorithm (i.e., the mean values of WTA1 are at least 10 times smaller, in most instances many orders of magnitude smaller than the mean of a constituent algorithm). If we refer to the standard deviation values, then WTA1 outperformed any constituent algorithm 71.2% of the time, while in 50.4% of the cases WTA1 significantly outperformed any constituent algorithm.

Table 1: Average and standard deviation of gbest values of the NA and WTA1 approaches for 50 experiments of each benchmark problem when $\mathbf{D = 10, N_p = 10}$. Boldfaced entries in the table correspond to NA results that are statistically significantly different, at 0.05 significance level, from the WTA1 results

Fun.	NA	WTA1
f_1	0 ± 0	0 ± 0
f_2	$2.28e-01 \pm 7.61e-01$	$7.39e-01 \pm 1.72$
f_3	$7.11e-16 \pm 1.44e-15$	$7.11e-16 \pm 1.44e-15$
f_4	$1.02e-03 \pm 4.36e-03$	$1.41e-03 \pm 5.74e-03$
f_5	0 ± 0	0 ± 0
f_6	$8.53e-16 \pm 4.46e-15$	$8.53e-16 \pm 4.5e-15$
f_7	0.06 ± 0.24	0.10 ± 0.30
f_8	$\mathbf{120.55 \pm 101.72}$	293.80 ± 140.43
f_9	$7.11e-16 \pm 1.44e-15$	$7.11e-16 \pm 1.44e-15$
f_{10}	$3.69e-10 \pm 2.61e-09$	$3.69e-10 \pm 2.6e-09$
f_{11}	0 ± 0	0 ± 0
f_{12}	0.85 ± 2.11	1.4 ± 3.29
f_{13}	$\mathbf{0.86 \pm 1.48}$	1.54 ± 3.28
f_{14}	$5.81e+02 \pm 1.93e+02$	$6.49e+02 \pm 2.53e+02$
f_{15}	22 ± 41.85	38 ± 60.24
f_{16}	53.09 ± 79.98	69.02 ± 85.19

The results in Table 1 show that WTA1 performs competitively compared with NA for $D = 10$,$N_p = 10$. In only 12.5% of the tests (i.e., a set of 50 runs with a particular benchmark functions of Table 1), NA attains a *gbest* value that is statistically different (better) than the WTA1 at the 0.05 significance level. Similar observations are obtained for other dimension and population size combinations.

4. DISCUSSION

In the late 1990's an approach called Racing Algorithms has been proposed to solve the model selection problem in classification and function approximation problems [3]. In [5] the authors applied two racing algorithms, F-Races and A-Races, to evaluate the goodness of multiple EDAs (Estimation of Distribution Algorithms) on a single optimization problem (Rastrigin problem). They discovered that after 15 restarts they are able to eliminate a good portion of the 50 EDAs that they started with because they are not good performers. Distinct differences of the work in [5] and our work is that we run the algorithms only once to determine their goodness in solving a particular problem and we use a non-statistically based, simple test to determine their goodness. A WTA methodology that is more statistically based appears to be a worthy future pursuit.

5. ACKNOWLEDGMENTS

The authors acknowledge the partial support from NSF grants: 0525429, 0717680, 0837332, 0806931, 0963146. Abigail Fuentes is acknowledged for contributing some of the code for this work.

6. REFERENCES

[1] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar. Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, 13:526–553, 2009.

[2] J. J. Liang, A. K. Qin, and P. N. Suganthan. Comprehensive learning praticle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10:281–295, 2006.

[3] O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems*, pages 59–66, 1994.

[4] F. van den Bergh and A. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8:225– 239, 2004.

[5] B. Yuan and M. Gallagher. Statistical racing techniques for improved empirical evaluation of evolutionary algorithms. In *PPSN*, pages 172–181, 2004.

[6] J. Zhang and A. C. Sanderson. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13:945–958, 2009.

[7] T. Zhang, G. Borrero, and M. Georgiopoulos. A Winner-Take-All methodology: Finding the best evolutionary algorithm for the global optimization of functions. Technical report, University of Central Florida, 2012.

[8] Q. Zheng, Y. Fan, Z. Shi, and Y. Wang. Adaptive inertia weight particle swarm optimization. In *Proceedings of Artificial Intelligence and Soft Computing - ICAISC 2006*, volume 4029, pages 450–459. Springer Berlin/Heidelberg, 2006.