Fitness Tracking Based Evolutionary Programming: A Novel Approach for Function Optimization

Md. Tanvir Alam Anik
Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology
Dhaka-1000, Bangladesh
tanviranik@gmail.com

Saif Ahmed
Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology
Dhaka-1000, Bangladesh
iconicsaif@gmail.com

Md. Monirul Islam
Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology
Dhaka-1000, Bangladesh
mdmonirulislam@cse.buet.ac.bd

ABSTRACT
In order to achieve a satisfactory optimization performance by evolutionary programming (EP), it is necessary to ensure proper balance between exploration and exploitation. It is obvious that one single mutation operator is not the answer. Moreover, early loss of genetic diversity causes premature trapping around locally optimal points of the fitness landscape. This paper presents a fitness tracking based evolutionary programming (FTEP) algorithm incorporating a fitness tracking scheme to find the locally trapped individuals and treat them in a different way so that they are able to improve their performance. FTEP also incorporates several mutation operators in one algorithm and employs a self-adaptive strategy to gradually self-adapt the mutation operators in order to apply an appropriate mutation operator on the individual based on its need. A test-suite of 25 functions has been used to evaluate the performance of FTEP.

Categories and Subject Descriptors
G.1.6 [**Mathematics of Computing**]: Optimization—*Global Optimization*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms
Algorithms, Performances

Keywords
evolutionary programming, mutation, stagnant population, fitness tracking

1. INTRODUCTION
Evolutionary algorithms, such as evolutionary programming (EP), evolution strategies (ESs) and genetic algorithms (GAs) have been very successful in solving many optimization problems. The basic difference between EP (or ESs) and GAs is that: Both EP and ES use only mutation operator to produce offspring while GAs use both crossover and mutation operators. Since mutation is the main operator in EP, a number of innovative mutation operators e.g. Gaussian mutation [2], Cauchy mutation [1], Lévy mutation [3] have been proposed to improve the performance of EP. In this paper, we have introduced a fitness tracking based EP (FTEP) algorithm that incorporates Fitness Tracking Process to avoid premature convergence. FTEP maintains a pool of different yet effective mutation operators integrating their advantages together in order to overcome the shortcomings of a pure strategy.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

2. METHOD
The proposed FTEP scheme uses fitness tracking to track the performance of an individual i.e. at what degree the individual is able to optimize its fitness. A pool of mutation operators is associated with FTEP to provide necessary search step sizes as the evolutionary process progresses. Each mutation operator as a probability associated with it which is same for all at the beginning. Each individual is assigned two mutation operators, chosen from the mutation pool by fitness proportionate selection. Therefore, two children are generated from each individual and the better candidate is selected as the offspring. At each generation, successful mutation rate of the mutation operators are recorded. After a predefined number of consecutive generations (known as learning period *LP*), the following tasks are done: a) Fitness optimization information of each individual is recorded and based on that information, less-fit (stagnant) individuals are separated from the main population. b) Each stagnant individual is further mutated by two differential mutation operators to generate two children as discussed in section 2.2 and the better candidate replaces its corresponding stagnant parent in the main population. c) Successful mutation rate is used to increase or decrease the probability of each mutation operator so that better mutation operators are rewarded and others are penalized. The necessary details of the components of FTEP are given in the following subsections.

2.1 Fitness Tracking Procedure
The central component of our proposed FTEP system is the Fitness Tracking scheme for separating the under-performing individuals. Although there exists some algorithms like Crowding, Fitness sharing that penalize similar individuals and promote diverse ones, fitness tracking scheme allows the apparently under-performing individuals to grow. In fact, fitness tracking keeps the optimization record of the individuals to distinguish the best-fit individuals from the less-fit ones as they evolve through generations. The tracking procedure continues for *LP* generations. Each individual has a fitness tracker associated with it which is initialized by the fitness value of its parent at the beginning of *LP*. During each generation of *LP,* the fitness of each individual is recorded by the fitness tracker associated with it. Thus after the completion of *LP* generations, a sequence of fitness values starting from root parent individual to the current offspring is obtained for each individual. This sequence of fitness values

associated with the individual therefore represents its improvement within the learning period. From this sequence of fitness values, optimization record of each individual can be obtained and based on that record stagnant individuals are detected and separated from the main population.

2.2 Stagnant Population Replacement Strategy

FTEP introduces differential mutation operators, picked from differential evolution (DE) literature instead of distribution based mutation operators, to treat the stagnant individuals so that they can improve themselves. It chooses DE/target-to-local-best/1 and DE/target-to-global-best/1, two differential mutation strategies from [4]. DE/target-to-global-best/1 is exploitative in nature and possesses higher ability to converge to the near-optimal solutions as quickly as possible. Thus it performs better for functions with a few local optima. On the other hand, DE/target-to-local-best/1 is explorative in nature and searches every region of the feasible search space. Thus, it is best suited for functions with many local optima. While choosing the local neighborhood for DE/target-to-local-best/1, the neighbor individuals are chosen randomly not considering their geographical nearness or similar fitness values. Details about the behavior of these differential mutation operators are one of the focuses of our future work. Each stagnant individual is mutated by these two differential mutation operators and the better candidate replaces its stagnant parent.

2.3 Self-Adaptive Mutation Pool

FTEP maintains a mutation pool containing a variety of mutation operators with effective yet diverse characteristics. During evolution, with respect to each individual in the current population, two mutation operators will be chosen from this mutation pool according to the probability learned from its previous experience of generating promising solutions and applied to perform the mutation operation. The more successfully the mutation operators behaved in previous *LP* generations to generate promising solutions, the higher the probability they will have to be chosen in the current generation. An ideal mutation pool should contain effective mutation operators having distinct capabilities when dealing with a specific problem at different stages of evolution. The member mutation operators of the mutation pool of FTEP are: Gaussian, Cauchy, four Lévy mutations (with its scaling parameter=1, 1.3, 1.5, 1.7), mean mutation operator (MMO) [6] and adaptive mean mutation operator (AMMO) [6]. Details about the construction of mutation pool and its member mutation operators are the topics for our future study.

3. RESULTS AND DISCUSSION

We have chosen the set of benchmark functions provided by CEC 2005 special session [5] to present a simple experimental study. Table 1 shows the obtained error results from the experiments for FTEP with problem dimension set to 30 and number of function evaluations set to 300,000 and. The error is computed as (Error = $f(x) - f(x^*)$), where $f(x)$ is the obtained solution by the algorithm and $f(x^*)$ is the already known global optimum for a particular benchmark function. It is apparent from the table that FTEP achieves excellent optimization performance for both unimodal (f_1-f_5) and multimodal functions (f_6-f_{14}). As the hybrid composition functions (f_{14}-f_{25}) are more challenging and difficult to solve, FTEP has not been able to solve them, but obtains better results if compared to some state-of-the-art works. It will be one of the focuses of our future work to present comparison experiment in more details. The convergence characteristics for four functions have been presented in Figure 1. It is obvious from the figure that FTEP converges smoothly without getting stuck at local minima until it reaches proximity of global minima.

In summary, fitness tracking, stagnant individuals' replacement through differential mutation and self-adaptive mutation pool lead to reliable optimization achieved by FTEP. In future, we are interested to explore this idea in more details and conduct experimental studies for a broader class of problems.

Table 1. Performance on CEC2005 benchmark functions with (over 25 runs)

No.	FTEP Mean Error	No.	FTEP Mean Error	No.	FTEP Mean Error
f_1	3.70e-02	f_{10}	6.52e+01	f_{19}	8.24e+02
f_2	6.10e-01	f_{11}	1.21e+01	f_{20}	8.27e+02
f_3	9.91e+05	f_{12}	9.31e+02	f_{21}	5.00e+02
f_4	2.55e+03	f_{13}	2.46e+00	f_{22}	5.12e+02
f_5	2.71e+03	f_{14}	1.24e+01	f_{23}	5.34e+02
f_6	1.59e+01	f_{15}	2.03e+02	f_{24}	2.11e+02
f_7	7.58e-02	f_{16}	9.24e+01	f_{25}	2.11e+02
f_8	2.07e+01	f_{17}	9.45e+01		
f_9	1.28e+01	f_{18}	8.27e+02		

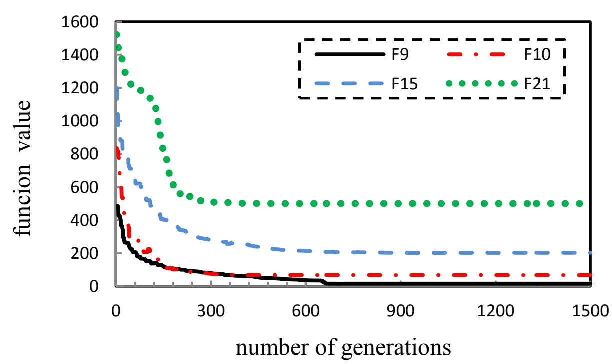

Figure 1. Convergence characteristics for f_9, f_{10}, f_{15} and f_{21}.

4. REFERENCES

[1] X. Yao, Y. Liu and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82-102, July 1999.

[2] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Comput.*, 1(1):1–23, 1993.

[3] C. Lee and X. Yao. Evolutionary programming using mutations based on the Lévy probability distribution. *IEEE Transactions on Evolutionary Comput.*, 8(1):1-13, February 2004.

[4] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.*, 13(2):398–417, April 2009.

[5] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization[online] . Available: http://www.ntu.edu.sg/home/EPNSugan

[6] K. Chellapilla. Combining mutation operators in evolutionary programming. *IEEE Transactions on Evolutionary Computation*, 2(3):91-96, September 1998.