Swarm and Evolutionary Computation 9 (2013) 58–68

Contents lists available at SciVerse ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Adaptive evolutionary programming with *p*-best mutation strategy

Swagatam Das [a,*], Rammohan Mallipeddi [c], Dipankar Maity [b]

[a] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata 700108, India
[b] Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India
[c] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 7 March 2012
Received in revised form
5 November 2012
Accepted 8 November 2012
Available online 29 November 2012

Keywords:
Evolutionary programming
Real-parameter optimization
Mutation
Parent selection
Evolutionary algorithms

ABSTRACT

Although initially conceived for evolving finite state machines, Evolutionary Programming (EP), in its present form, is largely used as a powerful real parameter optimizer. For function optimization, EP mainly relies on its mutation operators. Over past few years several mutation operators have been proposed to improve the performance of EP on a wide variety of numerical benchmarks. However, unlike real-coded GAs, there has been no fitness-induced bias in parent selection for mutation in EP. That means the *i*-th population member is selected deterministically for mutation and creation of the *i*-th offspring in each generation. In this article we propose a *p*-best mutation scheme for EP where any one from the p ($p \in [1,2,\ldots,\mu]$, where μ denotes population size) top-ranked population-members (according to fitness values) is selected randomly for mutation. The scheme is invoked with 50% probability with each index in the current population, i.e. the *i*-th offspring can now be obtained either by mutating the *i*-th parent or by mutating a randomly selected individual from the p top-ranked vectors. The percentage of best members is made dynamic by decreasing p in from $\mu/2$ to 1 with generations to favor explorative behavior at the early stages of search and exploitation during the later stages. We investigate the effectiveness of introducing controlled bias in parent selection in conjunction with an Adaptive Fast EP (AFEP), where the value of a strategy parameter is updated based on the previous records of successful mutations by the same parameter. Comparison with the recent and best-known versions of EP over 25 benchmark functions from the CEC (Congress on Evolutionary Computation) 2005 test-suite for real-parameter optimization and two other engineering optimization problems reflects the statistically validated superiority of the new scheme in terms of final accuracy, speed, and robustness. Comparison with AFEP without *p*-best mutation demonstrates the improvement of performance due to the proposed mutation scheme alone.

1. Introduction

Evolutionary Programming (EP), originally conceived by Lawrence J. Fogel and his coworkers in early 1960s [1,2], is a stochastic optimization technique similar to the Genetic Algorithms (GAs). However, unlike conventional GAs, it emphasizes on the behavioral linkage between parents and their offspring, rather than seeking to emulate specific genetic operators as observed in nature. In their seminal 1966 book—"Artificial Intelligence through Simulated Evolution" [3], Fogel et al. showed how EP can be used to evolve finite state automata for predicting symbol strings generated through Markov processes and non-stationary time series. Thereafter EP has been associated with the prediction tasks (viewed as a keystone to intelligent behavior) for a long time and it was using finite state machines as the medium of representation. Since early 1990s EP emerged as a continuous function optimizer with real-valued vector representations mainly through the works of Fogel and his colleagues [4,5].

Like EP, Evolutionary Algorithms have been designed in last few decades for the purpose of numerical function optimization. These Evolutionary Algorithms uses a group of particle as a population and in every iteration new particles are generated from the parent particles by some definite means. Some nature inspired algorithms like Particle Swarm Optimization (PSO) [6] exploits the schooling of fish. However, there are also some algorithms like Differential Evolution (DE) [7] do not imitate any natural foraging strategy. Auger et al. proposed a restart CMA evolution strategy (Restart CMA—ES) [8] where the population is increased in each restart. In order to improve the learning strategy of PSO, Zhan et al. proposed Orthogonal Learning Particle Swarm Optimization (OLPSO) [9]. The orthogonal learning strategy in OLPSO guides the particles to move in better a position by using an efficient and promising exemplar. In order to maintain the diversity of a swam and to stymie the premature

* Corresponding author.
E-mail addresses: swagatamdas19@yahoo.co.in, swagatam.das@isical.ac.in (S. Das), mallipeddi.ram@gmail.com (R. Mallipeddi), dipankarmaity1991@gmail.com (D. Maity).

2210-6502/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.swevo.2012.11.002

convergence, Liang et al. proposed Comprehensive Learning Particle swarm Optimization [10] in 2006. In the same year, Brest et al. [11] investigated a self-adaptive DE algorithm (*jDE*) where the control parameters *F* and *Cr* are self-adapted. Qin et al. [12] proposed SaDE which also self adaptively changes the control parameters *F* and *Cr* with simultaneous use of DE/rand/1 and DE/current-to-best/1 mutation strategies. Zhang and Sanderson [13] proposed a new mutation strategy, named DE/current-to-*p*-best which is a generalization of the DE/current-to-best strategy. In the same year, Das et al. [14] proposed a neighborhood based mutation strategy with Differential Evolution to improve the performance of classical DE. They used an improved version of the DE/target-to-best/bin/1 mutation strategy to balance the exploration and exploitation abilities of DE. Thus, the improvement of mutation strategy and the variation of control parameters of DE attracted a great deal of research in recent past. Mallipeddi et al. proposed EPSDE [15] that introduces an ensemble of mutation strategies and control parameters with the DE.

In numerical function optimization, EP uses real-valued vectors directly as individuals. EP emulates behavioral evolution (asexual reproduction using only mutation) rather than genetic evolution, which involves both mutation and crossover operations. One important feature of the EP is its dynamic strategy parameter. Survivor selection in EP is normally implemented by competition within the combined parent and offspring population by using the tournament selection method.

To mutate an individual, mutation operators based on various probability distributions such as Gaussian (CEP) [3], Cauchy (FEP) [16] and Lévy [17] are used. The role of mutation is very important in EP, since mutation is the only operator used to generate new candidate solutions. Gaussian mutation is the classical mutation operator which may fail on multi-modal problems. Cauchy is a special case of the Lévy mutation, which can outperform Gaussian mutation on multimodal problems. However, Cauchy mutation is less effective on unimodal problems. As each mutation operator has its advantages and disadvantages, overall performance of the EP can be improved by using different mutation operators simultaneously or by integrating several mutation operators into one algorithm or by adaptively controlled usage of mutation operators. The idea of integrating several mutation strategies into one algorithm within one population is referred to as mixed mutation strategy [18]. There are different ways to design a mixed mutation strategy [19,20], the earliest of which is a linear combination of Gaussian and Cauchy distributions [21]. Improved Fast EP (IFEP) [16] implements Cauchy and Gaussian mutations simultaneously and generates two offspring; the better one will be chosen to compete with the parent population during the tournament selection stage. The IFEP variants also included the Lévy mutation with various scaling parameters [17]. The idea of IFEP using all the three mutation operators was also investigated in [22]. Zhang and Lu proposed an EP based on Reinforcement Learning (RLEP) [23], where each individual learns its optimal mutation operator based on the immediate and delayed performance of mutation operators. Mutation operator selection is mapped into a reinforcement learning problem. Apart from experimenting with the mutation operator, a few attempts have been made to modify the survivor selection strategy of EP to improve its performance. Instead of following the conventional tournament selection, recently Chen et al. [24] incorporated three survivor selection rules in FEP in order to encourage both fitness diversity and solution diversity. Meanwhile, two solution exchange rules were introduced by them in an attempt to further exploit the preserved genetic diversity.

Unlike several variants of GA, EP so far incorporates no bias in selection of the parents that undergo mutation to produce offspring. In this article we propose a new mutation structure for EP by introducing a fitness-induced bias in the process of parent selection. Under this structure, the *i*-th offspring is produced either by mutating the *i*-th parent solution vector or by mutating any member (randomly picked up) from the top $p(p \in [1,2,\ldots,\mu]$, where μ is the population size) solutions from the current population, which is ranked according to the fitness values associated with the solution vectors. The parameter *p* is decreased from $\mu/2$ to 1 with generations so that initially parents can be selected from a large superior portion of the population (during first few generations the superior half, to be more precise), but gradually the optional parental archive reduces to only the best individual of the population. This simple strategy favors rapid exploration of the functional landscape at the onset of the search but gradually switches to exploitative detailed search around the best member during the final stages. The *p*-best mutation scheme can be integrated with any EP-variant like FEP, LEP, IFEP etc., and as revealed by our experiments, in all cases, it results into a significant improvement of performance over the original EP-variant. Here we, however, report the experiments conducted by integrating this mutation structure with an Adaptive Fast Evolutionary Programming (AFEP) [25]. Adaptive Evolutionary Programming (AEP) was recently proposed in [25] in conjunction with constrained optimization problems. AEP is similar to the EP, except the initialization and the adaptation of the strategy parameter values. In conventional EP the strategy parameters η are initialized to a constant (in CEP $\eta = 3$) and η is updated based on τ and τ' values (to be detailed in the next section). In AEP, the initialization of η values is done based on the search range of the decision variables and the way η values are updated is based on the previous record of successful mutation with the same parameter over past few generations (called 'learning period').

Performance of the *p*-best AFEP algorithm is compared with six other recent and best-known EP-variants, including AFEP without *p*-best mutation, on a test-suite of 25 benchmark functions taken from Congress on Evolutionary Computation (CEC) 2005 special session on real-parameter optimization, in 10 and 30 dimensions. The *p*-best AFEP is also tested on two engineering optimization problems involving the design of spread-spectrum radar polyphase codes and frequency modulated sound wave synthesis. Such comparison indicates that the *p*-best mutation scheme enjoys a statistically superior performance in comparison to the most state-of-the-art EP-variants over a wide variety of single-objective bound constrained optimization problems.

Rest of the paper is organized in the following way. The basic EP and its variants are outlined in Section 2. Section 3 describes the AFEP and *p*-best AFEP in sufficient details. Section 4 describes the experimental set-up, presents the comparison results, and discusses their implications. Finally Section 5 concludes the paper and unfolds a few important future research issues.

2. Evolutionary programming–an outline

In this section we provide an outline of EP family of algorithms applied to the Function Optimization Problems (FOPs). The task of function optimization is basically a search for such the parameter vector $\vec{x}^*$, which minimizes an objective function $f(\vec{x})(f: \Omega \subseteq \Re^n \to \Re)$ i.e. $f(\vec{x}^*) < f(\vec{x})$ for all $\vec{x} \in \Omega$, where Ω is a non-empty large finite set serving as the domain of the search. For unconstrained optimization problems $\Omega = \Re^n$. Since $\max\{f(\vec{x})\} = -\min\{-f(\vec{x})\}$, the restriction to minimization is without loss of generality. In general the optimization task is complicated by the existence of non-linear objective functions with multiple local minima. A local minimum $f_\ell = f(\vec{x}_\ell)$ may be defined as: $\exists \varepsilon > 0 \forall \vec{x} \in \Omega : ||\vec{x} - \vec{x}_\ell|| < \varepsilon \Rightarrow f_\ell \le f(\vec{x})$, where $||.||$ indicates any *p*-norm distance measure.

Depending on the mutation operator used to produce variation in the population, different versions of EP such as CEP and FEP