Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

Which algorithm should I choose: An evolutionary algorithm portfolio approach

Shiu Yin Yuen[a,*], Chi Kin Chow[a], Xin Zhang[b], Yang Lou[a]

[a] *Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China*
[b] *College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin, China*

A R T I C L E I N F O

Article history:
Received 18 July 2012
Received in revised form 6 May 2015
Accepted 14 December 2015
Available online xxx

Keywords:
Multiple evolutionary algorithms
Multi-method search
Algorithm portfolio
Performance prediction
Algorithm choice
Synergy
Scalability
Real world application

A B S T R A C T

Many good evolutionary algorithms have been proposed in the past. However, frequently, the question arises that given a problem, one is at a loss of which algorithm to choose. In this paper, we propose a novel algorithm portfolio approach to address the above problem for single objective optimization. A portfolio of evolutionary algorithms is first formed. Covariance Matrix Adaptation Evolution Strategy (CMA-ES), History driven Evolutionary Algorithm (HdEA), Particle Swarm Optimization (PSO2011) and Self adaptive Differential Evolution (SaDE) are chosen as component algorithms. Each algorithm runs independently with no information exchange. At any point in time, the algorithm with the best predicted performance is run for one generation, after which the performance is predicted again. The best algorithm runs for the next generation, and the process goes on. In this way, algorithms switch automatically as a function of the computational budget. This novel algorithm is named Multiple Evolutionary Algorithm (MultiEA). The predictor we introduced has the nice property of being parameter-less, and algorithms switch automatically as a function of budget. The following contributions are made: (1) experimental results on 24 benchmark functions show that MultiEA outperforms (i) Multialgorithm Genetically Adaptive Method for Single Objective Optimization (AMALGAM-SO); (ii) Population-based Algorithm Portfolio (PAP); (iii) a multiple algorithm approach which chooses an algorithm randomly (RandEA); and (iv) a multiple algorithm approach which divides the computational budget evenly and execute all algorithms in parallel (ExhEA). This shows that it outperforms existing portfolio approaches and the predictor is functioning well. (2) Moreover, a neck to neck comparison of MultiEA with CMA-ES, HdEA, PSO2011, and SaDE is also made. Experimental results show that the performance of MultiEA is very competitive. In particular, MultiEA, being a portfolio algorithm, is sometimes even better than all its individual algorithms, and has more robust performance. (3) Furthermore, a positive synergic effect is discovered, namely, MultiEA can sometimes perform better than the sum of its individual EAs. This gives interesting insights into why an algorithm portfolio is a good approach. (4) It is found that MultiEA scales as well as the best algorithm in the portfolio. This suggests that MultiEA scales up nicely, which is a desirable algorithmic feature. (5) Finally, the performance of MultiEA is investigated on a real world problem. It is found that MultiEA can select the most suitable algorithm for the problem and is much better than choosing algorithms randomly.

1. Introduction

Rapid advances in Evolutionary Computation (EC) have been witnessed in the past two decades. There are now many powerful Evolutionary Algorithms (EAs) that are applied to scientific and engineering applications to find good quality solutions for challenging optimization problems. Famous examples include Genetic Algorithm (GA), Evolution Strategy (ES), Evolutionary Programming (EP), Particle Swarm Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO), Artificial Immune System (AIS), Cultural Algorithm (CA), Estimation of Distribution Algorithm (EDA), Artificial Bee Colony algorithm (ABC), Biogeography Based Optimization (BBO), and others [1].

In spite of the proliferation of algorithms that use the evolution metaphor, for general users dealing with an optimization scenario, there is little readily available guideline of *which algorithm to choose*. Frequently, one resorts to words of mouth or fame of the algorithm, or try it out one by one in an exhaustive manner. The

* Corresponding author. Tel.: +86 85234427717.
E-mail address: kelviny.ee@cityu.edu.hk (S.Y. Yuen).

http://dx.doi.org/10.1016/j.asoc.2015.12.021
1568-4946/© 2015 Published by Elsevier B.V.

problem is compounded by the fact that individual algorithms will need parameter tuning to obtain the best performance, which is computational expensive or even prohibitive [2]. The current state of affairs motivates this paper.

Usually an algorithm has a standard recommended set of parameters defined by the researchers. This set of parameters is usually arrived at after many tests on benchmark functions and practical applications. Thus for each algorithm, we use the recommended set of parameters and do not attempt the challenging problem of parameter tuning and control [2]. Instead, we believe that the multiple algorithms can be complementary: an algorithm which does not work well on one problem will be replaced by an algorithm that works well for it. So the problem is how to select algorithms.

Note also that the question of choice of algorithm should be a function of the computational budget. For example, one algorithm may converge fast to a shallow local optimum, another may converge slower but to a deeper local optimum given enough time, still another may converge the slowest but eventually reach the global optimum. Which algorithm should one choose? If fitness evaluations are expensive, then only a small computational budget is allowed and the first one should be chosen. If fitness evaluations are relatively inexpensive or we have a design problem such that we can tolerate longer runs, then the second algorithm should be chosen. Finally, if fitness evaluations are cheap and we aim at solving a scientific problem of which finding the global optimum is essential, then the third algorithm should be chosen.

An algorithm portfolio approach is advocated to tackle the problem. Our conceptual framework is simple: (1) put promising EAs together in a portfolio; (2) an initialization is conducted in which each algorithm is run for some number of generations until there is a change in fitness; (3) use a predictive measure to predict the performance of each algorithm at the nearest common future point; (4) select the algorithm which has the best predicted performance to run for one generation; and (5) repeat step (3) and (4) until a given computational budget is reached.

Note that as the algorithm which has the best predicted performance may be different at different stages of the search, our approach will switch from one algorithm to another automatically and seamlessly.

In a nutshell, we propose to choose, at any point of the search, the algorithm which has the best predicted performance to run (for one generation). A typical scenario of running our algorithm is that after some trials in which each algorithm runs in parallel and interacts indirectly, an algorithm which has the best predicted performance by a considerable margin stands out, and only it is run for quite some time. If it is a very good algorithm that excels in small, medium and large budgets, then only it will run from then on. However, if it is an algorithm that converges fast to a local optimum, as discussed above, then it will run for awhile and gradually the predicted performance will not be as good compared with other algorithms. At which time a second algorithm, which has a better predicted performance, will take over. Like changes would occur as the search progresses.

A novel online performance prediction metric is proposed to advise which algorithm should be chosen to generate the next new generation of solutions. The metric is *parameter-less* – it does not introduce any new control parameter to the system – thus avoiding the difficult parameter tuning and control problem [2]. We name our algorithm Multiple Evolutionary Algorithm (MultiEA). It is designed for single objective optimization.

We choose four algorithms to compose the portfolio of MultiEA. They are (1) Covariance Matrix Adaptation Evolution Strategy (CMA-ES); (2) History driven Evolutionary Algorithm (HdEA); (3) Particle Swarm Optimization (PSO2011); and (4) Self adaptive Differential Evolution (SaDE).

These algorithms are chosen because they represent current state of the art methods:

CMA-ES [3] is one of the most powerful EAs available. It adapts its search strategy by evolving the covariance matrix of the current solutions. The idea is to increase the variance of the directions in which the search is successful and vice versa. A nice fundamental property which is unique to CMA-ES is its invariance to linear transformations of the search space.

HdEA [4] is a novel EA that uses the entire search history to make decision. The history is stored in a binary space partitioning (BSP) tree. When a new solution is generated by the EA, it finds the region it is contained within efficiently by traversing the tree to its leaf node. The local gradient is approximated by the local history information in the tree and a parameter-less gradient descent is executed to find a mutated solution. The performance of HdEA is tested on thirty four benchmark functions with dimensions ranging from 2 to 40. It outperforms eight benchmark evolutionary algorithms, which includes a Real Coded Genetic Algorithm (RCGA), classic DE, two improved DEs, CMA-ES, two improved PSOs, and EDA.

PSO introduces a new search paradigm which simulates the swarm behavior in birds and other animals. The velocity of each individual particle is modulated by both the historical local best position found by the particle and the historical global best position found by the whole swarm. Since its inception, two revised “standards” have been promulgated: PSO2007 and PSO2011. PSO2007 abandons the global best position concept. Instead, each individual chooses its own set of informants, and follows its local best as well as the global best amongst the informants. The idea is to distribute the search effort to avoid prematurely converging on the global best. It performs much better than the classic PSO. The PSO2011 [5] further improves PSO2007 by making the PSO more immune to linear transformations in the search space.

DE is also a powerful EA. Recently improved DE variants have won many EC competitions [6]. A fundamental idea in DE is to use the sum of a vector and the scaled difference of another two vectors to form the mutant vector. The recently proposed SaDE [7] employs four popular DE mutation strategies. During any one generation, DE maintains a set of trial vectors in its population pool. For each trial vector, a dice is rolled to select a DE strategy. Initially, all strategies have the same probability of being selected. Records are kept of the outcome of employing the strategies. A strategy is regarded as successful if it produces an offspring that survives into the next generation; otherwise, it is regarded as a failure. The success probability for each strategy is computed for the previous LP generations, where LP is the user defined learning period. The probability of selecting a particular strategy is proportional to its success probability. This probability is lower bounded by a small constant to preclude a strategy of attaining zero success probability and henceforth eliminated from all future considerations. SaDE shows remarkable performance. It outperforms classic DE and several recent adaptive DEs.

Though these are good EAs, it is also well acknowledged that they may not perform well for certain functions. For example, DE is known to be doing less well for non-separable functions [8] of which CMA-ES excels, and it is well known that CMA-ES performs poorly for the Rastrigin function if no artificial remedy such as enlarging the population is taken [9]. Finally, by the well known no free lunch (NFL) theorems, it is unlikely to find an algorithm which would do well for all problems [10,11]. So it would be of interest to see whether a portfolio approach can be successful in identifying the *correct* algorithm for a problem.

Experiments are conducted on 24 benchmark functions [12]. The effectiveness of MultiEA is demonstrated by comparing with four multiple algorithm approaches. They are (i) Multialgorithm Genetically Adaptive Method for Single-Objective Optimization