Multifactorial evolution: Toward evolutionary multitasking Research Report

Yuanjianhua

Abstract

In a time,evolutionary algorithms has typically been used efficiently to solve a single optimization problem.However with development of technology,more and more task need be solved simultaneously.Besides the implicit parallelism of population-based search, no attempt has yet been made to multitask, i.e., to solve multiple optimization problems simultaneously using a single population of evolving individuals. Accordingly, the author introduces evolutionary multitasking as a new paradigm in the field of optimization and evolutionary computation. They first formalize the concept of evolutionary multitasking and then propose an algorithm to handle such problems. The methodology is inspired by bio-cultural models of multifactorial inheritance, which explain the transmission of complex developmental traits to offspring through the interactions of genetic and cultural factors. Furthermore, They develop a cross-domain optimization platform that allows one to solve diverse problems concurrently.Through this way,different tasks can evolve simultaneously and exchange information with each other.This are the main point of MFEA.

Keywords： Evolutionary Multitasking, Memetic Computation, Continuous Optimization, Discrete Optimization.

1 Introduction

Evolutionary algorithms(EAs) are not like other algorithms，they are optimization metaheuristics which work on Darwinian principles of Natural Selection or Survival of the Fittest[1].It starts with a population of individuals that undergo computational analogues of sexual reproduction and mutation to produce a generation of offspring.This procedure reiterates itself with the aim of finding the best solution.Up to now,conventional EAs are normally designed and employed to solve optimization task.However,in our real world,there are a plenty of tasks that should be tackled meanwhile,multifactorial optimization is an evolutionary multitasking paradigm which could search multiple spaces corresponding to different tasks.And inspiered by models of multifactorial inheritance,the author proposed the multifactorial evolutionary algorithm(MFEA).Multitask optimization is the study of solving multiple optimization problems (tasks) simultaneously and thus independently Multitask optimization is the study of solving multiple optimization problems (tasks) simultaneously and thus independently to improve the performance of solving each task. It works on the assumption that some common useful knowledge exists in solving a task, and then the useful knowledge gained in solving this task may help in solving another task that is related to it. In practical applications, related optimization tasks are common. In fact, its makes full use of the implicit parallelism based on population search.

2 Related works

Generally speaking, conventional optimization problems can be classified into two categories: 1) single-objective optimization (SOO) problems, and 2) multi-objective optimization (MOO) problem. A SOO problem is to minimize or maximize a single objective function[2], where every point in the search space maps to a single scalar objective value. An MOO problem is referred to minimizing or maximizing two or more conflicting objective functions simultaneously[3], where every point in the search space maps to multiple objective function values. MFO problems have proposed as the third category in optimization community, which could be given to the problems that contains multiple optimization tasks[4].Unlike SOO and MOO, the purpose of MFO is to seek the optimal solution for each task while multiple optimization tasks are carried out at the same time.It has a marvels different from the MOO problem.The following is a detailed description.

2.1 Multifactorial Optimization

Without loss of generality,given K target minimazation tasks,the definition of an MFO problem is provided as follows:

$$\{\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_k}\} = \{argminT_1(X_1), argminT_2(X_2), ..., argminT_k(X_k)\} \tag{1}$$

where x_i is a feasible solution of the i-th task T_i (i = 1, 2, ..., K). Ti itself could be a SOO or MOO problem.In order to design evolutionary solvers for MFO,the author define a set of properties for every individual p_i,where i $\in \{1, 2, ..., |P|\}$,in a population P.They are the follows:

Definition 1 (Factorial Cost): For a given task T_j, the factorial cost Ψ_j^i individual pi is given by $\Psi_j^i = \lambda \cdot \delta_j^i + f_j^i$;where λ is a large penalizing multiplier, f_j^i and δ_j^i are the objective value and the total constraint violation, respectively, of p_i with respect to T_j. Accordingly, if p_i is feasible with respect to T_j (zero constraint violation), we have $\Psi_j^i = f_j^i$.

Definition 2 (Factorial Rank):The *factorial rank* r_j^i of p_i on task T_j is simply the index of p_i in the list of population members sorted in ascending order with respect to Ψ_j.

Definition 3 (Scalar Fitness):The list of factorial ranks $\{r_1^i, r_2^i, ..., r_k^i\}$ of an individual p_i is reduced to a *scalar fitness* φ_i based on its best rank over all tasks; i.e., φ_i = 1 / $\min_{j\in\{1,2,...,k\}}$ $\{r_j^i\}$.

Definition 4 (Skill Factor):The *skill factor* τ_i of p_i is the one task, amongst all other tasks in MFO, on which the individual is most effective, i.e., τ_i =$argmin_j\{r_j^i\}$,where $j \in \{1, 2, .., K\}$.

2.2 MFO and MOO:Similarities and distinctions

Both MOO and MFO problems involve the optimization of multiple objective functions, however, they are two distinct search paradigms. MOO focuses on efficiently resolving conflicts among competing objective function in one task. There typically exists a single genetic express space in an MOO problem. The decision variables of all objective functions of MOO problems are contained within that single space. As for MFO, the purpose is to share some good traits among distinct (but possibly similar) tasks by levering the parallelism of population-based search. In an MFO problem, there are multiple heterogeneous genetic express spaces. Solving a MOO task typically yields a set of Pareto optimal solutions that provide the best trade-offs among all objective functions. Differently, solving an MFO problem obtains a solution sets that contain the optimal solutions corresponding to each task. An MOO problem can be considered as a special MFO problem with single genetic express space, which reveals the greater generality of MFO paradigm.

3 Method

The MFEA is inspired by the bio-cultural models of multifactorial inheritance. As the working of the algorithm is based on the transmission of biological and cultural building blocks (genes and memes)[5-6] from parents to their offspring, the MFEA is regarded as belonging to the realm of memetic computation[7-8]. The main procedure of MFEA are follows.

Procedure 1 Basic structure of the MFEA.

1.Generate an initial population of individuals and store it in current-pop(P)
2.Evaluate every individual with respect to every optimization task in the multitasking environment
3.Compute the skill factor(τ)of each individual
while *stopping conditions are not satisfied* **do**
  i. Apply genetic operators on $current - pop$ to generate an $offspring - pop(C)$.Refer to Algorithm 2
  ii. Evaluate the individuals in $offspring - pop$ for selected optimization tasks only(see Algorithm3)
  iii. Concatenate $offspring - pop$ and $current - pop$ to form an intermediate-pop(P∪ C)
  iv. Update the scalar fitness(φ) and skill factor(τ)of every individual in $intermediate - pop$
  v. Select the fittest individuals from $intermediate - pop$ to form the next $current - pop(P)$
end

3.1 Population initialization

Assume that in K optimization tasks to be performed simultaneously, the dimensionality of the j^th task is given by D_j. Accordingly, we define a unified search space with dimensionality ($D_{multitask}$) equal to $max_j\{D_j\}$. During the population initialization step, every individual is thus endowed with a vector of $D_{multitask}$ random variables (each lying within the fixed range [0, 1]).

3.2 Genetic mechanisms

A key feature of the MFEA is that certain conditions must be satisfied for two randomly selected parent candidates to undergo crossover. The principle followed is that of nonrandom or assortative mating[9-10], which states that individuals prefer to mate with those belonging to the same cultural background. *In the MFEA, the skill factor (τ) is viewed as a computational representation of an individual's cultural bias*. Thus, two randomly selected parent candidates can freely undergo crossover if they possess the same skill factor. Conversely, if their skill factors differ, crossover only occurs as per a prescribed *random mating probability (rmp)*, or else mutation kicks in.The crossover steps according to these rules are provided in Algorithm 2;

Procedure 2 Assortative mating.

Consider two parent candidates p_a and p_b randomly selected from $current - pop$
1.Generate a random number $rand$ between 0 and 1.
if *($\tau_a == \tau_b$) or (rand $\leqslant$ rmp)* **then**
  i.Parents p_a and p_b crossover to give two offspring individuals c_a and c_b
else
  i.Parents p_a is mutated slightly to give an offspring c_a
  ii.Parents p_b is mutated slightly to give an offspring c_b
end

3.3 Selective evaluation

In multifactorial inheritance, vertical cultural transmission is a mode of inheritance that operates side by side with biological inheritance in a manner such that the phenotype of an offspring gets directly affected by the phenotype of its parents[9-12]. The computational analogue of the aforementioned phenomenon is realized in the MFEA by allowing offspring to imitate the skill factor (cultural trait) of any one of their parents. This feature, labeled as selective imitation, is achieved numerically by following the steps in Algorithm 3.

Procedure 3 Vertical cultural transmission via selective imitation.

An offspring 'c' will either have two parents (p_a and p_b) or a single parent (p_a or p_b) - see Algorithm 2.
if *('c' has 2 parents)* **then**
  i.Generate a random number $rand$ between 0 and 1
  if $rand \leqslant 0.5$ **then** 'c' imitates $p_a \rightarrow$ The offspring is evaluated only for task τ_a(the skill factor of p_a);
  else 'c' imitates $p_b \rightarrow$ The offspring is evaluated only for task τ_b(the skill factor of p_b);
else
  'c' imitates its single parent $\rightarrow$ The offspring is evaluated only for that task which is its parent's skill factor.
end
Factorial costs of 'c' with respect to all unevaluated tasks are artificially set to ∞ (a very large number).

3.4 Selection operation

As shown in Algorithm 1, the MFEA follows an elitist strategy which ensures that the best individuals survive through the generations.

3.5 Summarizing the salient features of the MFEA

In the MFEA, the transfer of genetic material is facilitated by allowing different skill groups to communicate with each other in a controlled manner, via occasional chromosomal crossover. This is achieved implicitly by two components of the algorithm operating in concert, namely, a) the rmp, which allows individuals with distinct skill factors to mate with some probability, thereby creating a multicultural environment for offspring to be reared in, and b) the fact that the generated offspring can then randomly select a parental skill factor for imitation (see Algorithm 3).

4 Implementation details

4.1 Comparing with released source codes

Baed the author's idea, we have some new thought on the two task MFEA .For convention two multifactorial EA,it just use the imply common knowledge between two tasks to facilitate together getting to the global optimum .We propose a new way that we could build a new function by the best-Evaluation of each 10 generation.And taking this function to promote the converge speed of other two tasks.This has make the a good effect in finding the global optimum.However, when we look at the result of this method,we will find it has a well-done effective in someplace.In this way,we begin to search for a good way to help these two task to do find their optimum effectively.We first think about the most population method in machine learning to find the optimum value and the most common method is gradient descent.However,these function in the experiment do not have a gradient.In this place, we begin to think that if we could through insert a new function with strong convexity which is easy to get their optimum by gradient descending.Based in this thought,I make a new function which is learned by using the polynomial fitting.The detail are showed in code.

```
function obj = Linear(var,~)
    filename = "MFEA/Data/data.mat";
    data = load(filename);
    data = data.data_MFEA.EvBestFitness;%读取目标值
    y1 = data(:,1);
    y2 = data(:,2);
    dim=length(var);
    ny1=y1(1:dim);
    ny2=y2(1:dim);
    p1=polyfit(var,ny1,8);%对数据进行二项式拟合
    p2=polyfit(var,ny2,8);
    p=0.4*p1+0.6*p2;%对拟合的结果进行加权平均
    for i=1:dim
        temp = polyval(p,var(i));%利用拟合的图形计算数据结果
        obj = obj + temp;
    end
end
```

Figure 1: New Function

5 Results

In order to compare how much the function I designed improves the original MFEA algorithm, I chose two function tasks as reference, theirs are the Rastrigin function and the Ackley function and Sphere Function. Their expressions are shown as follows.

1) Sphere function,

$\sum_{i=1}^{D} x_i^2$.

2) Ackley's function [27], [28],

$20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}(2\pi z_i)\right); \boldsymbol{z} = \mathbf{M}_A \times (\boldsymbol{x} - \boldsymbol{o}_A)$

3) Rastrigin's function [27], [28],

$\sum_{i=1}^{D}\left(z_i^2 - 10\cos(2\pi z_i) + 10\right); \boldsymbol{z} = \mathbf{M}_R \times (\boldsymbol{x} - \boldsymbol{O}_R)$ And based on the above idea we conducted the following experiment, we set up several control groups, which are (Rastrigin,Ackley),(Rastrigin,Ackley,Sphere),((Rastri the result are as follows.

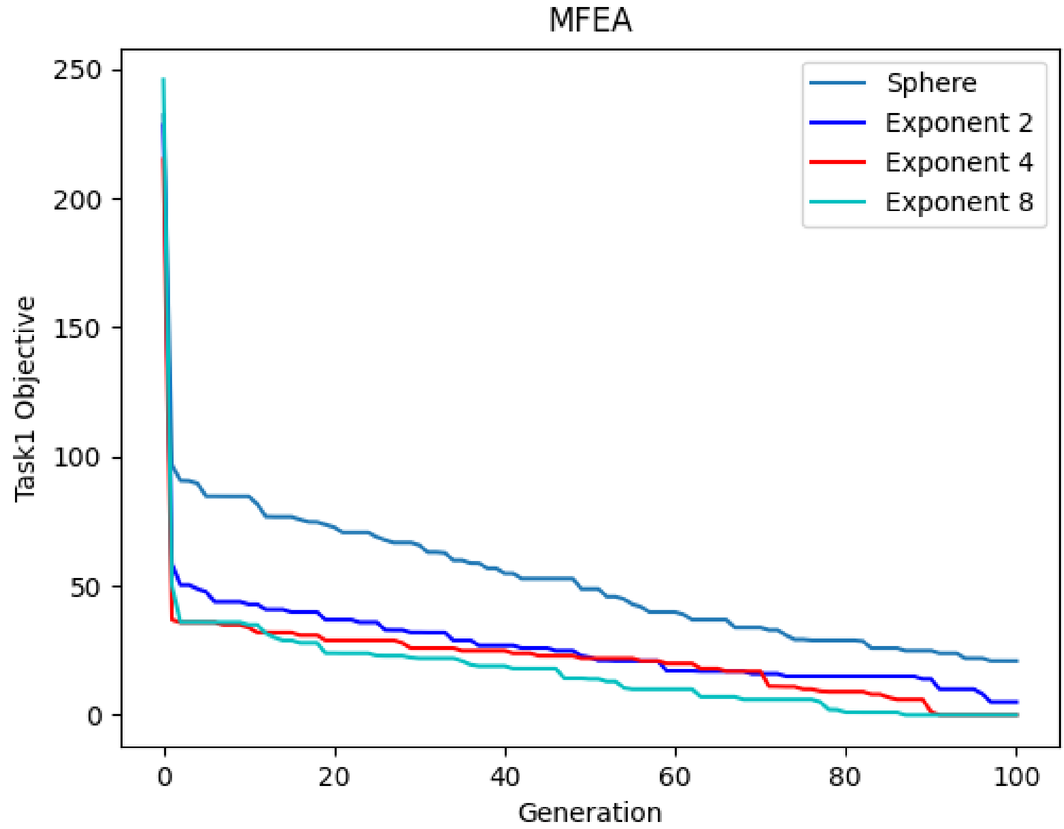

Figure 2: Rastrigin results

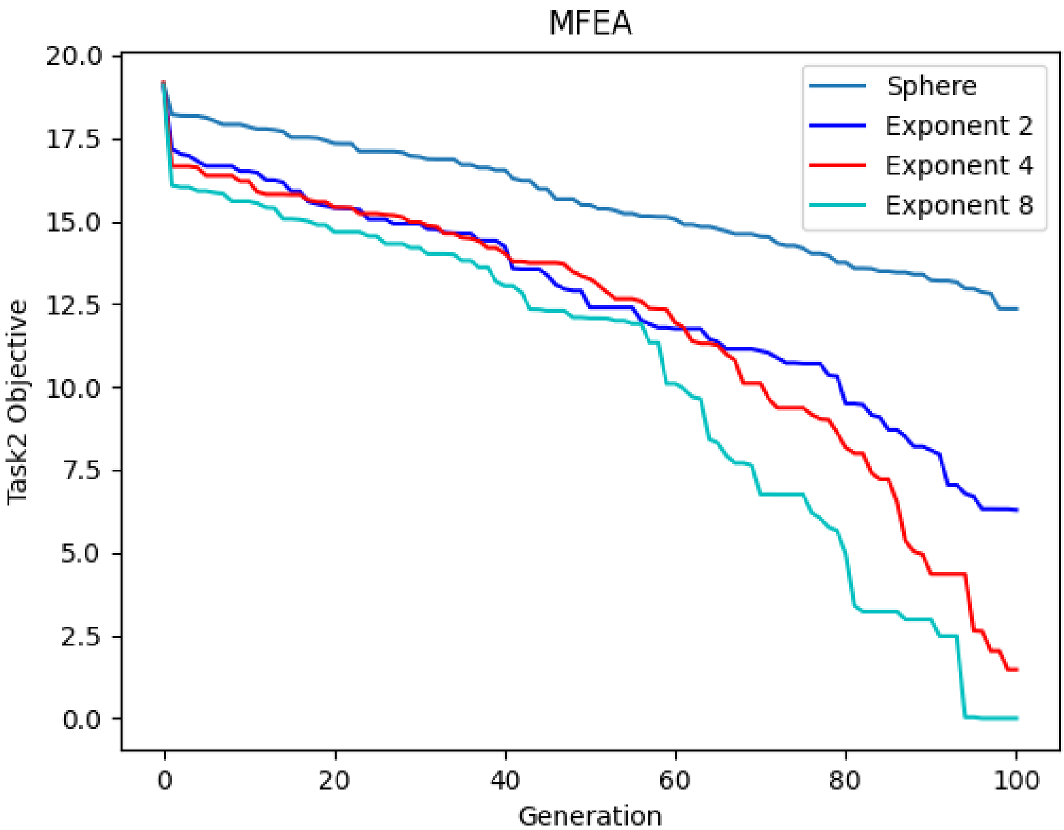

Figure 3: Ackley results

From above result,we can see that with the input of our function, MFEA can find more better minimums which are close to the zero.It has proved the effectiveness of our function.

6 Conclusion and future work

MFEA is a corresponding optimization algorithm proposed for the novel MTO problem, whose idea is similar to Darwinian proximity theory, pursuing the simultaneous evolution of multiple tasks in pursuit of their internal implicit public information to reach the optimal solution better and faster. Our new function is designed to better provide this useful information to facilitate the evolution of the task. Our next step is to introduce gradient into the evolutionary process of MFEA by combining the two to some extent and innovating it. There are two advantages of doing so, one is that the gradient information can provide more effective information to the evolution and promote the evolution of the task, and the other is that the gradient represents the direction of the fastest descent at that point, so the direction of evolution should be towards the direction of the closest solution, avoiding unnecessary negative knowledge migration. However, at the same time, gradient descent also has the inevitable problem of falling into the local optimal solution, which is a problem that needs to be addressed in future work.

References

[1] BACK T, HAMMEL U, SCHWEFEL H P. Evolutionary computation: comments on the history and current state[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 3-17. DOI: 10.1109/4235.585888.

[2] GUO S M, YANG C C. Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator[J]. IEEE Transactions on Evolutionary Computation, 2015, 19: 31-49.

[3] MA X, ZHANG Q, TIAN G, et al. On Tchebycheff Decomposition Approaches for Multiobjective Evolutionary Optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(2): 226-244. DOI: 10.1109/TEVC.2017.2704118.

[4] GUPTA A. Evolutionary Multitasking for Single-objective Continuous Optimization: Benchmark Problems, Performance Metric, and Baseline Results[J]. Nanyang Technological University, Technical Report, 2017.

[5] IQBAL M, BROWNE W N, ZHANG M. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems[J]. IEEE Transactions on Evolutionary Computation, 2013, 18(4): 465-480.

[6] MILLS R, JANSEN T, WATSON R A. Transforming evolutionary search into higher-level evolutionary search by capturing problem structure[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(5): 628-642.

[7] CHEN X, ONG Y S, LIM M H, et al. A Multi-Facet Survey on Memetic Computation[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(5): 591-607. DOI: 10.1109/TEVC.2011.2132725.

[8] ONG Y S, LIM M H, CHEN X. Research frontier-memetic computation—past, present & future[J]. IEEE Computational Intelligence Magazine, 2010, 5(2): 24.

[9] RICE J, CLONINGER C, REICH T. Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models.[J]. American journal of human genetics, 1978, 30(6): 618.

[10] CLONINGER C R, RICE J, REICH T. Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance.[J]. American journal of human genetics, 1979, 31(2): 176.

[11] CAVALLI-SFORZA L L, FELDMAN M W. Cultural versus biological inheritance: phenotypic transmission from parents to children.(A theory of the effect of parental phenotypes on children's phenotypes).[J]. American journal of human genetics, 1973, 25(6): 618.

[12] FELDMAN M W, LALAND K N. Gene-culture coevolutionary theory[J]. Trends in ecology & evolution, 1996, 11(11): 453-457.