

An Improved Estimation of Distribution Algorithm in Dynamic Environments

Xiaoxiong Liu
College of Automation
Northwestern Polytechnical
University, Xi'an 710072, China
nwpulxx@163.com

Yan Wu
School of Science Xidian
University Xi'an 710071, China
yanerch@163.com

Jimin Ye
School of Science Xidian
University Xi'an 710071, China
jmye@mail.xidian.edu.cn

Abstract

In dynamic environments, the optimal solution changes over time. To track the solution, an improved univariate marginal distribution algorithm(UMDA) is proposed. A transfer model is introduced to increase the diversity of population. The current information is used to avoid being trapped into the local optimization for dynamic optimization problems. The scheme is illustrated through simulations applying dynamic moving peaks benchmark. The results show that the proposed algorithm is effective and can accommodate the dynamic environments rapidly.

1. Introduction

In real-world, many optimization problems are dynamic. In such cases the optimization algorithm has to track a moving optimum as closely as possible, rather than just find a single good solution. Over the last decades, there has been an increasing interest to solve the dynamic optimization using evolutionary algorithm [1-10]. Evolutionary algorithms are search and optimization algorithms based on principles of biological evolution such as natural selection or survival of the fittest, which have been applied in many areas. In the literature of evolutionary algorithms, Estimation of Distribution Algorithms (EDAs)[6,8], also called Probabilistic Model Building Genetic Algorithms, have attracted the attention of many researchers. Compared with traditional evolutionary algorithms, EDAs are also population-based algorithms, but there are no crossover and mutation operators. In EDAs probabilistic models are built to describe the distribution of solutions, the offspring are generated by sampling from these models. As a result EDAs can avoid the disruption of building blocks known from crossover and mutation operators and overcome the deceptive problem. So EDAs are improvement of standard evolutionary algorithms, for example, Univariate marginal distribution algorithm (UMDA) and Bayesian Optimization Algorithms (BOA), etc.

In this paper a new univariate marginal distribution algorithm (UMDA) is proposed for dynamic optimization problems. In UMDA, all the variables are assumed to be independent in order to make the problem simple, and

hence, only need to estimate the marginal probability of each variable in the selected solutions at each generation. Thus the computation of UMDA is relatively small. In a completely novel approach, A transfer model is set up to increase the diversity of population and avoid converge local optimum. The proposed algorithm is combined the UMDA with the transfer model to track a moving optimum solution quickly. A simulation study on dynamic moving peaks benchmark was introduced to compare the performance of several UMDAs. The results show that the proposed algorithm can adapt the dynamic environments rapidly.

2. Univariate marginal distribution algorithm

The continue version of univariate marginal distribution algorithm (UMDA) is one class of Estimation of Distribution Algorithms and has been applied to many optimization problems [8].

The algorithm works as follows. At each step t , an n -dimensional random variable $X^t = (X_1^t, X_2^t, \dots, X_n^t)$ is maintained. It is usual to assume that the joint probability distribution of X^t follows an n -dimensional normal distribution which is factorized by a product of n unidimensional and independent normal densities. This assumption will be made here. Therefore each component of X^t is distributed as a unidimensional normal, that is $X_i^t \sim N(\mu_i^t, \sigma_i^t)$, where

$$f(X_i^t = x_i^t) = \frac{1}{\sqrt{2\sigma_i^t}} \exp\left\{-\frac{(x_i^t - \mu_i^t)^2}{2(\sigma_i^t)^2}\right\} \quad (1)$$

With $i = 1, \dots, n$. In other words $f(X_i^t = x_i^t)$ denotes the density function of a unidimensional normal with mean μ_i^t and standard deviation σ_i^t in point x_i^t .

Drawing the above n -dimensional random variable, N individuals are obtained to form the population $Pop(t+1)$. Then the μ best individuals are selected to form an interim population $S(t)$ which is used to obtain the means and standard deviations of the random variable X^{t+1} . These parameters are estimated by using their corresponding maximum likelihood estimators. In this

This work is supported by the National Natural Science Foundation of China (No. 60775013)

way the new unidimensional distributions at step $t+1$ are achieved.

As the search progresses, the elements in σ_i^{t+1} move away from their initial settings towards 0. The population loses the diversity when UMDA converges, so UMDA needs to be adapted for optimal results on dynamic optimization problems.

3.Improved UMDA in dynamic environments

In dynamic environments, the fundamental idea of the tracking optimum solution approach is to increase the diversity of the population when the environment changes, with the aim to enlarge the search space and find the promising region. However it is to take long time for tracking the optimal solution by diverging randomly. So a new approach to increase the diversity by guide fashion is proposed in this paper.

The proposed method is intended for dynamic problems of a discrete nature, where the objective changes in a non-infinitesimal way and the optimization engine has a fixed number of function evaluations available at each time step.

In the static environments, the optimal solution $X^*(t)$ of the current population is usually acted as an attractor, which attracts the other individuals move towards the attractor to find the promising region of the optimum. However in the dynamic environments, when the problem changes at generation t , the next optimal solution move away from $X^*(t)$. Then the information of optimal solution $X^*(t)$ may no longer be true and may actually be misguiding the search as attractor. But it is hoped that the information of $X^*(t)$ also can be used to guide the search. The use of the information is discussed in the following.

Assume that the problem changes at generation t , then the information of $X^*(t)$ may no longer be true, but it provides contrary information to us, it will not be the next optimal solution and it can be a trap. In order to find the new optimal solution, the algorithm needs increase the diversity and search new space which is different from the region presented by current population. That is to say we should search space away from the current population. According to the discussion above, we consider the $X^*(t)$ as a repeller. Then some selected individuals are diffused by the repeller in order to increase the diversity and find new optimal solution. It is called transfer model. The detail description is as following.

Assume that k is the environment change period index, if the environment changes at generation t (corresponding to period k), the optimal solution of the current

population is denoted by $B(k)$ ($B(k) = X^*(t)$, $X^*(t)$ is the optimal solution of generation t). Let U be the set of all selected individuals for transfer. For every individual $X \in U$, it is applied with the following formula to generate new individual X_{new} .

$$X_{new} = X + \lambda_1 (X - B(k)) \quad (2)$$

Where λ_1 is random number from $[0, 1]$. According to the transfer model, if the environment changes at generation t , then $X^*(t)$ can be considered as a repeller rather than an attractor. Then the new individuals are generated by transfer model to increase the diversity and search new space.

By transfer model, note that when the optimal solution moved far away from the current population, the step size λ_1 is too small to track. In order to enhance the track, formula (2) is used repeatedly until the next change. Thus, the transfer model makes the individuals diffuse from repeller by wave formulation continuously.

Based on the considerations above, the Improved UMDA is described in following.

Step1. $t \leftarrow 0$, obtain randomly the parameters of a normal probability distribution for each variable.

Step2. Sample X^t to obtain a population $Pop(t)$ of N individuals;

Step3. Test for change. If the environment changes, then memories $B(k) = X^*(t)$ and turn to step 4, otherwise step 6;

Step4. Generate μ neighbor individuals of $X^*(t)$ to form the neighbor set U , where the neighbor distance is controlled in $[0.05, 0.15]$

Step5. Apply formula (2) on individuals of U to form one set of D , then turn to step 7;

Step6. Apply the transfer model on individuals of U and get one set of D ;

Step7. Select μ best individuals from $Pop(t) \cup D$ to form an interim population $S(t)$, $U = S(t)$;

Step8. Estimate the parameters of the new density functions:

$$\mu_i^{t+1} = \frac{\sum_{j=1}^N x_i^{j,t}}{N} \quad i = 1, \dots, n \quad (3)$$

$$\sigma_i^{t+1} = \sqrt{\frac{\sum_{j=1}^N (x_i^{j,t} - \mu_i^{t+1})^2}{N}} \quad i = 1, \dots, n \quad (4)$$

Step9. Mutate σ_i^{t+1} according to mutation probability;

Step10. If the termination is not satisfied, $t \leftarrow t + 1$, go to step 2, otherwise, stop.

According above algorithm, the improved univariate marginal distribution algorithm is achieved by using transfer model technique.

4. Simulation results

In order to test the performance of the proposed algorithm, we use the publicly available moving peaks benchmark(MPB)[4-5]. The function is formulated as follows:

$$F(x, t) = \max_{i=1, \dots, 10} \frac{H_i(t)}{1 + W_i(t) \sum_{j=1}^5 (x_j - X_j(t))^2} \quad (5)$$

where $H_i(t) = H_i(t-1) + k_H \sigma$, $W_i(t) = W_i(t-1) + k_W \sigma$, $\sigma \in N(0, 1)$, $X(t) = X(t-1) + \omega_i(t)$,

$$\omega_i(t) = \frac{s}{|r + \omega_i(t-1)|} ((1 - \lambda)r + \lambda \omega_i(t-1)).$$

The moving peak function consists of a number of peaks $X(t)$, of varying heights $H_i(t)$ and widths $W_i(t)$, moving by a fixed shift length s in random directions. The peaks change position every τ generation by a distance of $s=1$ in a random direction. These parameter settings are summarized in Table 1.

Table 1 Standard settings for the moving peaks benchmark

Parameter	Setting
Number of peaks p	10
Number of dimensions d	5
Peak heights	[30,70]
Peak widths	[1,12]
Change severity s	1
Correlation coefficient λ	0

Experiments were carried out to compare the performance of the proposed algorithm, IUMDA and MUMDA on the test environments constructed above. IUMDA is UMDA with random immigrants[9], and MUMDA is hyper mutation based UMDA[10], in which hyper mutation is selected to increase the diversity when the problem is changed.

For each experiment of combining different algorithm and test problems, 50 independent runs were executed with the same set of random seeds. For each run of different algorithm on each problem, the best of generation fitness was recorded every generation. And for

each run of an algorithm on a dynamic problem, 500 generations are allowed.

In order to compare performance of different algorithms, the total population size N was fixed at 40 individuals. The parameter of UMDA $\alpha = 0.5$ was used for all experiments. And the average errors which are the averages of the error of the best solution on each generation over all run are located in figures. The simulation results on different change severities s are shown in Fig.1-3 (the solid line represent the improved UMDA, the dotted line represent the IUMDA, and the dash-dot line represent the MUMDA).

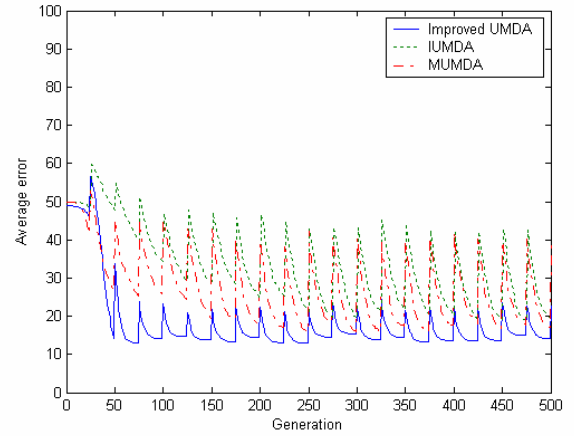


Fig. 1 Experimental results on MPB
($p = 10, \tau = 25, s = 1$)

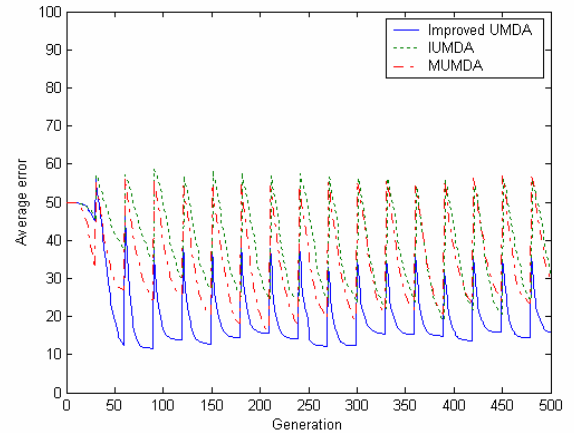


Fig.2 Experimental results on MPB
($p = 10, \tau = 30, s = 2$)

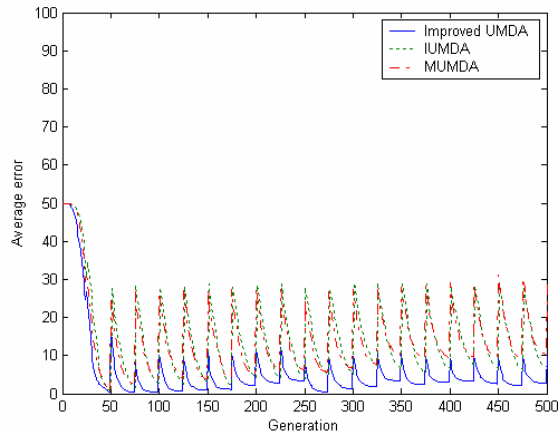


Fig. 3 Experimental results on MPB
($p = 1, \tau = 25, s = 1$)

From the simulation results, the following considerations can be drawn.

(1) First, generally, the improved UMDA outperforms IUMDA and MUMDA in the same environment. One straight forward approach to make EAs more suitable for dynamic environments is to increase the diversity of the population after a change.

(2) The proposed algorithm increases the diversity by generating the new individuals with combining transfer model. Thus the proposed algorithm makes useful of the current and history information of the optimal solutions to increase the diversity by guide fashion. The new individuals are close to or in the promising region. So the time to track the optimum for the proposed algorithm is shorter than other algorithm.

(3) By using improved UMDA, the optimum solution is achieved in dynamic environments. Dynamic optimum problem field is extended by using EDAs.

5. Conclusion

In dynamic environments, it is important that the optimization algorithm is able to continuously track the moving optimum solution over time. In this paper, a new approach is proposed to adapt dynamic environments. A transfer model is set up to increase the diversity in a guide fashion when environment changes. Combined the transfer model, an improved univariate marginal distribution algorithm is introduced to deal with dynamic optimization problems. The simulation results show that the proposed algorithm can adapt the dynamic environments rapidly to compare the performance of several UMDAs.

In this paper, because of considering a new strategy for dynamic optimal problems, the proposed method is verified by only using moving peak optimum solution, and is good than other UMDA faintly. To get better

optimum results in dynamic environments, the algorithm is to improve in future yet.

Reference

- [1] Yaochu J, Branke J, Evolutionary optimization in uncertain environments-a survey, *IEEE transactions on evolutionary computations*, 2005, 9(3):1-15
- [2] Wang Hong-feng, Wang Ding-wei, Yang Sheng-xiang, Evolutionary algorithms in dynamic environments, *Control and Decision*, 2007, 22(2):127-131
- [3] Liu Chun-an, Wang Yu-ping, Evolutionary algorithm for dynamic multi-objective optimization problems and its convergences, *Acta electronica sinica*, 2007, 35(6):1118-1121
- [4] J. Branke. The moving peaks benchmark website. Available: <http://www.aifb.uni-karlsruhe.de/~jbr/movpeaks>
- [5] Tim Blackwell and Jürgen Branke, Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments. *IEEE transactions on evolutionary computations*, 2006 10(4): 459-473
- [6] Shengxiang Yang, Memory-Enhanced Univariate Marginal Distribution Algorithms for Dynamic Optimization Problems, *IEEE Congress on Evolutionary Computation*. 2005, Volume: 3: 2560- 2567
- [7] Iason Hatzakis, David Wallace, Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Forward-Looking Approach, *Genetic And Evolutionary Computation Conference, Proceedings of the 8th annual conference*, 2006, pp:1201 - 1208, Seattle, Washington, USA
- [8] P. Larranaga R. Etxeberria J. A. Lozano, and J. M. Pena. Optimization by Learning and Simulation of Bayesian and Gaussian Networks. Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999, pp:2254-2265
- [9] Cobb H G. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environment. Washington, Naval Research Laboratory, 1990, pp:224-235
- [10] Grefenstette John, Genetic algorithms for changing environments. *Proceeding of the Parallel Problem Solving from Nature*, Brussels, 1992: 137-144