Estimation of distribution algorithm with scatter search for dynamic optimisation problems

Fahong Yu*

College of Mathematics and Information Engineering,
Jiaxing University,
Zhejiang, China
and
Ningbo Institute of Technology,
Zhejiang University,
Zhejiang, China
Email: fhyu520@whu.edu.cn
*Corresponding author

Feng He

College of Mathematics and Information Engineering,
Jiaxing University,
Zhejiang, China
Email: fenghe9930@126.com

Meijia Chen

Center of Economic Managing Experiment,
Jiaxing University,
Zhejiang, China
Email: mjchen1980@126.com

Longhua Ma

Ningbo Institute of Technology,
Zhejiang University,
Zhejiang, China
Email: lhma@iipc.zju.edu.cn

Zheming Lu

School of Aeronautics and Astronautics,
Zhejiang University,
Zhejiang, China
Email: zheminglu@zju.edu.cn

Abstract: Aiming at the trouble to track the optima in dynamic environments with estimation of distribution algorithms (EDAs). An estimation of distribution algorithm with scatter search (EDASS) is proposed in this paper. Its basic idea is to employ a scatter search to increase the diversity in a guided fashion and an adaptive leader clustering method to locate multiple local optima. Both the information of current population and the part history information were referred for building probability model. The experimental results show that the EDASS is effective for dynamic optimisation problems.

Keywords: estimation of distribution algorithms; EDAs; dynamic optimisation problems; DOPs; clustering; scatter search.

Reference to this paper should be made as follows: Yu, F., He, F., Chen, M., Ma, L. and Lu, Z. (2015) 'Estimation of distribution algorithm with scatter search for dynamic optimisation problems', *Int. J. Computing Science and Mathematics*, Vol. 6, No. 3, pp.221–231.

Biographical notes: Fahong Yu received his PhD in Computer Science from Whuhan University in 2012. He is currently an Assistant Professor at Jiaxing University. His current research interests include nonlinear time series and complex networks.

Feng He is a Professor of Computer Science at the University of Jiaxing. His current research focuses on intelligent computing.

Meijia Chen is an Assistant Professor at the University of Jiaxing. Her research focuses on intelligent computing.

Longhua Ma is a Professor of Information Engineering at the Ningbo Institute of Technology. His current research focuses on intelligent computing.

Zheming Lu is a Professor of Computer Science at the University of Zhejiang. His current research focuses on complex networks.

1 Introduction

The premature convergence issue becomes even more important and complex when dealing with dynamic optimisation problems (DOPs). When changes occur, solutions already found may be no longer valuable and the process must engage in a new search effort (Branke, 2002). Traditional evolutionary algorithms (EAs), for instance, may encounter difficulties while solving dynamic problems if the first convergence stage reduces population diversity, thus decreasing its capability to react to sudden changes.

For the dynamic multimodal problems (DMOPs), how to find the multi-optima solves is important. Many researchers have considered multi-populations as a means of enhancing the diversity of EAs to address DOPs. A PSO algorithm that uses a k-means clustering algorithm to identify the centres of different clusters of particles in the population and then uses these cluster centres to substitute the personal best or neighbourhood best positions (Li and Yang, 2009). The limitation of this clustering approach lies in that the number of clusters must be predefined (Parrott and Li, 2006). In

SOS, the population is composed of a parent population that searches through the entire search space and child populations that track local optima.

Estimation of distribution algorithms (EDAs) (Fan et al., 2013) has exhibited good performance for optimisation problems. EDAs employ probabilistic models to describe the promising area in the solution space and use these models to guide the generation of the candidate solutions for the next generation. When EDAs is applied to optimisation problems in dynamic environments, the challenge (Goncalves, 2011) is that the objective is not only to locate an optimum, but also to track that moving optimum as closely as possible. In this paper, improved EDAs is proposed to implement them and it constitutes a new efficient way to deal with DMOPs.

The motivation of the improved algorithm is to improve the performance for locating multiple local peaks by incremental clustering and multi-models sampling. Meanwhile, a policy of diffusion search was introduced to enhancing the diversity of the population in a guided fashion when the environment has changed. The policy uses both the information of current population and the part history information of the optimal solutions availability.

2 UMDA

UMDA is a probability model based EA in which a probability distribution model of promising solutions is built to guide the search. In UMDA, $X^t = (X_1^t, X_2^t, \ldots, X_n^t)$ denotes an n-dimensional random variable of generation t. The parameters can be calculated by the formula (1).

$$\mu_i^{t+1} = \frac{\sum_{j=1}^{M} x_i^{j,t}}{M} \quad \sigma_i^{t+1} = \sqrt{\frac{\sum_{j=1}^{M} \left(x_i^{j,t} - \mu_i^{t+1}\right)^2}{M}} \tag{1}$$

where $x_i^{j,t}$ denotes the value of gene i of individual j of the interim population P_t, i = 1,2,…,n, j = 1,2,…,M. The framework of the basic UMDA is shown in algorithm 1.

Algorithm 1 UMDA

1: $t \leftarrow 0$, obtain randomly the parameters of a normal probability distribution for each variable;

2: Sample X^t to obtain a population P_t of N individuals;

3: Evaluate the fitness of each individual;

4: Select M best individuals from $P_t \cup$ elitists;

5: Estimate the parameters by the formulas (1);

6: Mutate σ_i^{t+1} according to mutation probability p_m;

7: If the termination is not satisfied, $t \leftarrow t+1$, go to step 2.

3 EDA with scatter search

3.1 Framework of the EDASS

For DMOPs, it is pointed out in Section 1 that the algorithms are required to find multiple optima and track the optima founded. So in this paper we apply the clustering approach to find multiple peaks. In order to allow each cluster to track its peak, the new Scatter search is proposed to sustain diversity within a population. The framework of EDASS for DMOPs is given in algorithm 2.

Algorithm 2 EDASS Algorithm

```
1:  Define arrays farr, iarr and scarr to record fitness, corresponding individual and the
    number of previous cluster where the individual generated respectively;
2:  Define an array prearr to record the probability model every cluster in previous
    generation;
3:  Define arrays carr and cfarr to record the centroid and its fitness of every cluster in
    current generation;
4:  Set the fitness evaluation counter evals = 0;
5:  Generate an initial population P0 randomly;
6:  while stop criteria is not satisfied {
7:      flag =DetectChange(carr, evals);
8:      The fitness of each candidate solution is evaluated;
9:      The best M individuals are selected by truncation selection;
10:     cnum = Clustering(Pt, carr);
11:     for (i=1; i<=cnum; i++) DiffSample(carr[i], flag, t);
12: }
```

3.2 Adaptive leader clustering

RLA, a cluster algorithm, need not explicitly specify a priori knowledge about the number of clusters. Instead, the RLA considerably reduces the degree of difficulty by transforming a hard problem into a simpler one. Thus, it offers EDA practitioners a high degree of freedom in deciding the number of clusters. However, the RLA does not reconcile the leader with the current set of members. The performance is, more often than not, unstable. Therefore, an improved adaptive leader clustering based on fitness-aided ordering scheme (ALCF) was presented as algorithm 3.

Algorithm 3 *Clustering(P, carr, cfarr)*

```
1:  Set the initial value of cflag[M] is zero respectively to denote that a individual is not
    belong to any cluster;
2:  SortOnFit(farr, iarr, scarr, M);
3:  for ((i=0; i<M; i++))
4:      for (j=i+; j<= M; j++) {
5:          dmatrix[i][j]=d(i, j);
6:          if (maxdist < d(i, j)) maxdist = d(i, j);
```

```
7:          if (d(i, j)==0) cflag[j] = –1;
8:    }
9:    Set the initial value of rarr[M] is r respectively to denote the radius of every individual
      when they act as the centroid of a cluster. The value of variable r =ξ*maxdist;
10:   carr[0]=1;     //record the quantity of clusters at present;
11:   carr[1]=0;          //record the centroid of every cluster
12:   cflag[0]=1; //record which cluster the individual belong to
13:   for (i=1; i<M; i++) {
14:       if (cflag[i]==0)
15:           for (j=1; j<= carr[0]; j++) {
16:               if ((farr[i] == farr[carr[j]])&&( dmatrix[i][carr[j]] <= rarr[carr[j]]))
17:                   rarr[carr[j]]= rarr[carr[j]]/2;
18:               else if (d(i, carr[j])<= rarr[carr[j]])
19:                   cflag[i] = j;
20:           }
21:       if (cflag[i]==0) {
22:           carr[0]++;
23:           carr[carr[0]]=i;
24:           cflag[i] = carr[0];
25:       }
26:   }
27:   return carr[0];
```

Note that the threshold $r = \xi * maxdist$. The adoptive radius r directly determines how many clusters can be obtained by ALCF.

3.3 Scatter search

In the dynamic environments, to find the new best solution and enhance the speed of evolution to the best solution is necessary (Ruan and Zhang, 2013). When the problem changes at the end of time step t, the next optimal solution may move away from $x^*(t)$, $x^*(t)$ denotes the optimal solution during time step t. In order to find the new optimal solution and track the moving peak, a scatter search is introduced to EDAs to improve the individuals' quality efficiently in local area in this paper. The objective of scatter search is the local best solution $x^*(t)$ that is considered as a repulsive core.

To realise the objective of diffusing gradually from the repulsive core by guide fashion after the problem changed, a neighbour set of the repulsive core is generated firstly. Let $v(t) = x^*(t) - x^*(t-1)$, then $v(t)$ is called the velocity of optimal solution at time t is used as the inertia velocity at time $t+1$ to predict the next optimal solution. Then the more promising solutions can be found by a simple inertia prediction on the neighbour set of the repulsive core is combined as formula (2).

$$x_{new} = x + \lambda_1 v(t) \tag{2}$$

Where x is the individual selected from the current cluster according overcrowding select strategy as Algorithm 6, λ_1 is random number from the field [0, 1], which denotes the step size to move in direction of $v(t)$. From (2), if the prediction is correct, the prediction set aids to discover the promising region of optimal solution quickly. Moreover, if the prediction is not correct, the individuals newly generated may increase the diversity. The frame work of the scatter search process is described in algorithm 4.

Algorithm 4 *DiffSample(carr, flag, t)*

```
regen=0;
num=0;
for (i=1; i<= carr[0]; i++) {
    if ((t!= 0)&&(scar[carr[i]]!=0)){
        if v(t) is not a zero vector {
            r = rarr[carr[i]];
            num=Selectcrowd(r, S, P');
            if (num==1) regen=1;
            The new individuals is generated according to (2) and combined into the cluster C_i;
        }else regen=1;
    }
    Estimate the distributed parameters by the formulas (1);
    Sampling(i, m, μ, σ);
    if (regen==1) num new individuals are generated according to (3);
}
```

Note that if the velocity $v(t)$ equals to zero that mean that the current cluster may be met convergence, EDASS have to explore new region in search space. So the formula (3) is adopted to generate some new individuals.

$$x_{new} = \lambda_2 x + (1 - \lambda_2) x' \tag{3}$$

Where x' is a value that uniformly sample in the search space, λ_2 is random number from the field [0, 1], which denotes the step size to move in direction of x.

The process of selecting some individuals according to overcrowding degree of individuals in current cluster is presented as the algorithm 5.

Algorithm 5 *Selectcrowd(r, S, P')*

```
S = φ;
P' = C_i;
num=0;
while (sizeof(P')!=0){
Get the best fitness individual X ∈ P;'
num = num + 1;
S:=S ∪ { X};
remove X from P';
```

9: for each $Y \in P$

10: if ($(d(X, Y)<r_s$) remove Y from P';

11: *}*

12: return *num;*

It is easy to understand that this algorithm aims to pick out multiple high-quality individuals in different regions of the current cluster space. The parameter r_s can control the crowded degree of S and ensure Scatter search exploit different peaks in the search space.

4 Experimental study

4.1 Test functions and parameter settings

In order to test the performance of the proposed algorithm, the moving peaks benchmark (MPB) is used. The MPB problem (Branke, 2002) has been widely used as dynamic benchmark problems in the literature.

$$f\left(\vec{x}, t\right) = \max_{i=1,\ldots,p} \frac{H_i(t)}{1 + W_i(t)\sum_j^D \left(x_j(t) - X_{ij}(t)\right)^2} \tag{4}$$

Where $H_i(t)$ and $W_i(t)$ are the height and width of peak i at t, respectively and X_{ij}(t) is the first j element of the location of peak i at time t. The p independently specified peaks are blended together by the 'max' function. More formally, a change of a single peak can be described as follows:

$$H_i(t) = H_i(t-1) + height\_severity * \sigma \tag{5}$$

$$W_i(t) = W_i(t-1) + width\_severity * \sigma \tag{6}$$

$$X_i(t) = x_i(t-1) + \vec{v_i}(t) \tag{7}$$

$$\vec{v_i}(t) = \frac{s}{|\vec{r} + \vec{v_i}(t-1)|}\left((1-\lambda)\vec{r} + \lambda\vec{v_i}(t-1)\right) \tag{8}$$

Table 1 Default settings for the MPB

Parameter	*Value*
Number of peaks, p	10
Number of dimensions, D	5
Peak heights	[30.0, 70.0]
Peak widths	[1, 12]
Shift length s	1.0
Correlation coefficient λ	0
Height severity	7.0
Width severity	1.0

The definition for the landscape of MPB used of this paper is formulated as Table 1.

The performance measure used is the offline error, which is defined as follows:

$$\eta = \frac{1}{K}\sum_{k=1}^{k}(h_k - f_k) \tag{9}$$

For each run, there were $K = 100$ environments, which result in $K \times U = 5 \times 10^5$ fitness evaluations. All the results reported are based on the average over 50 independent runs with different random seeds.

In these experiments, some of them are used to investigate the work mechanism and the sensitivity of EDASS to these parameter settings. Other experimental results of EDASS are also compared with that of MDUMDA (Wu and Wang, 2008) and MQSO (Li and Yang, 2009) to testify the efficiency of EDASS.

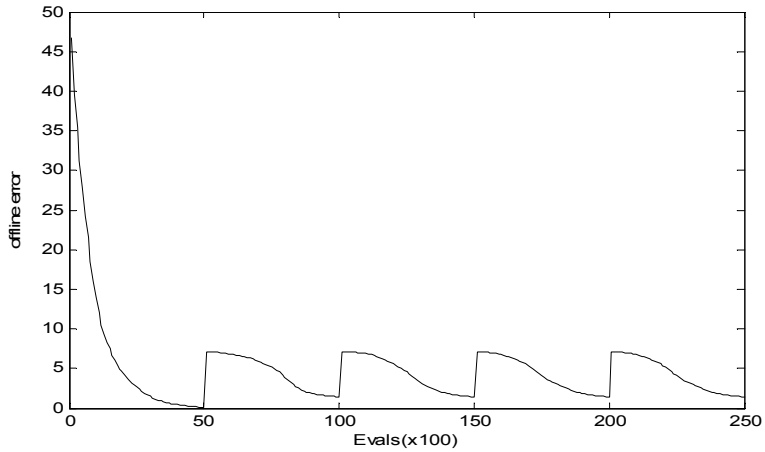
Unless stated otherwise, the parameter settings in those compared algorithms are set in the following. For EDASS, the population size is 100, the truncation selection threshold $\alpha = 0.5$. The radius coefficient $\xi = 1/40$ and the overcrowding coefficient $\zeta = 1/3$. For MQSO, the population size is 100, the corresponding parameters of PSO are $C1 = C2 = 1.796$, $w = 0.729$, exclusion radius r_{excl} is equal to convergence $r_{conv} = 30$, quantum particle movement radius $R_{cloud} = 1$, the configuration is $M(N + N^q)$ (M is the total number of swarms, N and N^q are the numbers of neutral and quantum particles in each swarm). The population size N is set to be 200, the truncation selection threshold $\alpha = 0.5$. The distance d is set to be [0, 0.35] for MDUMDA.

4.2 Experimental investigation

4.2.1 Testing the work mechanism of EDASS.

The average dynamic behaviour of the offline error against the solving process for five environmental changes based on the default settings of the MPB problem over 50 runs is showed in Figure 1. It is can be seen that the offline error reaches almost zero for the initial environment, that is because the ten peaks have the same initial height in the fitness landscape, which enable the algorithm to easily find one or some of the peaks.

Figure 1 Offline error for five environmental changes



4.2.2 Effect of varying the configurations

The aim of this set of experiments is to examine the effect of the different configurations on the performance of EDASS. EDASS was run 50 times with different combined values of radius coefficient ξ and the overcrowding coefficient ζ. The offline error is shown in Table 2.

Table 2 Offline error ± standard error for different configurations

	ζ=1	*ζ=1/2*	*ζ=1/3*	*ζ=1/5*	*ζ=1/10*
ξ = 1/4	4.71 ± 0.15	4.21 ± 0.46	4.17 ± 0.09	4.74 ± 0.68	5.42 ± 0.31
ξ = 1/10	3.83 ± 0.77	3.65 ± 0.27	3.46 ± 0.27	3.98 ± 0.06	4.83 ± 1.03
ξ = 1/20	2.62 ± 0.59	2.33 ± 0.18	2.22 ± 0.04	2.67 ± 0.28	3.22 ± 0.35
ξ = 1/30	2.12 ± 0.04	1.58 ± 0.18	1.37 ± 0.05	2.26 ± 0.57	2.37 ± 0.17
ξ = 1/40	1.46 ± 0.29	1.26 ± 0.11	1.12 ± 0.16	1.45 ± 0.09	1.76 ± 0.26
ξ = 1/50	1.67 ± 0.46	1.19 ± 0.08	1.23 ± 0.39	1.76 ± 0.52	2.25 ± 0.37
ξ = 1/100	3.15 ± 0.31	4.08 ± 0.62	4.37 ± 0.75	5.09 ± 0.08	5.18 ± 0.42

From Table 2, it can be seen that the value of radius coefficient directly determines the number of clusters that are generated by the clustering method. This is why the performance of EDASS greatly depends on the value of radius coefficient. Too large or too small radius coefficient will cause too few or too many clusters created, which maybe far away from the real number of peaks in the search space. When the radius coefficient is fixed a specific value, setting the overcrowding coefficient to a too large or too small value will affect the performance of EDASS, vice versa. The optimal configuration of EDASS for the default settings of the MPB problem with ten peaks is ξ = 1/40 and ζ = 1/3, which enables EDASS to achieve the smallest offline error of 1.12.

Table 3 Offline error under different number of peaks

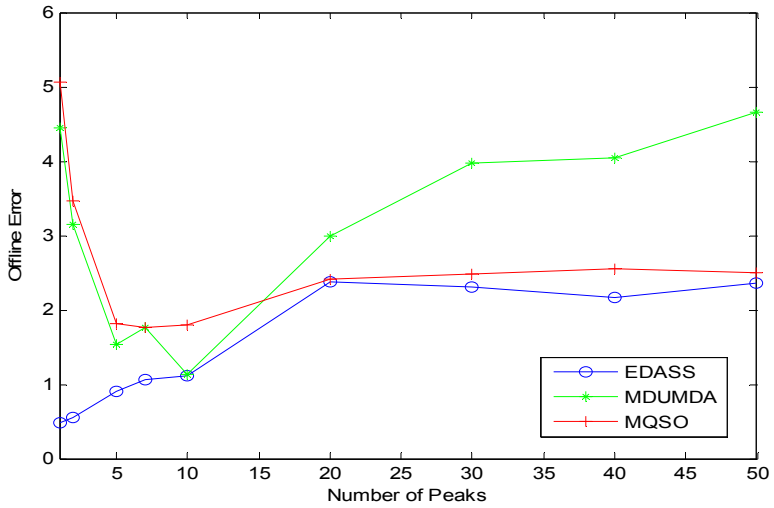
	EDASS	*MDUMDA*	*MQSO*
p = 1	0.49 ± 0.54	4.45 ± 0.69	5.07 ± 0.17
p = 2	0.56 ± 0.31	3.15 ± 0.91	3.47 ± 0.22
p = 5	0.90 ± 0.26	1.53 ± 0.47	1.81 ± 0.04
p = 7	1.07 ± 0.32	1.76 ± 0.61	1.77 ± 0.08
p = 10	1.12 ± 0.16	1.14 ± 0.39	1.80 ± 0.06
p = 20	2.41 ± 0.02	2.99 ± 0.50	2.42 ± 0.01
p = 30	2.36 ± 0.11	3.98 ± 0.74	2.48 ± 0.03
p = 40	2.28 ± 0.14	4.05 ± 0.46	2.55 ± 0.07
p = 50	2.34 ± 0.08	4.66 ± 0.54	2.50 ± 0.03

4.2.3 Comparison under varying the number of peaks

Experiments in this sub-section are designed to investigate the performance of the algorithm when the number of peaks varies (severity s = 1 for all experiments in this sub-section) and the results are shown in Table 3 and Figure 2. The experiments are compared with MDUMDA and MQSO and the number of peaks in these experiments

changes between the ranges from 1 to 50. Here the configuration of MQSO is set to be $10(5 + 5^q)$.

Figure 2 The performance under varying the number of peaks (see online version for colours)



From Figure 2, it can be seen that the offline error of EDASS increases gradually with the increasing the number of peaks until the number of peaks reach to 20. Generally speaking, a large number of peaks need more clusters to locate and track, which means that an algorithm with a good diversity maintaining mechanism may perform well to find more relatively better local optima. While the offline error influences reposeful during the number of peaks increases from 20 to 50. The reason for this is that when the number of peaks increases, there will be more local optima that have a similar height as the global optima and hence, there will be a higher probability for algorithms to find relatively better local optima. Comparing the results of EDASS with the other two algorithms, the offline error achieved by EDASS is much less than that achieved by the other algorithms. So EDASS performs were better than MDUMDA and MQSO. The experiment confirms that the clustering approach is able to cope well with environments consisting of a large number of peaks.

5 Conclusions and future work

This paper investigated and improved EDAs for locating and tracking multiple optima in dynamic environments. In order to track and locate multiple optima, an adaptive leader clustering method with low computation complexity and Scatter search were applied in EDASS. Experimental studies on the MPB are carried out to evaluate the performance of the proposed algorithm in comparison with several state-of-the-art algorithms.

There are several relevant works to pursue in the future. How to deal with the environmental changes is another important issue. More effective methods rather than a simple restart with elitism scheme need introducing to address the dynamism in DOPs.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant No. 71061001 and No. 61272020.

References

Branke, J. (2002) *Evolutionary Optimization in Dynamic Environments*, Kluver Academic Publishers, Massachusetts, USA.

Fan, J., Xu, Q. and Liang, Y. (2013) 'A novel classification learning framework based on estimation of distribution algorithms', *International Journal of Computing Science and Mathematics*, Vol. 3, No. 1, pp.353–366.

Goncalves, A.R. (2011) 'Online learning in estimation of distribution algorithms for dynamic environments', in *Proc. Evolutionary Computation*, pp.62–69.

Li, C. and Yang, S. (2009) 'A clustering particle swarm optimizer for dynamic optimization', in *Proc. Congr. Evol. Comput.*, November, pp.439–446.

Parrott, D. and Li, X. (2006) 'Locating and tracking multiple dynamic optima by a particle swarm model using speciation', *IEEE Trans. Evol. Comput.*, August, Vol. 10, No. 2, pp.440–458.

Ruan, D. and Zhang, X. (2013) 'The dynamic optimization strategy research of wireless sensor network based on multi-populations cultural framework', *International Journal of Advancements in Computing Technology*, Vol. 5, No. 1, pp.393–404.

Wang, H., Wu, Z. and Rahnamayan, S. (2011) 'Enhancing particle swarm optimization using generalized opposition-based learning', *Information Sciences*, Vol. 8, No. 2, pp.4699–4714.

Wu, Y. and Wang, Y. (2008) 'Multi-population univariate marginal distribution algorithm for dynamic optimization problems', *Control and Decision*, December, Vol. 23, No. 1, pp.1401–1406.