# An Estimation of Distribution Algorithm Guided by Mean Shift

Hui Fang, Aimin Zhou, and Guixu Zhang
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology
East China Normal University,500 Dongchuan Road, Shanghai, China
Email: 51151201029@stu.ecnu.edu.cn , {amzhou,gxzhang}@cs.ecnu.edu.cn

***Abstract*—The estimation of distribution algorithm is widely used to solve global optimization problems in recent years. The basic idea is using machine learning methods to extract relevant features of the search space among the selected individuals and to construct a probabilistic model for sampling new solutions. As we know, EDAs mainly focus on the global distribution information of population and are lack of solution location information. In this paper, we extend our previous work to propose a new EDA guided by the mean shift method, which is originally proposed as a density estimation method and is used as a local search method in this paper. In the new approach, at first a set of candidate solutions are generated by EDA. Then the mean shift method is used to refine some good parent solutions. Finally the sampled candidate solutions and the refined solutions are combined to form the offspring solutions. By this way, the global distribution information and the solution location information are used in offspring reproduction. We apply the new approach to a set of test instances and the experiment results indicate that the new algorithm can obtain good performance in most functions with a faster convergence rate.**

***Index Terms*—estimation of distribution algorithm, mean shift, local search, global optimization**

## I. Introduction

In this paper, we consider the following box-constrained continuous global optimization problem.

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \Omega, \end{array} \tag{1}$$

where $x = (x^1, x^2, \cdots, x^d)^T$ is a decision variable vector, $\Omega = [a^i, b^i]^d$ is the feasible region of the decision space ($a^i < b^i$, $a^i \in R$ and $b^i \in R$ are the lower and upper boundaries of the decision space in the $i$th dimension, respectively), $f(x) : \Omega \to R$ is an objective function, and $R$ is the objective space.

There exists a variety of methods to deal with the global optimization problems. Among them, the *evolutionary algorithms (EAs)* [1] have been attracting much attention partly due to their weak assumptions on the optimization problem. Different EAs have been proposed, such as *genetic algorithm (GA)* [2], *particle swarm optimization (PSO)* [3], and *different evolution(DE)* [4]. *Estimation of distribution algorithms (EDAs)* [5] is another EA paradigm. Their major character is that they can produce new individuals by predicting the best search area through the search space sampling and statistical learning, while general EAs use crossover and mutation operators to generate new individuals.

In the main procedure of an EDA [6] [7], a probabilistic model is constructed first by extracting the global information from the population. Then new solutions are created by sampling from the probabilistic model. The concept of EDAs was first proposed in 1996 [5] and it has been developed rapidly in recent years. Different strategies have been applied to improve the performance of EDAs, such as probability model improvement [5], [8], population diversity maintaince [9], [10], hybrid EDA [11], [12], [13], [14], [15], [16], [17].

It is noted that EDAs pay more attention to the global distribution information of the population in offspring generation, whereas traditional EAs utilize crossover and/or mutation operators to generate offspring solutions, which pay more attention to the individual location information. An ideal algorithm should use both the two kinds of information. Following this idea, some algorithms have been proposed, such as the guided mutation [18], and the EDA and DE hybrid method [19].

Recently, we have also proposed an algorithm by combining EDA with expensive and cheap local search, denoted as EDA/LS [20], to improve the algorithm performance. The EDA part utilizes the global population distribution information, while the local search part utilizes the individual location information. In this paper, we propose an algorithm, named EDA/LS-MS, to find the optimal solution using information of global distribution and local search. This paper extends the work in EDA/LS by replacing the cheap local search by the mean shift method [21]. Mean shift is a nonparametric density estimation method. It can also be applied to do optimization [22], which does not need any function evaluation to guide the optimization. This is the motivation why we try to use it and this paper reports our preliminary work on this investigation.

The rest of this paper is organized as follows. In Section II, the improved version of EDA/LS by mean shift is introduced in detail. The experiment result and analysis are given in Section III. Finally, this paper is concluded in Section IV.

## II. Proposed Algorithm

### A. Algorithm Description

Denote the EDA/LS guided by mean shift as EDA/LS-MS. In each generation, EDA/LS-MS maintains

- a set of $N$ populations $\{x_1, x_2, \cdots, x_N\}$,

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

- their objective values $\{f(x_1), f(x_2), \cdots, f(x_N)\}$.

The algorithm framework of the proposed EDA/LS-MS is shown in Algorithm 1.

**Algorithm 1:** Procedure of EDA/LS-MS

```
1  Initialize the population $pop = \{x^1, \cdots, x^N\}$ and evaluate them;
2  while $not\ terminate$ do
3      $M \leftarrow model(pop)$.
4      $\{y_1, \cdots, y_N\} \leftarrow sample(M)$.
5      $\{z_1, \cdots, z_{\lfloor NP_a \rfloor}\} \leftarrow MS(pop, P_a)$.
6      foreach $y \in \{y_1, \cdots, y_N\}$ do
7          $k \leftarrow random\_select\{1, 2, \cdots, \lfloor NP_a \rfloor\}$.
8          for $j$ = $1$ to $n$ do
9              if $rand() < P_b$ then
10                 $y^j \leftarrow z^j_k$.
11             end
12         end
13     end
14     Evaluate $\{y_1, \cdots, y_N\}$.
15     $pop \leftarrow select\left(\{y_1, \cdots, y_N\} \cup \{x_1, \cdots, x_N\}\right)$.
16     if $convergent$ then
17         $t \leftarrow random\_select\{1, 2, \cdots, N\}$.
18         $x^* \leftarrow Powell\_search(x_t)$.
19         if $f(x^t) < f(x^*)$ then
20             $x^t \leftarrow x^*$.
21         end
22     end
23 end
```

The algorithm is the same as EDA/LS except in *Line 5* where the mean shift based search is applied. More details of the algorithm are referred to [20].

We would like to describe the algorithm from several perspectives as follow:

- *Population Initialization:* The initial population is uniformly and randomly sampled from the search space in *Line 1*.
- *Termination Condition:* The EDA/LS-MS terminates when a given maximum number of generations is reached in *Line 2*.
- *Selection Procedure:* The best solutions according to the objective function values are selected form the combination of the parent solutions and the new solutions in *Line 14*.
- *Reproduction Procedure:* Firstly, a probabilistic model is built based on the current population in *Line 3*. Then a set of candidate solutions are sampled form the model in *Line 4*, and some reference solutions are generated by the mean shift search in *Line 5*. Some components of the candidate solutions are replaced by the reference solutions in *Lines 6-14* and the procedure of MS will be given later. The concrete method of combination is in *Lines 6-13*. When the algorithm converges, the Powell based expensive local search is applied to improve a randomly selected solution in *Lines 16-22*.

Some details of the method will be discussed shortly in the following section.

### B. Mean Shift Based Search

Mean shift is a non-parametric feature-space analysis technique for locating the maxima of a density function. There is no need to calculate the values of estimation functions when we use mean shift to search optimal solutions. We have proved in [23] that the maximization of the density of a distribution equals to the maximization of a corresponding objective. In this section, we propose a local search strategy based on mean shift.

As proposed in [21], the mean shift algorithm assumes that the data points in a $d$-dimensional Euclidean space $R^d$ are independent and identically distributed samples taken from a population with an unknown density function $\hat{f}(x)$. The kernel density estimation at the location $x$ can be calculated as follows:

$$\hat{f}_{h,K}(x) = \frac{1}{nh^d} \sum_{i=1}^{N} K\left( \left\| \frac{x - x_i}{h} \right\|^2 \right), \tag{2}$$

with the bandwidth parameter $h > 0$ and the scalar kernel function $K$ under the conditions given in [21].

To maximize $\hat{f}(x)$, the gradient of the density estimator is formulated by exploiting the linearity of (2):

$$\hat{\bigtriangledown} f_{h,K}(x) = \frac{1}{nh^d} \left[ \textstyle\sum_{i=1}^{N} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \right] m_{h,g}(x), \tag{3}$$

where

$$m_{h,g}(x) = \frac{\sum_{i=1}^{N} x_i g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{N} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \tag{4}$$

and

$$g(x) = -K'(x). \tag{5}$$

In order to obtain the direction of the point with maximum density, the value in (3) should equal to zero. Hence, we can attain the new coordinate value of $x$ as follow:

$$x_{new} = \frac{\sum_{i=1}^{N} x_i g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{N} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}, \tag{6}$$

It is obvious that $x$ will converge at a nearby point where the multivariate kernel density estimator has zero gradient after sufficient iterations. Fig. 1 illustrates the idea of applying the mean shift method on searching for the local density maxima. The initial point keeps moving to the new mean location along the direction of the nearest stationary point.

TABLE I
THE 12 TEST INSTANCES USED IN COMPARISON STUDY

| Test Instance | Search Space |
|---|---|
| $f_1(x) = \sum_{i=1}^{d} (x^i)^2$ | $[-100, 100]^d$ |
| $f_2(x) = \sum_{i=1}^{d} \lvert x^i \rvert + \prod_{i=1}^{d} \lvert x^i \rvert$ | $[-10, 10]^d$ |
| $f_3(x) = \sum_{i=1}^{d} \left( \sum_{j=1}^{i} x^j \right)^2$ | $[-100, 100]^d$ |
| $f_4(x) = \max \left\{ \lvert x^i \rvert \right\}$ | $[-100, 100]^d$ |
| $f_5(x) = \sum_{i=1}^{d-1} \left[ 100(x^{i+1} - (x^i)^2)^2 + (x^i - 1)^2 \right]$ | $[-30, 30]^d$ |
| $f_6(x) = \sum_{i=1}^{d} \lfloor x^i + 0.5 \rfloor^2$ | $[-100, 100]^d$ |
| $f_7(x) = \sum_{i=1}^{n} i(x^i)^4 + rand[0, 1)$ | $[-1.28, 1.28]^d$ |
| $f_8(x) = \sum_{i=1}^{n} -(x^i) \sin(\sqrt{\lvert x^i \rvert})$ | $[-50, 50]^d$ |
| $f_9(x) = \sum_{i=1}^{d} \left[ (x^i)^2 - 10 \cos(2\pi x^i) + 10 \right]$ | $[-5.12, 5.12]^d$ |
| $f_{10}(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} (x^i)^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x^i) \right) + 20 + e$ | $[-32, 32]^d$ |
| $f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{d} (x^i)^2 \prod_{i=1}^{d} \cos \left( \frac{x^i}{\sqrt{i}} \right) + 1$ | $[-600, 600]^d$ |
| $f_{12}(x) = \frac{\pi}{d} \left\{ 10 \sin^2(\pi y^i) + \sum_{i=1}^{d-1} (y^i - 1)^2 [1 + 10sin^2(\pi y^i + 1)] + (y^d - 1)^2 \right\} + \sum_{i=1}^{d} u(x^i, 10, 100, 4)$<br>where $y^i = 1 + \frac{1}{4}(x^i + 1)$, and<br>$u(x^i, a, k, m) = \begin{cases} k(x^i - a)^m & x^i > a \\ 0 & -a \leq x^i \leq a \\ k(-x^i - a)^m & x^i < -a \end{cases}$ | $[-50, 50]^d$ |

In standard method, the bandwidth is a constant value, which means it might not have much impact on the procedure of density estimation when the distances of all data points are very small. So, we provide a method to update the bandwidth adaptively and make some improvements in (6):

$$x_{new} = \frac{\sum_{i=1}^{m} x_i \varrho \left( \left\| \frac{x - x_i}{s} \right\|^2 \right)}{\sum_{i=1}^{m} \varrho \left( \left\| \frac{x - x_i}{s} \right\|^2 \right)}, \tag{7}$$

where $\|x\|^2$ is $L_2$ norm of $x$, $m$ is the number of points that are used to calculate the coordinate value of new solution, $s$ is the adaptive bandwidth of searching window calculated as follow:

$$s = \sqrt{\frac{1}{d} \sum_{j=1}^{d} (p^j - q^j)^2} \tag{8}$$

where $p$ and $q$ is the maximum and the minimum values of all the solutions in the current population.

It should be noted that the kernel function in the mean shift process essentially imposes additional weights on the data points according to their distances to the shifting mean [24]. Cheng [22] described two main types of kernel functions that have frequently been used in the mean shift research, namely flat and Gaussian kernel. Compared to Gaussian kernels, flat kernel gives these points that close to the initial point the same weights, which ignores the influence of the distance between

TABLE II

MEDIAN,STD. VALUES OF THE RESULTS OBTAINED BY THE TWO COMPARISON ALGORITHMS AFTER 500, 1000, AND 1500 GENERATIONS OVER 20 RUNS FOR ALL THE TEST INSTANCES.

| | $generation = 500$ | | $generation = 1000$ | | $generation = 1500$ | |
|---|---|---|---|---|---|---|
| | EDA/LS-MS | EDA/LS | EDA/LS-MS | EDA/LS | EDA/LS-MS | EDA/LS |
| $f_1$ | **2.78e-33**±6.01e-020 | 6.74e-24±3.81e-24 | **1.11e-65**±3.13e-66 | 2.34e-65±2.77e-65 | 8.72e-66±1.88e-66 | **1.35e-99**±1.31e-99 |
| $f_2$ | **1.11e-15**±1.07e-09 | 6.64e-12±2.02e-12 | **3.31e-44**±1.07e-30 | 1.22e-31±5.15e-32 | 6.32e-48±5.55e-44 | **5.04e-48**±1.73e-48 |
| $f_3$ | **2.25e-02**±1.24e-01 | 2.18e+02±1.73e+02 | **1.51e-36**±2.80e-33 | 2.60e-31±2.93e-02 | **1.18e-36**±8.38e-36 | 2.90e-36±6.27e-34 |
| $f_4$ | **1.27e-102**±1.38e-00 | 1.40e-07±3.93e-08 | **5.99e-103**±6.29e-16 | 1.32e-20±5.55e-21 | **2.21e-103**±6.12e-16 | 1.56e-31±8.17e-32 |
| $f_5$ | **3.66e-10**±4.69e-06 | 9.69e-07±7.80e-03 | 4.81e-29±5.14e-29 | **3.67e-29**±2.81e-29 | 3.09e-29±2.52e-29 | **3.05e-29**±2.06e-29 |
| $f_6$ | **0.00e+00**±0.00e+00 | **0.00e+00**±0.00e+00 | **0.00e+00**±0.00e+00 | **0.00e+00**±0.00e+00 | **0.00e+00**±0.00e+00 | **0.00e+00**±0.00e+00 |
| $f_7$ | **4.39e-03**±2.41e-03 | 6.44e-03±2.86e-03 | **3.13e-03**±1.03e-03 | 5.05e-03±1.30e-03 | **2.44e-03**±9.62e-04 | 3.70e-03±8.62e-04 |
| $f_8$ | **1.82e-12**±2.22e-06 | 7.00e-11±3.16e-06 | **0.00e+00**±4.07e-13 | **0.00e+00**±0.00e+00 | **0.00e+00**±4.07e-13 | **0.00e+00**±0.00e+00 |
| $f_9$ | **0.00e+00**±3.90e-06 | 7.96e+00±3.30e+00 | **0.00e+00**±0.00e+00 | 2.98e+00±2.37e+00 | **0.00e+00**±0.00e+00 | **0.00e+00**±5.21e-01 |
| $f_{10}$ | **6.84e-14**±2.07e-11 | 8.30e-13±1.78e-13 | 7.99e-15±1.74e-15 | **4.44e-15**±0.00e+00 | 7.99e-15±0.00e+00 | **4.44e-15**±0.00e+00 |
| $f_{11}$ | **0.00e+00**±3.32e-03 | **0.00e+00**±0.00e+00 | **0.00e+00**±3.32e-03 | **0.00e+00**±0.00e+00 | **0.00e+00**±3.32e-03 | **0.00e+00**±0.00e+00 |
| $f_{12}$ | **1.57e-32**±2.81e-48 | 1.67e-25±5.22e-26 | **1.57e-32**±2.81e-48 | **1.57e-32**±2.81e-48 | **1.57e-32**±2.81e-48 | **1.57e-32**±2.81e-48 |

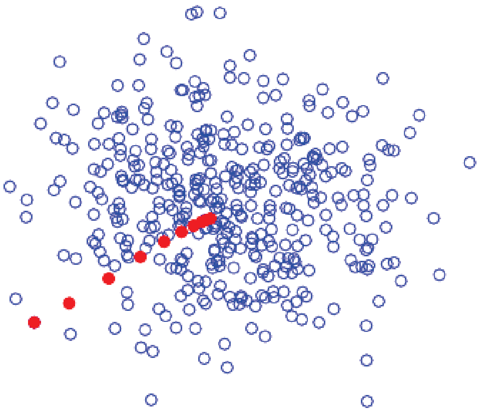


Fig. 1. Illustration of the search procedure of a mean shift

two points. In the paper, we use the following Gaussian kernel.

$$g(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(u)^2}. \tag{9}$$

where

$$u = \left\| \frac{x - x_i}{s} \right\|^2. \tag{10}$$

The value of $u$ represents the distance between two points. To make the algorithm more effective, the value of kernel function will be set as zero when $u > 2$. Under this condition, the distant point will not have any impact on the procedure of mean shift.

The detailed algorithm framework of the mean shift based search strategy is described in Algorithm 2.

```
Algorithm 2: $\{z_1, \cdots, z_{\lfloor NP_a \rfloor}\} \leftarrow MS(pop, P_a)$
1 Set $p^j = \max_{x \in pop} x^j$ and $q^j = \min_{x \in pop} x^j$ for
  $j = 1, \cdots, d$.
2 Calculate $s$ as (8).
3 Choose the best $\lfloor NP_a \rfloor$ solutions from $pop$ and save
  them in $Q$.
4 For each $x \in Q$, estimate a new position $z$ by (7).
```

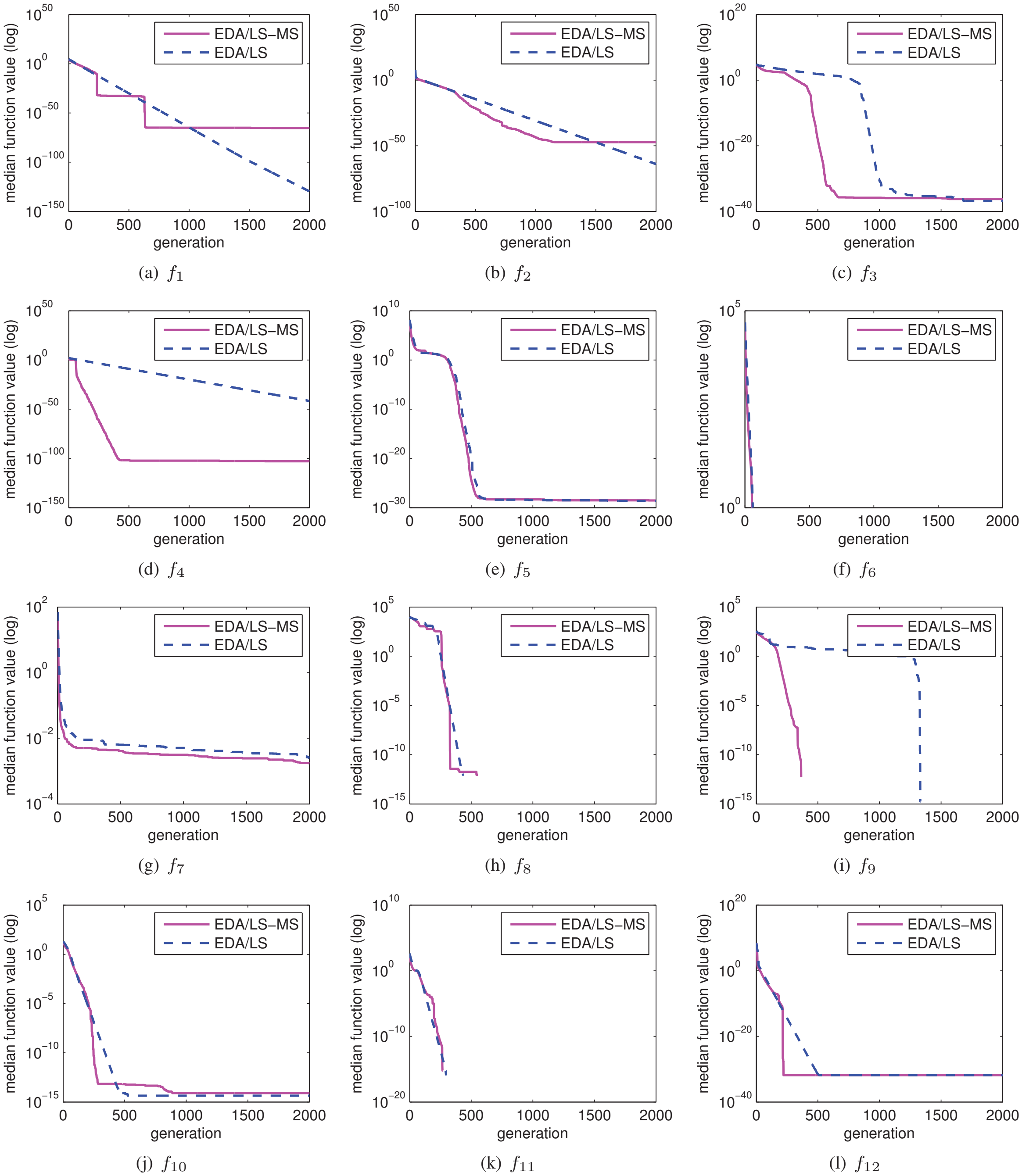
## III. EXPERIMENTAL RESULTS

In this section, we compare EDA/LS-MS with EDA/LS on 12 widely used benchmark functions from [25]. The 12 benchmark functions contain nine unimodal functions $f_1$-$f_9$ and 3 multimodal functions $f_{10}$-$f_{12}$. All the test instances have a global minimum objective value of 0. The details of these test instances are listed in Table I.

### A. Parameter Settings

The parameters in the experiment study are as follows:

- The population size $N = 150$ and the dimension of the test instance is $n = 30$ for all test instances.
- The maximum number of generations is $G = 2000$ and the two algorithms are executed independently for 20 runs for each instances.
- EDAs : The number of bins used in a VWH model, which is proposed in [12], is $M = 15$;
- The proportion of the new solutions generated by MS in the EDAs solutions $P_a = 0.2$; The probability to use the location information $P_b = 0.2$; and the convergence threshold is $\theta = 0.1$;



(a) $f_1$ (b) $f_2$ (c) $f_3$

(d) $f_4$ (e) $f_5$ (f) $f_6$

(g) $f_7$ (h) $f_8$ (i) $f_9$

(j) $f_{10}$ (k) $f_{11}$ (l) $f_{12}$

Fig. 2. The median function value of the best solutions obtained by the two methods versus function evaluation on f1-f12.

All the algorithms are implemented in Matlab R2010b and executed in Lenovo Thinkpad E430 with i5-3210M CPU @ 2.50GHz, 6.00GB RAM, and Windows 7.

### B. Comparison Results and Analysis

EDA/LS-MS is compared with EDA/LS on the 12 test instances. The median and standard deviation of the results obtained by the two algorithms after 500, 1000, and 1500 generations over 20 independent runs are shown in Table II. Fig. 2 shows the median function values obtained by the two algorithms versus the generations on $f_1$-$f_{12}$.

In Table II, it can be seen clearly that EDA/LS-MS outperforms EDA/LS in most functions after 500 generations and both EDA/LS-MS and EDA/LS converges to the global optimal on $f_6$ and $f_9$. EDA/LS-MS achieves the goal on $f_9$ and $f_{12}$ though EDA/LS dose not reach the stage of convergence. With the increase of the number of generations, It can be seen that the results in EDA/LS and EDA/LS-MS are improved

TABLE III

MEAN±STD. VALUES OF THE RESULTS OBTAINED BY EDA/LS-MS AFTER $2,000$ GENERATIONS OVER $20$ RUNS FOR ALL THE TEST INSTANCES WITH DIFFERENT VALUES OF ($s = 10, 15, 20,$AND AN ADAPTIVE VALUE).

| | 10 | 20 | 30 | $adaptive$ |
|---|---|---|---|---|
| $f_1$ | **1.865e-066**±2.087e-066 | 3.582e-066±4.225e-066 | 3.322e-066±3.159e-066 | 3.450e-066±4.064e-066 |
| $f_2$ | **1.686e-046**±1.408e-044 | 1.710e-046±1.289e-044 | 3.248e-046±5.395e-044 | 3.580e-046±5.552e-044 |
| $f_3$ | **2.922e-034**±2.849e-034 | 2.179e-033±2.397e-036 | 7.847e-034±4.143e-034 | 7.727e-033±5.184e-036 |
| $f_4$ | 2.399e-103±1.283e-015 | 6.241e-103±5.369e-015 | 3.572e-103±2.746e-016 | **2.372e-103**±1.358e-014 |
| $f_5$ | **2.502e-029**±3.255e-029 | 2.836e-029±3.223e-029 | 2.787e-029±3.948e-029 | 3.006e-029±2.621e-029 |
| $f_6$ | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 |
| $f_7$ | 1.638e-003±1.467e-003 | 1.726e-003±1.712e-003 | **4.010e-004**±3.844e-004 | 5.634e-004±6.016e-004 |
| $f_8$ | **0.000e+000**±9.095e-014 | 9.095e-014±9.095e-014 | **0.000e+000**±4.067e-013 | **0.000e+000**±4.067e-013 |
| $f_9$ | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 | **0.000e+000**±0.000e+000 |
| $f_{10}$ | 8.349e-015±7.994e-015 | 7.994e-015±7.994e-015 | **1.589e-015**±0.000e+000 | 7.994e-015±0.000e+000 |
| $f_{11}$ | **0.000e+000**±0.000e+000 | 9.856e-004±1.602e-003 | **0.000e+000**±0.000e+000 | **0.000e+000**±3.324e-003 |
| $f_{12}$ | 1.571e-032±1.571e-032 | 1.571e-032±1.571e-032 | **2.808e-048**±2.808e-048 | **2.808e-048**±2.808e-048 |

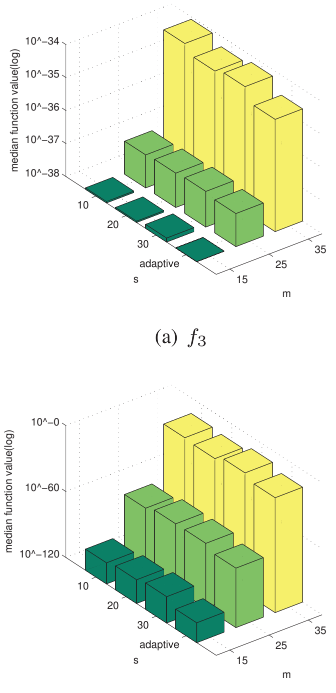
obviously on $f_1$-$f_5$ after 1000 generations. In the meanwhile, the statistical results of the two algorithms are consistent with each other on $f_1$, $f_5$-$f_{12}$ and EDA/LS-MS performs better than EDA/LS on $f_2$-$f_4$. EDA/LS continues to converge whereas EDA/LS-MS remains moderate after 1500 generations and the two algorithms both perform well on $f_1$, $f_3$-$f_6$, $f_8 - f_{12}$. In the finial stages, the results of EDA/LS are better than those of EDA/LS-MS in most test instances. It is evident that EDA/LS-MS converges quickly in its early stage and gets a better result than EDA/LS on $f_4$.

As is shown in Fig. 2, EDA/LS-MS converges faster than EDA/LS on most test instances. It it obvious that EDA/LS-MS reaches the convergent stage earlier than EDA/LS on $f_1$-$f_4$, $f_8$-$f_{10}$, $f_{12}$ and the speed of convergence of the two algorithms are consistent with each other on $f_5$-$f_7$ and $f_{11}$. In EDA/LS-MS, there are very significant downward trends of median function value on $f_3$, $f_4$, $f_9$ and $f_{12}$. The reason that EDA/LS-MS converges faster than EDA/LS could be that the mean shift method plays an significant role at the convergence rate in searching for a local optimal solution .

### C. Sensitivity to Control Parameters

In this section, the influence of control parameters on the EDA/LS-MS will be discussed. During the procedure of mean shift, bandwidth and the number of chosen solutions in (7) are the major control parameters. We use $s$ and $m$ to denote them respectively.

In the procedure of calculating the value of density estimation, the bandwidth of the searching window can be defined as a constant or a variable. $m$ is the number of solutions that used to calculate the coordinate value of a new solution. In general, the solution of the better function value is chosen as the candidate. We set $s$ = 10, 20, 30, and the adaptive value in (8), $m$ = 15, 25, and 35. The other parameters of EDA/LS-MS are set as the values used in the previous section. $f_3$ and $f_4$ are chosen for the study.



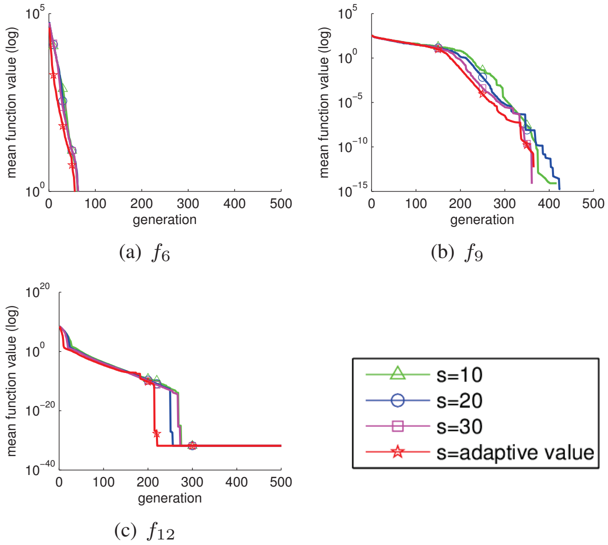
(a) $f_3$

(b) $f_4$

Fig. 3. The median function values obtained by EDA/LS-MS with different parameter settings on $f3$ and $f4$.

It is clear from Fig. 3 that $m$ has a significant impact on the performance of EDA/LS-MS. The statistical results become better with the decline of value $m$ and reach the best stages when $m = 15$. It can be seen that $m$ makes a great improvement in the performance on $f_4$. EDA/LS-MS with $m$ = 15 performs better than EDA/LS-MS with $m$ = 25 and 35. Furthermore, we can see that when $s$ is the adaptive value, EDA/LS-MS with $m$ = 15 performs the best while when $s$ = 30, EDA/LS-MS with $m$ = 35 performs the worst. From the plot, the influence of $s$ on the performance is very small and the results of function keep steady with the four different values of $s$. It can be noted that EDA/LS-MS performs a little better when $s$ is the adaptive value on $f_3$ and $f_4$ than other constant values.

As we can see, there is not much difference in the performance of EDA/LS-MS with four distinct values of bandwidth. The experiment results of the mean and std. deviation values of the algorithm over 20 runs for the test instances after 2000 generations with different values of bandwidth are shown in Table III. It is obvious that all of them perform well and achieve the optimal zero on $f_6$, $f_9$, and $f_{11}$. On $f_{12}$, EDA/LS-MS with $s$ = 30 and the adaptive value outperform than EDA/LS-MS with $s$ = 10 and 20.



(a) $f_6$

(b) $f_9$

(c) $f_{12}$

Fig. 4. The mean function values obtained by EDA/LS-MS with different settings of bandwidth versus generations on $f_6$, $f_9$, and $f_{12}$.

The change of bandwidth does not affect the value of final solution. However, the speed of convergence is largely influenced by the choice of bandwidth in most test instances. Fig. 4 shows the average objective values versus generations on $f_6$, $f_9$ and $f_{12}$ respectively with different settings of $s = 10, 20, 30$, and an adaptive value in (8). We can see that the value of function decline sharply after 200 generations on $f_{12}$ with the adaptive value. It is obvious that the variable bandwidth outperforms other constants by the speed of achieving the convergence. The strategy of setting a variable value to the bandwidth can raise the efficiency of convergence in EDA/LS-MS.

## IV. Conclusions

The estimation of distribution algorithm is widely used to solve global optimization problems. It is noted that new solutions are created by probabilistic models in EDA without the location information from parents. In this paper, we propose a new algorithm, named EDA/LS-MS, to guide EDA with mean shift. In this approach, we combine both population distribution information and individual location information for generating better offspring solutions to speed up the algorithm convergence.

To evaluate the performance of our approach, 12 test instances are used as the benchmark functions. The experiment results indicate that EDA/LS-MS can obtain good performance and have a faster convergence rate compared to EDA/LS on most of functions. The reason might be that mean shift can help to locate promising areas quickly. We also empirically studied the influences of bandwidth in the mean shift with different values. The results suggest that adaptive bandwidth is a better choice than other constant values.

The work reported in this paper is preliminary and some work has been left for further investigation. In the future, the influence of different kernel function in mean shift will be studied. Beside, combining mean shift with other EAs is another direction.

## Acknowledgment

This work is supported by China National Instrumentation Program under Grant No.2012YQ180132, the National Natural Science Foundation of China under Grant No.61273313, and the Science and Technology Commission of Shanghai Municipality under Grant No.14DZ2260800.

## References

[1] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of evolutionary computation*. IOP Publishing Ltd., 1997.

[2] D. E. Goldberg and M. P. Samtani, “Engineering optimization via genetic algorithm,” in *Electronic Computation (1986)*. ASCE, 1986, pp. 471–482.

[3] J. Kennedy, “Particle swarm optimization,” in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[4] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[5] L. dos Santos Coelho, “A quantum particle swarm optimizer with chaotic mutation operator,” *Chaos, Solitons & Fractals*, vol. 37, no. 5, pp. 1409–1418, 2008.

[6] H. Mühlenbein and G. Paass, “From recombination of genes to the estimation of distributions i. binary parameters,” in *Parallel Problem Solving from NaturełPPSN IV*. Springer, 1996, pp. 178–187.

[7] H. Mühlenbein, J. Bendisch, and H.-M. Voigt, “From recombination of genes to the estimation of distributions ii. continuous parameters,” in *Parallel Problem Solving from NaturełPPSN IV*. Springer, 1996, pp. 188–197.

[8] W. Dong and X. Yao, “Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms,” *Information Sciences*, vol. 178, no. 15, pp. 3000–3023, 2008.

[9] Y.-H. Cheng, X.-S. Wang, and M.-L. Hao, “An estimation of distribution algorithm with diversity preservation,” *Dianzi Xuebao(Acta Electronica Sinica)*, vol. 38, no. 3, pp. 591–597, 2010.

[10] L. DelaOssa, J. Gámez, J. L. Mateo, J. M. Puerta *et al.*, “Avoiding premature convergence in estimation of distribution algorithms,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*. IEEE, 2009, pp. 455–462.

[11] E. Bengoetxea and P. Larrañaga, "Eda-pso: a hybrid paradigm combining estimation of distribution algorithms and particle swarm optimization," in *Swarm Intelligence*. Springer, 2010, pp. 416–423.

[12] X. Wang, Y. Cheng, and M. Hao, "Estimation of distribution algorithm based on bacterialforaging and its application in predictive control," *acta electronica sinica*, vol. 38, pp. 333–339, 2010.

[13] T. Miquélez, E. Bengoetxea, A. Mendiburu, and P. Larrañaga, "Combining bayesian classifiers and estimation of distribution algorithms for optimization in continuous domains," *Connection Science*, vol. 19, no. 4, pp. 297–319, 2007.

[14] J. A. Lozano, *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*. Springer Science & Business Media, 2006, vol. 192.

[15] A. S. Lodge, "Constitutive equations from molecular network theories for polymer solutions," *Rheologica Acta*, vol. 7, no. 4, pp. 379–392, 1968.

[16] X. Huang, P. Jia, and B. Liu, "Controlling chaos by an improved estimation of distribution algorithm," *Mathematical and Computational Applications*, vol. 15, no. 5, pp. 866–871, 2010.

[17] H. Wu, W.-P. Wang, L. Wang, and F. Yang, "Research on cooperative optimization of improved estimation of distribution algorithm," *Jisuan-ji Gongcheng yu Yingyong(Computer Engineering and Applications)*, vol. 46, no. 26, 2010.

[18] Q. Zhang, J. Sun, and E. Tsang, "An evolutionary algorithm with guided mutation for the maximum clique problem," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 2, pp. 192–200, 2005.

[19] J. Sun, Q. Zhang, and E. P. Tsang, "De/eda: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3, pp. 249–262, 2005.

[20] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 6, pp. 807–882, 2015.

[21] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, pp. 32–40, 1975.

[22] Y. Cheng, "Mean shift, mode seeking, and clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 790–799, 1995.

[23] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 5, pp. 746–758, 2015.

[24] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[25] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.