

# Estimation of Distribution Algorithm using Variety of Information

Juan Yu  
College of Marine  
Northwestern Polytechnical University  
Xi'an 710072, China  
yujuan@snnu.edu.cn

Yuyao He  
College of Marine  
Northwestern Polytechnical University  
Xi'an 710072, China  
heyyao@nwpu.edu.cn

## ABSTRACT

Former information of probability model and inferior individuals were discarded in the research of estimation of distribution algorithm usually, but they may contain useful information. In this paper, the former probability information is introduced to avoid premature convergence caused by continuously select superior individuals of current population to build probability model, and the individual sampling from superior probability model is filtered by inferior probability model to avoid generating inferior individuals. The algorithm is simulated through the widely used knapsack examples, the results verify the validity of the proposed method, and give suggestion for the choice of parameter through simulation and analysis.

## Categories and Subject Descriptors

G.1.6 [optimization]: constrained optimization

## General Terms

Algorithms, Performance, Experimentation, Verification.

## Keywords

Estimation of distribution algorithm, probability model, former information of probability model, inferior individuals

## 1. INTRODUCTION

Estimation of distribution algorithm (EDA), the combination of statistical learning and evolutionary algorithms, is proposed to solve the building block damage problem of genetic algorithm, and has been claimed as a paradigm shift in the field of evolutionary computation[1]. EDA get estimate of the probability distribution of superior solutions set, then new solutions are generated by sampling the distribution encoded by this model.

The operation is repeated to realize the evolution of population. Sampling probability model avoids damage to the promising solutions[2], which is contrary to the operation of crossover and mutation in genetic algorithm, it is also the main characteristic different from genetic algorithm. So estimation of distribution algorithm has also been called probability model building genetic

algorithms[1]. EDA controls the evolutionary direction of population from macro, solves high dimensional problem and difficult optimization problem effectively and reduces the time complexity[3]. It has been widely studied in recent years[4-9].

EDA generates new individuals through sampling from probability model, so probability model is the key factor, efficient probability model can produce high quality solutions and diversity solutions, how to build the probability model is the key to EDA. In most EDAs, only superior solutions of current population are used to build probability models and then sampling from the probability model to produce new individuals, the information of former probability model is discarded, thus it may lead to overfitting of the distribution of solution space when learning the probability model in the process of evolution, and also result in no longer generate diversity solutions after several generations and result in premature convergence. Therefore the information of former probability model is used to improve the estimation of distribution in this paper.

The Ref [10] proposed to use inferior solutions when generate new individuals. In this paper, the method will be modified and combined with utilization of former probability model to generate a new EDA.

The rest of this paper is organized as follows: Section 2 describes basic estimation of distribution algorithm. Section 3 describes the two operators, one is the use of information of former probability model, the other is the use of information of inferior solutions. Experimental results are presented in section 4. Finally, we conclude this paper in section 5.

## 2. ESTIMATION OF DISTRIBUTION ALGORITHM

### 2.1 Probability Model

Probability vector  $P(g) = (p(x_1), \dots, p(x_k), \dots, p(x_N))$  is probability model to describe the distribution of solution space,  $p(x_k)$  is the probability of  $x_k = 1$ . Offspring solutions are generated by sampling from probability model, i.e. randomly generate a number  $r$ ,  $r \in [0, 1]$ , if  $r < p(x_k)$ ,  $x_i = 1$ ; otherwise  $x_i = 0$ .

In this paper, the UMDA [11] probability model is adopted, and the probability model is built by subset  $s(g-1)$  which is composed of  $m$  individuals with best fitness in  $pop(g-1)$ ,  $m = \lambda * M$ , where:  $\lambda < 1$ ,  $M$  is population size.  $p(x_i)$  updated by Eq.(1), where  $x_i^j$  is the value of the  $i$ th gene of the  $j$ th

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

EAST'14, May 26, 2014, Nanjing, China  
Copyright 2014 ACM 978-1-4503-2965-1/14/05...\$15.00  
<http://dx.doi.org/10.1145/2627508.2627513>

individual.

$$p(x_i) = \frac{\sum_{j \in s(g-1)} x_i^j}{\lambda * M} \quad (1)$$

The probability of generating solution  $X = (x_1, \dots, x_N)$  is calculated by Eq.(2):

$$p(X) = \prod_{i=1}^N [p(x_i) * x_i + (1 - p(x_i)) * (1 - x_i)] \quad (2)$$

## 2.2 BasicEDA

The information of former probability and inferior solutions will be added on the basis of basicEDA, the framework of basicEDA is shown in Figure. 1

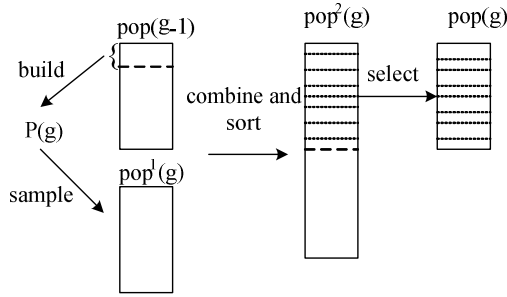


Figure 1: The framework of basicEDA

Build the probability model  $P(g)$  by set  $s(g-1)$  which is composed of superior individuals of the population at iteration  $g-1$ ,  $M$  solutions are generated by sampling from  $P(g)$  and formed population  $pop_1(g)$ , then  $pop(g-1)$  and  $pop_1(g)$  combined to form  $pop_2(g)$ , finally select  $M$  individuals with best fitness from  $pop_2(g)$  as the population  $pop(g)$ .

basicEDA is described as follows:

### Algorithm: basicEDA

#### Begin

1. Generate initial population  $pop(0)$  with  $P(0)=(0.5, \dots, 0.5)$ , let  $g=1$ .

**While**("stopping criterion is not met")

2.1. Update  $P(g)$  by Eq.(1).

2.2. Generate  $pop_1(g)$  by sampling from  $P(g)$ .

2.3.  $pop_2(g) = pop(g-1) \cup pop_1(g)$

2.4. Generate  $pop(g)$  by select  $M$  superior individuals from  $pop_2(g)$ .

2.5.  $g++$ .

**End while**

3. Output result

**End**

## 3. ESTIMATION OF DISTRIBUTION ALGORITHM USING VARIETY OF INFORMATION

In this paper, the following two operators, operator A and operator B, were used based on basicEDA described in

section 2.

### 3.1 Operator A: Make Use of Information of Former Probability

Along with the iteration, EDA will lose diversity, population variance is more and more small, finally decays to zero, probability model evolves to only produce same solutions[12].

The main reason is that EDAs generally select superior individuals of current population to build probability model in each generation. The continuous selection of the best part of the population could lead to overfitting of the distribution of solution space and to a premature homogenisation of the population, therefore, the search process is stagnated. It can be relieved by slowing down the update speed of probability model, making updating speed relatively slower compared to the search speed of algorithm.

To distinguish with inferior solutions probability model below,  $P(g)$  is replaced by  $P_{superior}(g)$  to represent the superior solutions probability model at iteration  $g$ .

So the probability model  $P_{superior}(g)$  can be updated by Eq.(3):

$$P_{superior}(g) = \alpha P_{superior}'(g) + (1 - \alpha) P_{superior}(g-1) \quad (3)$$

$$\text{Where } P_{superior}'(x_i) = \frac{\sum_{j \in s(g-1)} x_i^j}{\lambda * M} \quad (4)$$

$P_{superior}(g)$  is composed of  $P_{superior}'(g)$  and  $P_{superior}(g-1)$ ,  $P_{superior}'(g)$  is built by superior solutions set  $s(g-1)$  of  $pop(g-1)$  as show in Eq.(4), it is same as Eq.(1),  $P_{superior}(g-1)$  is probability model of last iteration.  $\alpha$  is adjustment coefficient,  $\alpha \in (0,1)$ . When  $\alpha = 0$ ,  $P_{superior}(g) = P_{superior}(g-1)$ , the probability model will not be updated, and keep as the initial probability model; when  $\alpha = 1$ ,  $P_{superior}(g) = P_{superior}'(g)$ , it will be same as basicEDA, namely,  $P_{superior}(g)$  is built by the superior solutions of current population only.

$P_{superior}(g)$  is composed of  $P_{superior}'(g)$  and  $P_{superior}(g-1)$ , on one hand,  $P_{superior}'(g)$  and  $P_{superior}(g-1)$  contain information of superior solutions, it ensures the quality of the solutions generated by sampling from  $P_{superior}(g)$ ; on the other hand,  $P_{superior}'(g)$  and  $P_{superior}(g-1)$  are obtained in different ways, it breaks the model of build probability model which only make use of superior solutions set of  $pop(g-1)$ , thus problem of diversity loss can be relieved.

### 3.2 Operator B: Make Use of Information of Inferior Individuals When Generating New Individuals

Most of EDAs only make use of the superior solutions to build probability model, the inferior solutions are regarded as useless and discarded. In fact, the inferior solutions may contain useful information. The distribution characteristic of superior solutions can be known by built probability model from superior solutions, so the distribution characteristic of inferior solution can be known by built probability model from inferior solutions.

Let  $P_{inferior}(g) = (p_{inferior}(x_1), \dots, p_{inferior}(x_N))$  represent the probability model built from inferior solutions.

$P_{inferior}(g)$  can be built through two methods as follows:

**Method 1:**  $P_{inferior}(g)$  is built through  $I(g-1)$  which is composed of  $\lambda * M$  individuals with worst fitness in  $pop(g-1)$  as shown in Eq.(5):

$$p_{inferior}(x_i) = \frac{\sum_{j \in I(g-1)} x_i^j}{\lambda * M} \quad (5)$$

**Method 2:** The same as  $P_{superior}(g)$ , consider the information of former probability,  $P_{inferior}(g)$  consists of two parts:  $P_{inferior}'(g)$  and  $P_{inferior}(g-1)$ , as shown in Eq.(6):

$$P_{inferior}(g) = \alpha P_{inferior}'(g) + (1 - \alpha) P_{inferior}(g-1) \quad (6)$$

$$\text{Where: } p_{inferior}'(x_i) = \frac{\sum_{j \in I(g-1)} x_i^j}{\lambda * M} \quad (7)$$

After built  $P_{superior}(g)$  and  $P_{inferior}(g)$ ,  $P_{superior}(g)$  is used to generate new individuals,  $P_{inferior}(g)$  is used to filter the individual generated by  $P_{superior}(g)$ . For the generated individual  $X$ , if  $p_s(X) < p_i(X)$ , it means that the probability of  $X$  be a superior individual is less than it be a inferior individual, so discard the individual and sampling from  $P_{superior}(g)$  again.  $p_s(X)$  and  $p_i(X)$  are calculated by Eq.(8):

$$\begin{cases} p_s(X) = \prod_{i=1}^N [p_{superior}(x_i) * x_i + (1 - p_{superior}(x_i)) * (1 - x_i)] \\ p_i(X) = \prod_{i=1}^N [p_{inferior}(x_i) * x_i + (1 - p_{inferior}(x_i)) * (1 - x_i)] \end{cases} \quad (8)$$

### 3.3 Estimation of Distribution Algorithm Using Variety of Information

The algorithm employed operator A and operator B based on basicEDA is named as VI-EDA, it is described as follows:

#### Algorithm :VI-EDA

##### Begin

1. Generate initial population  $pop(0)$  with  $P(0)=(0.5, \dots, 0.5)$ , let  $g=1$ .

**While**("stopping criterion is not met")

2.1. Update  $P_{superior}(g)$  by Eq.(3) described in operator A.

2.2. Update  $P_{inferior}(g)$  by Eq.(5) or Eq.(6) described in operator B.

2.3. Generate  $pop^1(g)$ : generate new individual  $X_i$  by sampling from  $P_{superior}(g)$  and filter by  $P_{inferior}(g)$ .

2.3.  $pop^2(g) = pop(g-1) \cup pop^1(g)$ .

2.4. Generate  $pop(g)$  by select  $M$  superior individuals from  $pop^2(g)$ .

2.5  $g++$ .

##### End while

3. Output result.

##### End

In step 2.2,  $P_{inferior}(g)$  can be built by two methods, if by Eq.(5), name it VI-EDA1; if by Eq.(6), name it VI-EDA2. The two algorithms will be compared in Section 4.

## 4. EXPERIMENTAL STUDY

### 4.1 Test Problem

Knapsack problem is typical NP hard problem. Many practical problems can be modeled as a knapsack problem, such as capital budget, cargo loading, pollution prevention and control etc. Research on the method to solve the problem is important both in theory and in application. So the algorithm is simulated and tested by knapsack problem in this paper.

The knapsack problem is described as follows:

The maximum weight that a knapsack could bear is  $C$ , there are  $N$  items, the weight of them represented by  $W = (w_1, w_2, \dots, w_N)$ , the value of them represented by  $V = (v_1, v_2, \dots, v_N)$ . Select several items from the  $N$  items to make the total value of items in the knapsack is maximum.

The mathematical models of knapsack problem as shown in Eq.(9):

$$\begin{cases} \max f(X) = \sum_{i=1}^N v_i x_i \\ s.t. \sum_{i=1}^N w_i x_i \leq C \\ x_i \in \{0, 1\} \end{cases} \quad (9)$$

$x_i = 1$  means that the  $i$ th item is put into the knapsack,

$x_i = 0$  means that the  $i$ th item is not put into the knapsack.

Verify the algorithm by two widely used knapsack problem examples [13] as shown below, in which Example 1 is a well-known example used to test the performance of the algorithm and is more difficult than Example 2.

Example 1:  $C=1000, N=50$

$V=\{220, 208, 198, 192, 180, 180, 165, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105, 101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 20, 15, 10, 8, 5, 3, 1\};$

$W=\{80, 82, 85, 70, 72, 70, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 25, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1\};$

Example 2:  $C=6718, N=100$

$V=\{597, 596, 593, 586, 581, 568, 567, 560, 549, 548, 547, 529, 529, 527, 520, 491, 482, 478, 475, 475, 466, 462, 459, 458, 454, 451, 449, 443, 442, 421, 410, 409, 395, 394, 390, 377, 375, 366, 361, 347, 334, 322, 315, 313, 311, 309, 296, 295, 294, 289, 285, 279, 277, 276, 272, 248, 246, 245, 238, 237, 232, 231, 230, 225, 192, 184, 183, 176, 174, 171, 169, 165, 165, 154, 153, 150, 149, 147, 143, 140, 138, 134, 132, 127, 124, 123, 114, 111, 104, 89, 74, 63, 62, 58, 55, 48, 27, 22, 12, 6\};$

$W=\{54, 183, 106, 82, 30, 58, 71, 166, 117, 190, 90, 191, 205, 128, 110, 89, 63, 6, 140, 86, 30, 91, 156, 31, 70, 199, 142, 98, 178, 16, 140, 31, 24, 197, 101, 73, 169, 73, 92, 159, 71, 102, 144, 151, 27, 131, 209, 164, 177, 177, 129, 146, 17, 53, 164, 146, 43, 170, 180, 171, 130, 183, 5, 113, 207, 57, 13, 163, 20, 63, 12, 24, 9, 42, 6, 109, 170, 108, 46, 69, 43, 175, 81, 5, 34, 146, 148, 114, 160, 174, 156, 82, 47, 126, 102, 83, 58, 34, 21, 14\};$

The simulation is performed based on the above two examples to study:

- 1) Determination of  $\lambda$ .
- 2) Determination of  $\alpha$ .
- 3) The impact of operator A on probability model.
- 4) The impact of operator A and operator B on performance of the algorithm.

As compare, results of some literatures listed in Table 1, Ref [13] adopt genetic algorithm with schema replaced (MGA) solving KP, Ref [14] adopt good point set based genetic algorithm (JGA), Ref [15] adopt a heuristics algorithm based on tabu search (TBA) solving Example 1; Greedy algorithm is a commonly used algorithm solving KP, the results of greedy algorithm also listed. “\*\*” means that the algorithm did not solve the Example.

**Table 1 Results in literatures**

Algorithm	Exa1	Exa2
Greedy algorithm	3095	26380
MGA	3103	26559
JGA	3103	**
TBA	3103	**

Since the literatures obtained the optimal value of Example 1 is 3103, the optimal value of Example 2 is 26559, therefore the simulation results is compared with 3103、26559.

Each case independently run 30 times. Sn is defined as the times of the simulation results is not worse than the known optimal value in 30 runs.

## 4.2 Determine the Parameters

In this paper,  $M=200$ , termination criterion is achieve maximum number of iterations  $G$ ,  $G = 1000$ .

### 4.2.1 Determination of $\lambda$

In Ref [10], the best half solutions of the population were used to build the probability model and generate new individuals, the worst half solutions of the population were used to build the probability model and filter the generated new individuals, i.e.  $\lambda = 50\%$  in Eq.(1) and Eq.(5). Because in prior study, the result of basicEDA is the best when  $\lambda = 25\%$  in Eq.(1), so let  $\lambda = 25\%$  when employ method 1 to build  $P_{inferior}(g)$ . Named the algorithm when  $\lambda = 25\%$  as 25%-EDA, named the algorithm when  $\lambda = 50\%$  as 50%-EDA. 25%-EDA and 50%-EDA are based on basicEDA, for consistent with Ref [10], information of former probability model is not used in 25%-EDA and 50%-EDA, the information of inferior individuals is used only.

The main difference between the three algorithms mentioned above described as followings:

**basicEDA:** select the best  $25\% * M$  individuals of population to build probability model  $P(g)$  and sampling from it to generate population.

**50%-EDA、25%-EDA:** select the best  $\lambda * M$  individuals of population to build probability model  $P_{superior}(g)$ , select the worst  $\lambda * M$  individuals of population to build probability model  $P_{inferior}(g)$  to filter the individuals generated by  $P_{superior}(g)$ ,  $\lambda$  is 50 % and 25% respectively for 50%-EDA and 25%-EDA.

The results of 50%-EDA and 25%-EDA are compared, as shown in Table 2. The results of basicEDA are shown in Table 2 as reference. In the table, Exa represents Example, Ave.value represents the average value of 30 runs, Ave.iter represents the average evolution iteration of 30 runs.

**Table 2 Results under different  $\lambda$**

Exa	Algorithm	Best value	Ave. value	Worst value	Sn	Ave. iter
1	basicEDA	3114	3080.7	3059	2	26.0
	25%-EDA	3118	3089.5	3055	6	27.1
	50%-EDA	3101	3077.9	3056	0	60.6
2	basicEDA	26559	26531.5	26492	5	21.0
	25%-EDA	26559	26541.5	26501	11	24.0
	50%-EDA	26559	26548.8	26536	13	27.6

As can be seen from the data shown in Table 2:

For Example 1, the best value of 25%-EDA is 3118, Sn is 6, the optimization result is the best in the three algorithms ; the optimization result of 50%-EDA is the worst, even worse than basicEDA, the best value is less than the known optimal value 3103.

For Example 2, the result of 50%-EDA is better than the result of 25%-EDA, Sn is 13 and 11 respectively, average value is 26548.8 and 26541.5, but the difference of them is not obvious.

So  $\lambda=25\%$  is used in this paper.

Furthermore, the optimization results of 25%-EDA are unsatisfactory, although it improved basicEDA. This shows that the improvement is limited if use information of inferior individuals only.

#### 4.2.2 Determination of $\alpha$

Simulation is performed for VI-EDA1, in which

$P_{inferior}(g)$  is built through method 1, when  $\alpha$  takes 0.1, 0.2, 0.4, 0.6, 0.8 respectively, results are shown in Table 3.

As can be seen from the data shown in Table 3, when  $\alpha$  is 0.1, the results of two examples are the best. For Example 1, the results of 30 runs are all better than 3103, the best value obtained is 3119, the average value of 30 runs is 3117.3, the worst value is 3108; For Example 2, the results of 30 runs are all 26559. With the increasing of  $\alpha$ , the average value, the worst value, Sn of two examples are all becoming worse.

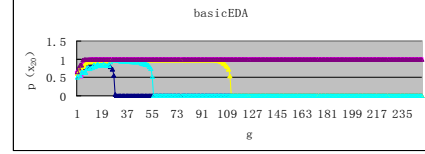
Average evolution iteration increases with  $\alpha$  decreases, but it exchanges the better optimization result. This is because the smaller the  $\alpha$ , the greater the impact on current generation probability model of last generation probability model, the slower the update speed of probability model, thus diversity solutions can be generated and better result can be got. The larger the  $\alpha$ , the faster the update speed of probability model, the more prone to premature convergence. So,  $\alpha$  takes 0.1 in the following simulations.

### 4.3 The Impact of Operator A and Operator B

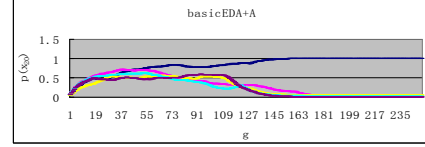
#### 4.3.1 The Impact on Probability Model of Operator A

The impact on probability model of operation A are shown in Figure.2. The algorithm employed operator A only on the basis of basicEDA is named as basicEDA+A. Randomly select gene  $x_{20}$  of Example1,  $x_{50}$  of Example 2 to observe

impact on the change trend of  $p(x_{20})$ 、 $p(x_{50})$  with increased generation. Figure. 2(a) shown the change trend of  $p(x_{20})$  of Example 1 under basicEDA and basicEDA+A. Figure.2(b) shown the change trend of  $p(x_{50})$  of Example 2 under basicEDA and basicEDA+A. Each case was ran five times.

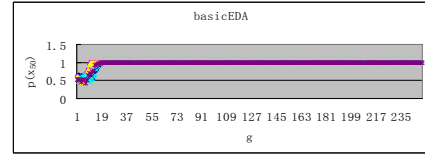


(a)-1 basicEDA

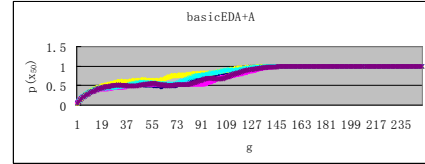


(a)-2 basicEDA+A

(a) Example1



(b)-1 basicEDA



(b)-2 basicEDA+A

(b) Example2

Figure 2 : Change trend of probability

Once  $p(x_i) = 0$ ,  $x_i$  will be 0 constant;  $p(x_i) = 1$ ,  $x_i$  will be 1 constant. In Figure 2 (a)-1 and Figure 2 (b)-1,  $p(x_{20})$  and  $p(x_{50})$  quickly converged to 0 or 1 under basicEDA, this means that the value of  $x_{20}$  and  $x_{50}$  will be fixed early, so lead to the decrease of combination of N genes, namely, the diversity of sampled solutions decreased, eventually lead to premature convergence. After employed operator A on the basis

Table 3 Results under different  $\alpha$

$\alpha$	Best value		Ave. value		Worst value		Sn		Ave.iter	
	Exa1	Exa2	Exa1	Exa2	Exa1	Exa2	Exa1	Exa2	Exa1	Exa2
0.1	3119	26559	3117.3	26559	3108	26559	30	30	117.7	145.6
0.2	3119	26559	3111.6	26556.1	3095	26536	24	24	84.3	93.9
0.4	3118	26559	3103.3	26555.8	3086	26536	14	23	60.5	50.2
0.6	3118	26559	3098.3	26550.3	3083	26511	10	18	58.5	39.3
0.8	3118	26559	3094.1	26547.5	3070	26512	8	14	57.9	34.3

of basicEDA , the results are shown in Figure 2 (a)-2 and Figure 2 (b)-2, as can be seen from them, the convergence speed of  $p(x_{20})$  、  $p(x_{50})$  is slowed down significantly compared with Figure.2 (a)-1 and Figure.2 (b)-1, thus the value of  $x_{20}$  and  $x_{50}$  will not be fixed early, so increasing the diversity of solutions, eventually got better optimization results. Therefore, the application of informaton of former probability model slowed down the update speed of probability model and improved the loss of diversity.

#### 4.3.2 The Impact of Operator A and Operator B on Performance of the Algorithm

Simulation is performed for basicEDA 、 the algorithm employed operator A only on the basis of basicEDA ( i.e.basicEDA+A)、 the algorithm employed both operator A and operator B (i.e.VI-EDA1 and VI-EDA2) to study the impact of two kinds of information . The simulation results are shown in Table 4.

Compare the results of basicEDA and basicEDA+A. For Example 1, after employed operator A, the best value increased from 3114 to 3119, average value increased from 3080.7 to 3115.7, the worst value increased from 3059 to 3098, Sn increased from 2 to 29 . For Example 2, after employed operator A, Sn increased from 5 to 29 , average value increased from 26531.5 to 26558, the worst value increased from 26492 to 26547. So the operator A improves the algorithm performance greatly. It further validates the conclusion of Section 4.3.1. For the two examples, the average evolution iteration of basicEDA is less than the average evolution iteration after employed operator A, This is because basicEDA continuously selects the best part of current population to building the probability model leading to the premature convergence.

Compare the results of the algorithm basicEDA+A and results of the algorithm VI-EAD:

a). When  $P_{\text{inferior}}(g)$  is built by method 1 in operator B (i.e.VI-EAD1): For Example 1, the best value obtained by basicEDA+A and VI-EAD1 are the same, the average value increased from 3115.3 to 3117.3, the worst value increased from 3098 to 3108,Sn increased from 29 to 30 after employed operator B; For Example 2, the best value obtained by two algorithms are the same, the average value increased from 26558 to 26559, the

worst value increased from 26547 to 26559,Sn increased from 27 to 30 after employed operator B. So the operator B with method 1 has further improved the performance of the algorithm use operator A only.

b). When  $P_{\text{inferior}}(g)$  is built by method 2 in operator B (i.e.VI-EAD2) : there is no significant difference in the optimization results between basicEDA+A and VI-EAD2, only the average evolution iteration is different, for Example 1, the average evolution iteration decreased from 130.7 to 113.1 after employed operator B; For Example 2, the average evolution iteration decreased from 130.9 to 125.9 after employed operator B, so the operator B with method 2 accelerated the search speed .

## 5. CONCLUSION

The algorithm proposed utilize the the former information of superior probability model when building superior probability model in EDA, and utilize the information of inferior solutions of population to filter the individuals generated by sampling from superior probability model,and validate the efficiency of proposed algorithm by simulation on knapsack problem.

The use of former information of superior probability model improves the loss of population diversity and prevents premature convergence, improves the performance of basicEDA greatly; the use of the information of inferior solutions further improves the performance of algorithm. when  $P_{\text{inferior}}(g)$  is built without former information of inferior probability model, operator B improved optimization ability of the basicEDA+A to some extend; when  $P_{\text{inferior}}(g)$  is built with former information of probability model, there is no significant improvement for the optimization results of basicEDA+A , but accelerates the convergence speed..

## 6. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61271143 and No.60871080

## 7. REFERENCES

- [1] LuisMartí · JesúsGarcía · AntonioBerlanga · JoséM.Molina .Multi-objectiveoptimization with an adaptive resonance theory-based estimation Of distribution algorithm. Ann Math Artif Intell.2013,68:247–273.
- [2] Shih-HsinChen ,Min-ChihChen ,Pei-ChannChang,QingfuZhang,Yuh-MinChen .Guidelines for

**Table 4 The impact of operator Aand operator B**

Example	Algorithm	Best value	Ave.value	Worst value	Sn	Ave.iter
1	basicEDA	3114	3080.7	3059	2	26.0
	basicEDA+A	3119	3115.3	3098	29	130.7
	VI-EDA1	3119	3117.3	3108	30	117.7
	VI-EDA2	3119	3116.7	3099	29	113.1
2	basicEDA	26559	26531.5	26492	5	21.0
	basicEDA+A	26559	26558	26547	27	130.9
	VI-EDA1	26559	26559	26559	30	145.6
	VI-EDA2	26559	26558	26547	27	125.9

- developing effective Estimation of distribution algorithms in solving single machine scheduling problems[J]. Expert Systems with Applications , 2010,37(9):6441-6451.
- [3] Zhang,Q.,Muhlenbein,H.On the convergence of a class of estimation of distribution algorithms[J].IEEE Transactions on Evolutionary Computation, 2004,8(2):127-136
- [4] Ceberio,J.E.,Irurozki,A.M.,Lozano,J..A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems[J]. Progress in Artificial Intelligence , 2012,1(1):103-117.
- [5] Roberto Santana, Pedro Larrañaga, José A. Lozano. Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain Placement problem[J]. Journal of Heuristics, 2008,14(5):519-547
- [6] Ling Wang, Sheng-yao Wang, Ye Xu. An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem[J]. Expert Systems with Applications 2012, 39: 5593-5599
- [7] Christopher Expósito Izquierdo, José Luis González Velarde, Belén Melián-Batista ,et al.Hybrid Estimation of distribution algorithm for the quay crane scheduling problem[J]. Applied Soft Computing , 2013,13: 4063-4076
- [8] Santiago Muelas, Alexander Mendiburu, Antonio LaTorre et al., Distributed Estimation of Distribution Algorithms for continuous optimization: How does the exchanged information influence their behavior?. Inform. Sci. (2013), <http://dx.doi.org/10.1016/j.ins.2013.10.026>
- [9] Zhongping Wan, Lijun Mao, Guangmin Wang. Estimation of distribution algorithm for a class of nonlinear bilevel programming problems. Information Sciences ,2014, 256:184-196
- [10] Yi Hong,Guopu Zhu,Sam Kwong et .al.Estimation of distribution algorithms making use of both high quality and low quality individuals[c].FUZZ-IEEE 2009,Korea,August 20-24,2009
- [11] Mark Hauschild,Martin Pelikan. An introduction and survey of estimation of distribution algorithms[J],Swarm and Evolutionary Computation , 2011,1(3):111-128
- [12] Jürgen Branke , Clemens Lode , Jonathan L. Shapiro.Addressing sampling errors and diversity loss in UMDA[C]. Proceedings of the 9th annual conference on Genetic and evolutionary computation, London,England,United Kingdom,2007:508-515
- [13] LI Kang-shun , JIA Yu-zhen , ZHANG Wen-sheng . Genetic algorithm with schema replaced for solving 0/1 knapsack problem[J] .App lication Research of Computers.2009,26(2):470-471,505.
- [14] ZHANG Ling, ZHAN G Bo. Good Point Set Based Genetic Algorithm[J].CHINESE J J1COM PUTERS.2001,24 (9) : 917-922
- [15] Heuristics Algorithm for Knapsack Problem Based on the Tabu Search. Journal of the University of Electronic Science and Technology of China, 2005,34(3):359-362