

# Research and New Development of Application of the Estimation of Distribution Algorithms

Dandan Wang, Yong Jiang, Jianliang LI  
Department of Information and Computing Science  
Nanjing University of Science and Technology  
NanJing, 210094, P. R. China  
e-mail: math@mail.njust.edu.cn

**Abstract**—The Estimation of Distribution Algorithms (EDAs) is a novel class of evolutionary algorithms which is motivated by the idea of building probabilistic graphical model of promising solutions to represent linkage information between variables in chromosome. Through learning of and sampling from probabilistic graphical model, new population is generated and optimization procedure is repeated until the stopping criteria are met. In this paper, the mechanism of the Estimation of Distribution Algorithms is analyzed. Currently existing EDAs are surveyed and categorized according to the probabilistic model they used.

**Keywords**—genetic algorithm; evolutionary programming; estimation of distribution algorithms

## I. Introduction

Estimation of Distribution Algorithm is a new kind of intelligent optimization algorithm developed on the basis of a genetic algorithm. From the principle overcome the learning problems in the traditional genetic algorithm are difficult to resolve in the chain. And avoid the building block destruction of genetic algorithm crossover and mutation operation, decrease the parameter setting and blindness and randomness of the genetic operator to choose, a higher theoretical basis and evolution-oriented.

## II. Evolutionary Algorithms

Evolutionary Algorithms -EAs- (Back et al., 1997) are stochastic search techniques designed as an attempt to solve adaptive and hard optimization tasks on computers. In fact, it is possible to find this kind of algorithms applied for solving complex problems like constrained optimization tasks, problems with a noisy objective function, or problems having high epistasis and multimodality. Evolutionary algorithms include genetic algorithms, evolutionary strategy, evolutionary programming.

## III. Genetic algorithms

Genetic algorithm (GA) is a highly parallel and adaptive global optimization random search algorithm which references biological natural genetic mechanisms. It

has strong function, good robust, simple calculation, infinite the search space and other characteristic. However, genetic algorithm itself has still some problems. First, genetic algorithm's key is to deal with the building blocks of evolutionary process, but cross operator and mutation operator does not have learning and identify linkage capacity between gene, so the actual recombination operation usually works the destruction of the building blocks, resulting in algorithm approximation for the local optimal solution or premature; in addition, the genetic algorithm of operate parameters has strong dependent, and even parameters selection itself is a optimization problem.

## IV. Evolutionary Programming

Evolutionary Programming (EP) is introduced in the 1960s by the U.S. L.J. Fogel, and in order to solve the problem of a finite machine evolutionary model, L.J. Fogel borrowing, such as the evolution idea on the finite state machines to evolution for a more Good finite state machines, they applied this method to data diagnosis 、 pattern recognition and classification and control system design and so on, and has achieved good results. Classical EP (CEP) using Gaussian mutation, according to the adaptive capacity of its variation is divided into two categories, with the capacity of adaptive mutation CEP better.

Adaptive CEP can be described as follows:

1). Initialized population. Produce  $L$  individuals, and the  $t = 1$ . Each individual Expressed as  $\{\alpha_i, \eta_i\}$ ,  $i = 1, \dots, L$ ,  $\alpha_i$  is the target variable,  $\eta_i$  is the standard deviation of Gaussian mutation, subscript is the No.  $i$  individual.

2). Fitness evaluation. Fitness function is  $f(\alpha_i)$ , calculate fitness

Of each individual  $\{\alpha_i, \eta_i\}$ .

3). Mutation. Each parent individual  $\{\alpha_i, \eta_i\}$  to produce a progeny individual  $\{\alpha'_i, \eta'_i\}$ , resulting in the following manner:  $\{\alpha'_i, \eta'_i\}(j) = \alpha_i(j) + \eta_i(j)N_j(0,1)$

$\{\alpha'_i, \eta'_i\}(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_j(0,1))$

Which  $\alpha_i(j), \alpha'_i(j), \eta_i(j), \eta'_i(j)$  are the first

j-dimensional vector component

respectively;  $N(0,1)$  is random number with zero mean and standard deviation of normal distribution of 1,  $N_j(\mathbf{0}, \mathbf{1})$  is a random number generated by component  $j$ ; factor  $\tau', \tau$  are  $(\sqrt{2k})^{-1}, (\sqrt{2\sqrt{k}})^{-1}$  respectively,,  $k$  is the vector dimension.

- 4). Calculation fitness of each individual of offspring,  $i=1, \dots, L$ .
- 5). Comparison. Parent individual and the corresponding individual offspring do 1-to-1 comparison, performance good as the winner.
- 6). Selection. From the parent and offspring individuals to choose the  $L$  winner individuals as the parent of the next cycle.
- 7). If the number of iterations to meet the criteria or requirements, then stop algorithm; Otherwise, the  $t = t + 1$ , go to Step 3.

## V. Estimation of Distribution Algorithms

In order to solve these problems of genetic algorithms, the improved method is to alter the basic principles of recombination operation, changes cross and mutation operation of genetic algorithm to learning the probability distribution of good solutions, the basic idea is to Select some excellent solutions from the current population, and use these excellent solutions to estimate and learning distribution model, and then sampling the distribution model to produce a new population. Take turns iteration, finally approaching the optimal solution. Based on this distribution model the algorithm is called Estimation of Distribution Algorithm. It can be described as follows:

$D_0 \leftarrow$  Generated  $M$  individuals as the initial population randomly.

Repeat for  $l = 1, 2, \dots$  until the termination of the guidelines to achieve.

$D_{l-1}^{Se} \leftarrow$  According to selection to choose  $N \leq M$  individuals from  $D_{l-1}$  as the advantaged groups.

$p_l(x) = p(x | D_{l-1}^{Se}) \leftarrow$  Estimated the joint probability distribution.

$D_l \leftarrow$  Sampling  $M$  times From  $p_l(x)$ , get a new groups.

According to the model order of complexity, EDAs divides into three kinds:

- (1) Without dependencies
- (2) Bivariate dependencies
- (3) Multiple dependencies.

### A. EDA approaches to combinational optimization

- Without dependencies  
It has three algorithms:  
(1) Univariate Marginal Distribution Algorithm UMDA

$$p_l(x) = p(x | D_{l-1}^{Se}) = \prod_{i=1}^n p_l(x_i) = \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i = x_i | D_{l-1}^{Se})}{N}$$

### (2) Population Based Incremental Learning, PBIL

$$p_{l+1}(x) = (1 + \alpha) p_l(x) + \alpha \frac{1}{N} \sum_{k=1}^N x_{k:M}$$

### (3) compact Genetic Algorithm, cGA

- Bivariate dependencies

It has three algorithms:

(1) Mutual Information Maximization for Input Clustering, MIMIC

$$p_l^\pi(x) = p_l(x_{i1} | x_{i2}) \cdot p_l(x_{i2} | x_{i3}) \cdots p_l(x_{in-1} | x_{in}) \cdot p_l(x_{in})$$

(2) Combining Optimizers with Mutual Information Trees COMIT

$$p_l(x) = p(x | D_{l-1}^{Se}) = \prod_{i=1}^n p_l(x_i | x_{j(i)})$$

### (3) Bivariate Marginal Distribution Algorithm, BMDA.

$$p_l(x) = \prod_{x_r \in R_l} p_l(x_r) \prod_{x_s \in I \cup R_l} p_l(x_s | x_{j(s)})$$

MIMIC, COMIT and BMDA all assume interdependencies between the variables from the database of selected individuals, they adopt chain, tree and forest structure.

- Multiple dependencies

It has six algorithms:

### (1) Extended compact Genetic Algorithm, EcGA

$$p_l(x) = p(x | D_{l-1}^{Se}) = \prod_{c \in C_l} p_l(x_c | D_{l-1}^{Se})$$

$$\min -N \sum_{c \in C_l} \sum_{x_c} p(X_c = x_c) \log p(X_c = x_c) + \log N \sum_{c \in C_l} \dim X_c$$

### (2) Factorized Distribution Algorithm FDA

$$p_l(x) = \prod_{i=1}^k p_l(x_{b_i} | x_{c_i})$$

### (3) Polytrees Approximation of Distribution Algorithms PADA

### (4) Estimation of Bayesian network Algorithm EBNA

### (5) Bayesian Optimization Algorithm, BOA

$$p(X) = \prod_{i=1}^n p(X_i | \prod_{x_i} x_i)$$

### (6) Mixture models

$$f_N(x; \mu, \Sigma) = \prod_{i=1}^n f_N(x_i; \mu_i, \sigma_i^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{x_i - \mu_i}{\sigma_i})^2}$$

### B. EDA approaches in continuous domains

- Without dependencies

It has three algorithms:

(1) The Univariate Marginal Algorithm for continuous domains UMDAc

$$f_l(x; \theta') = \prod_{i=1}^n f_l(x_i, \hat{\theta}_i')$$

(2) Stochastic Hill Climbing with Learning by Vector of Normal Distributions SHCLVND

(3)an extension of Boolean PBIL algorithm to continuous spaces PBILc

- Bivariate dependencies  $MIMIC_c^G$
- Multiple dependencies

It has six algorithms:

- (1)Estimation of multivariate normal algorithm  $EMNA_{global}$
- (2)Estimation of multivariate normal algorithm adaptive  $EMNA_a$
- (3)Estimation of multivariate normal algorithm incremental  $EMNA_i$
- (4)Estimation of Gaussian Network Algorithm, EGNA Including  $EGNA_{ee}$ 、 $EGNA_{BGe}$ 、 $EGNA_{BIC}$
- (5)Iterated Density Evolutionary Algorithms IDIA
- (6)Mixture models

## VI. The new algorithm idea

The intelligent mixture algorithm is the research hot spot of current intelligent optimization algorithm, may fuse many kinds of optimized algorithms' superiority, and enhances the algorithm the performance. Other intelligent method and Estimate of Distribution Algorithm's mixture will enhance search ability of the algorithm, better balanced astringency of algorithm and community multiplicity. I proposed the combination between Evolution Programming and Estimate of Distribution Algorithm.

We also note that the optimization of the large data, there are many local optimum point, when the EP adopted to control the division nearest neighbor (that is, in the nearest neighbor division, the introduction of control factor, change the distance metric weights to achieve different division results. Plot out control factor by the Evolution Programming, then the coding space is relatively small.), its speed of convergence relatively slow. Based on probability and statistics idea of the EDA is a better method to make up. EDA can use the EP in the evolutionary process to produce populations ,then use the probability distribution estimated to predict the optimal individual, provide search direction for the EP and accelerate their convergence.

Evolutionary Programming (EP) has evolutionary trend, but its lack is that convergence is relatively slow. the classic Evolutionary Programming (CEP) using Gaussian variation, Gaussian variation suitable for the small neighboring search Space, that is, Gaussian mutation operator to provide a relatively small search step. When the EP runs into a local optimal individual, but the global optimal individual apart from the local optimal individual, if want to get away from the local optimal individual, a small step may not be more effective, although the Adapt of Evolutionary Programming provide the adaptive mutation mechanism for the EP.

EDA requires a considerable population size, may also need to improve the probability model, to abandon a univariate Gaussian distribution assumptions, these two points is not easy to do in the optimization of the data. EDA

through the estimate of probability distribution to guide the search, in order to get a more precise estimate,the population size requirements higher. In addition, in the case of having many local optimal individuals ,according to fitness and selection mechanism, good individuals can be form some groups, each group contains a local optimal individual, then, if under the assumption that a single Gaussian distribution, estimates the parents, the estimated results (offspring individual) will stay away from the o real global optimal individual. That is to say in this wrong estimate, the new population will fall into the wrong search space, leading to a convergence slowly, even will convergence to a local optimum.

EP and the EDA have their own advantages and disadvantages, can be a combination of both, have a valid design algorithm, the following analysis gives a feasible scheme. First, the algorithm used the EP to find some local optimal individuals, and the population size can be relatively small, and assuming that these local optimal individuals surround the global optimal individual, then, introduce EDA, use the advantages of probability and statistics, for the EP produce some offspring individuals, approaching the global optimal individual. To simplify the algorithm, and taking into account the good individuals concerned with EDA estimated are not large, we assume that this part obey a single Gaussian distribution. Finally, test the estimated results, if the performance of the estimated results is good, and then retain it and enter the next generation, accelerate the convergence of EP. If the choice of this part of the distribution of the individual is not Gaussian distribution, the results will provide a wrong Search direction for the algorithm, then algorithm under the control of the EP, through adaptive mechanisms, also fled out from the wrong search direction, and because only part of the individuals to participate in EDA operation, risk with less.

EDA operation can be described as follows:

- 1) set up a certain proportion of the optimal number of individual  $M_1 = M \cdot P_L$ ,  $0 < P_L < 1$ ;
  - 2) According to the value of the father of individual to sort
  - 3) Get the means of the first  $M_1$  good individual of parent individual, recorded as EDA D;
  - 4) Produce a random number, if the random number is greater than the EDA set probability  $P_{EDA}$ , will be used EDA D for EP variation operation, and produce a probability estimate "best" of the individual offspring.
- EDA-EP realizes the step description to be as follows:
- 1)Generate a random initialization individual,  $t = 0$ , initialization individual, that is nearest neighbor of control factor a and the variation step b.
  - 2)Updating of the individual, and calculate the individual fitness.
  - 3)According to the father of the individual to sort of size (corresponding to change a, b in the order), directly retain a

certain percentage of good individual; may be appropriate to introduce cross operator.

4) Transfer the EDA operation. Use the percentage of good individual of parents to estimate the individuals' best individual value.

5) Mutation. Mutant the parents which are selected and crossed, then produce their own offspring individuals, some of which according to certain probability to generate the offspring individuals.

6) Calculate the fitness of offspring individual.

7) Compare parents with offspring individuals one-to-one, the individual of good performance for the winner, as the parent cycle into the next round .

8) If achieves the optimal solution , stop the algorithm ; Otherwise,  $t = t + 1$ , go to Step 3.

## REFERENCES

- [1] Hong Wei Huo, Jin Xu and Zheng Bao, "An algorithm of job-scheduling problem based on genetic algorithm," Systems [1] *Larrñaga*, P., Etxeberria, R., Lozano, J. A., *Pena*, J. M., 1999, Optimization by learning and simulation of Bayesian and Gaussian networks, Technical Report KZZA-IK-4-99, Dept. of Computer Science and Artificial Intelligence, University of the Basque Country.
- [2] *Larrñaga*, P., Lozano, J. A., 2002, Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Kluwer Academic Publishers.
- [3] Maxwell, B., Anderson, S., 1999, Training hidden Markov models using population-based learning, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO), W. Banzhaf et al., eds., Morgan Kaufmann, Orlando, Florida, USA, pp. 944.
- [4] Baluja S. Population-based incremental learning : A method for integrating genetic searching based function optimization and competing learning. Carnegie Mellon University, CMUTech. Rep. CMU. CS. 94. 163 : 1994.
- [5] Back T. Evolutionary Algorithms in Theory and Practice [M]. London : Oxford University Press , 1996.
- [6] Syswerda G. Simulated Crossover in Genetic Algorithms [M] . San Mateo , CA : Morgan Kaufmann , 1993.
- [7] Harik G, Goldberg D E. Linkage learning [ A ] . In : Belew R. Foundations of Genetic Algorithms [C] . San Mateo , CA : Morgan Kaufmann , 1996.