Genetic Drift in Univariate Marginal Distribution Algorithm

Yi Hong
Dept. of Computer Sci. and Eng.
Shanghai Jiaotong University
Huashan Road
86-21-62933891
goodji@sjtu.edu.cn

Qingsheng Ren
Dept. of Computer Sci. and Eng.
Shanghai Jiaotong University
Huashan Road
86-21-62932089
ren-qs@cs.sjtu.edu.cn

Jin Zeng
Dept. of Math.
Shanghai Jiaotong University
Huashan Road
86-21-54742403
zengjin@sjtu.edu.cn

ABSTRACT

Like Darwinian-type genetic algorithms, there also exists genetic drift in Univariate Marginal Distribution Algorithm (UMDA). Since the universal analysis of genetic drift in UMDA is very difficult, in this paper, we just approach a certain kind of problem (WOneMax Problem). For WOneMax Problem, The individual space in UMDA can be denoted as a full binary tree, and the selecting process in UMDA can be considered as a process of cutting branch. We employ this binary tree to calculate the probability change of each variable between two adjacent generations. Comparing this change with our experimental data, we find that when the population size is limited, there exists genetic drift in UMDA. In order to avoid genetic drift, we model the probability of each variable as a signal with noise, and then use smoothing filter to eliminate genetic drift. Numerical results show this method is effective.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization– *global optimization, unconstraint optimization.*

General Terms

Theory, Algorithms.

Keywords

Estimation of distribution algorithms, Genetic drift, Univariate marginal distribution algorithm.

1. GENETIC DRIFT IN UMDA

WOneMax problem is defined as:

$$\max \ f(X) = \left(\sum_{i=1}^{n} w_i x_i \right) \quad x_i \in \{0,1\} \tag{1}$$

Where

$$X = \{x_1, x_2, \ldots, x_n\}, \ w_i > 0, \ w_i > \sum_{j=i+1}^{n} w_j$$

For WOneMax problem, the searching space can be modeled as a full binary tree. In this binary tree, all nodes in i^{th} layer denote the variable x_i. If X_i equals to 1, the individual belongs to the left branch; else belongs to the right branch. Leaves represent individuals in the whole population space. Suppose the population size equals to G , in the l^{th} generation, UMDA uses the probability $p_l(x)$ to generate the new population D . This process can be seen as partitioning G individuals form top to down. Define O_{ij} is the j^{th} node in i^{th} layer of the binary tree from left to right, $NUM(O_{ij})$ is the number of individual passing from O_{ij} . When the population size is infinite, we can calculate the number of individuals passing each node:

$$NUM(O_{ij}) \le G \prod_{k=1}^{i-1} \max\{ p_l(x_k), 1 - p_l(x_{ki}) \} \tag{2}$$

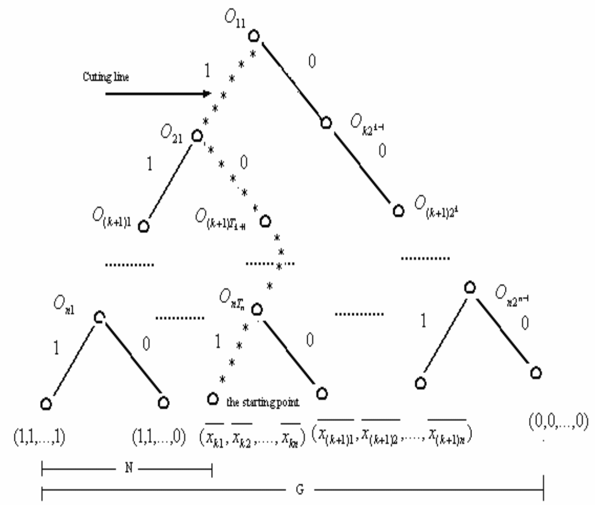


Figure 1. Cutting process

There are many effective selecting methods, here truncation selection is adopted, viz. selecting N best individuals to form the subpopulation, and use the frequency of variables in the subpopulation to calculate $p_l(x)$, $N = aG$. This process can be simulated as a process of cutting branch (see figure 1). Basing on the analyses of the cutting process, we can calculate the probability change between two adjacent generations.

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

This work was supported by National Nature Science Foundation of China (Nos. 60271033)

$$0 \le p_{l+1}(x_i) - p_l(x_i) \le \frac{(1-p_l(x_i))\prod_{j=1}^{i-1}\max\left\{p_l(x_j), 1-p_l(x_j)\right\}}{\alpha} \tag{3}$$

Since in UMDA $p_{l+1}(x_i) = p_l(x_i \,|\, D_l^{Se})$, so:

$$0 \le p_l(x_i \,|\, D_l^{Se}) - p_{l-1}(x_i \,|\, D_{l-1}^{Se}) \le \frac{(1-p_l(x_i))\prod_{j=1}^{i-1}\max\left\{p_l(x_j), 1-p_l(x_j)\right\}}{\alpha} \tag{4}$$

From (3) and (4), we can see that for most bits, the change of probability between two adjacent generations is very tiny. In experiments, we find that as the population size is limited, there are some great changes in probability that aren't consistent with (3) and (4). This is the phenomenon of genetic drift.

2. FRAME OF UMDA WITH SMOOTHING

To diminish genetic drift, we model the probability as a signal with noise, and then smooth the probability in two spaces.

$$S_1 : \left\{ {}^1p_l(x \,|\, {}^1D_l^{Se}), \dots, {}^kp_l(x \,|\, {}^kD_l^{Se}), \dots, {}^np_l(\ x \,|\, {}^nD_l^{Se}) \right\}$$

$$S_2 : \left\{ {}^kp_0(x \,|\, {}^kD_0^{Se}) \dots, {}^kp_j(x \,|\, {}^kD_j^{Se}), \dots, {}^kp_l(x \,|\, {}^kD_l^{Se}) \right\}$$

Parameters illustration:

${}^kD_l^{Se}$: Selected subpopulation in l^{th} generation at k^{th} experiment;

${}^kp_l(x \,|\, {}^kD_l^{Se})$: Probability of individual x in l^{th} generation in ${}^kD_l^{Se}$;

The probability of each variable is calculated as:

$F_{2\times 2}$:

$$p_{l+1}(x_j) = \frac{\begin{bmatrix} {}^1p_{l-1}(x_j \,|\, {}^1D_{l-1}^{Se}) & {}^2p_{l-1}(x_j \,|\, {}^2D_{l-1}^{Se}) \\ {}^1p_l(x_j \,|\, {}^1D_l^{Se}) & {}^2p_l(x_j \,|\, {}^2D_l^{Se}) \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}{4} \tag{5}$$

and $F_{3\times 3}$:

$$p_{l+1}(x_j) = \frac{\left\{ \begin{bmatrix} {}^1p_{l-2}(x_j \,|\, {}^1D_{l-2}^{Se}) & {}^2p_{l-2}(x_j \,|\, {}^2D_{l-2}^{Se}) & {}^3p_{l-2}(x_j \,|\, {}^3D_{l-2}^{Se}) \\ {}^1p_{l-1}(x_j \,|\, {}^1D_{l-1}^{Se}) & {}^2p_{l-1}(x_j \,|\, {}^2D_{l-1}^{Se}) & {}^3p_{l-1}(x_j \,|\, {}^3D_{l-1}^{Se}) \\ {}^1p_l(x_j \,|\, {}^1D_l^{Se}) & {}^2p_l(x_j \,|\, {}^2D_l^{Se}) & {}^3p_l(x_j \,|\, {}^3D_l^{Se}) \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right\}}{9} \tag{6}$$

$*$: Convolution operator.

${}^kp_l(x_i \,|\, {}^kD_l^{Se})$: Probability of variable x_i in l^{th} sample.

Frame of UMDA with smoothing filter (SUMDA(2)):

```
1)   Set $p_1(x_1, x_2, ..., x_n) = (0.5, 0.5, ..., 0.5)$
     ${}^kp_l(x_j \,|\, {}^kD_0^{Se}) = 0.5$,  $j = 1:n$;  $k = 1:2$ ;  $l = 1$ ;  $k = 1$ ;
2)   Use $p_l(x)$ to generate individuals $X_1^l, ..., X_G^l$ , ${}^kD_l$ is
     composed by these individuals;
3)   Evaluate and rank $X_1^l, X_2^l, ..., X_G^l$ ;
4)   Select $N(N < G)$ best individuals:;
5)   Compute ${}^kp_l(x_i \,|\, {}^kD_l^{Se})$ ;
6)   If $\mathrm{k} \le 2$,  $\mathrm{k} = \mathrm{k} + 1$, goto 2) ; else $\mathrm{k} = 1$
7)   Calculate $p_{l+1}(x_j)$ with (5):
8)    Update probability vector $p_{l+1}(x)$ ;
9)   If the termination criterion is met, finish;
10)   $l = l + 1$ , go to 2).
```

If using (6) to calculate $p_{l+1}(x_j)$, we call it SUMDA(3).

3. NUMERICAL RERULTS

Example 1. Find the minimal value of f_1 .

$$f_1(x) = \sum_{i=1}^{5n} 2^i x_i \quad x_i \in \{0,1\} .$$

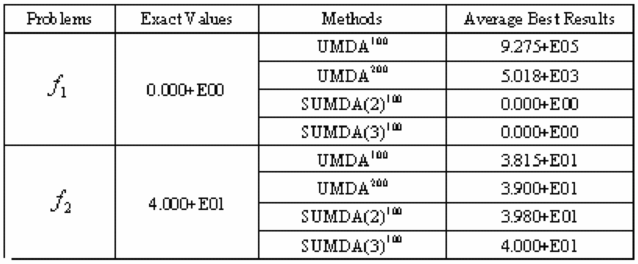
Example 2. Find the maximal value of f_2 .

$$F_{muhl}^5(x) = \begin{cases} 3.0 & for \quad x = (0,0,0,0,1) \\ 2.0 & for \quad x = (0,0,0,1,1) \\ 1.0 & for \quad x = (0,0,1,1,1) \\ 3.5 & for \quad x = (1,1,1,1,1) \\ 4.0 & for \quad x = (0,0,0,0,0) \\ 0.0 & otherwise \end{cases}$$

$$f_2(x) = \sum_{j=1}^{n} F_{muhl}^5(s_j)$$

Where $s_j = (x_{5j-4}, x_{5j-3}, x_{5j-2}, x_{5j-1}, x_{5j})$.

Table 1. Average of best results after 50 executions, n=10



Problems	Exact Values	Methods	Average Best Results
f_1	0.000+E00	UMDA^{100}	9.275+E05
		UMDA^{200}	5.018+E03
		SUMDA(2)^{100}	0.000+E00
		SUMDA(3)^{100}	0.000+E00
f_2	4.000+E01	UMDA^{100}	3.815+E01
		UMDA^{200}	3.900+E01
		SUMDA(2)^{100}	3.980+E01
		SUMDA(3)^{100}	4.000+E01

$\alpha = 0.30$

UMDA^{100}: UMDA whose population size equals to 100;

UMDA^{200}: UMDA whose population size equals to 200;

SUMDA(2)^{100}: SUMDA whose size equals to 100;

SUMDA(3)^{100}: SUMDA whose size equals to 100;