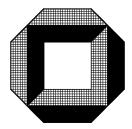**Studienarbeit**

Reducing diversity loss in estimation of distribution algorithms

Universität Karlsruhe (TH)
Fakultät für Informatik?
Institut für Angewandte Informatik und Formale
Beschreibungsverfahren

Verantw. Betreuer: Prof. Dr. H. Schmeck
Betr. Mitarbeiter: PD Dr. J. Branke

Beginn: 15.08.2006

Reducing diversity loss in estimation of distribution algorithms

Clemens Lode, Jürgen Branke, John Shapiro

Contents

Chapter 1

Reducing diversity loss in estimation of distribution algorithms

1.1 Introduction

Introduction:

With inappropriate settings, many EDAs can reach a state from which the probability of ever finding the optimum is zero. This is due to diversity loss which cannot be restored. If any component of the data vectors does not take one of its allowed values anywhere in the entire population, that value can never be restored. If that value is required in the optimum, the optimum will never be sampled.

The flat landscape is the simplest problem in which this can be studied. It was shown that this diversity loss is the same for a whole class of EDAs. A consequence of this is that for a problem which is almost everywhere flat, such as the Needle problem, the probability of diversity loss before the optimum is sampled is also universal for the class, and it was shown that using an exponentially large population size would avoid this.

In this paper I will look into a different approach to the problem of diversity loss

Abstract. Using the same general class of EDAs as [1] (probability model is build using only data sampled from the current probability model) and the result we come to the conclusion that by changing the distribution vector p accordingly to the population size n and the number of selected individuals $\tilde{n}$, we can correct the diversity loss. This is true for a flat landscape and tests have shown that this correction also works for non-flat problems.

1.2 Definitions

The diversity of a given population can be measured by the 'trace of the empirical co-variance matrix'. Let

1. C: number of components of each individual

2. n: number of individuals in the population

3. $\tilde{n}$: number of selected individuals

4. A: set of different values a component can take

5. $|A|$: number of different values a component can take

$$x_i^\mu: \text{ component i of individual } \mu \tag{1.1}$$

$$v_i^a = \frac{1}{n} \sum_{\mu=1}^{n} \varphi(x_i^\mu = a) \tag{1.2}$$

$$v = \frac{1}{|A|} \sum_{i=1}^{C} \sum_{a=0}^{|A|-1} v_i^a (1 - v_i^a) \tag{1.3}$$

Our goal is to determine the resulting diversity of a population created on a given distribution p, so we have to calculate all possible combinations of individuals in that populations of the size n. For simplicity we are looking at strings of the size 1, i.e. $C = 1$, in the case of UMDAs on a flat landscape the results for $C > 1$ are the same, see chapter X.

To create a population with the distribution p each individual has in its component with the probability p a '1' (or with the probability $(1 - p)$ a '0'). Creating a population with the size of n with $|A|$ different values in each component we then have a combination with repetition, i.e. there would be

$$\frac{(|A| + n - 1)!}{n!(|A| - 1)!} = \binom{|A| + n - 1}{n} \tag{1.4}$$

possible different population. For example for $|A| = 2$ there would be $n + 1$ different populations (...000, ...001, ...011, ...111, ...) as the position of an individual in the population is not important. In this paper I will only look into the case of $|A| = 2$, i.e. bitsrings. The main difference with $|A| > 2$ compared to $|A| = 2$ is that the probability for a certain population is different and that we no longer can identify a certain population with a 'k', with k being the number of '1's.

The probability for a certain population is

$$p_k = p^k (1-p)^{n-k} \binom{n}{k} \tag{1.5}$$

with k being the number of '1's.

We further define a v_k which represents the diversity for a population with k '1's. The v_i^a values of a v_k with a given k is then:

$$v_i^a(k) = \frac{1}{n} \sum_{\mu=1}^{n} \varphi(x_i^\mu(k) = a) \tag{1.6}$$

As we have defined k as the number of '1's and set $C = 1$ the sum over all $\varphi(x_i^\mu = 1)$ is k (and the sum over all $\varphi(x_i^\mu = 0)$ is $n - k$) and our v_1^a (we only need the v_1^as because we only have one component, i.e. $C = 1$) is:

$$v_1^0 = \frac{n-k}{n}$$

$$v_1^1 = \frac{k}{n}$$

We define the diversity of a given population (with k '1's) as

$$v_k = \frac{1}{2} \sum_{i=1}^{C} \sum_{a=0}^{|A|-1} v_i^a (1 - v_i^a) \tag{1.7}$$

In our case, with bits ($|A| = 2$) and one component ($C = 1$), we get the following:

$$v_k = \frac{1}{2} \sum_{i=1}^{1} \sum_{a=0}^{1} v_i^a (1 - v_i^a) =$$

$$\frac{1}{2} [v_1^0 (1 - v_1^0) + v_1^1 (1 - v_1^1)] =$$

$$\frac{1}{2} [\frac{n-k}{n} (1 - \frac{n-k}{n}) + \frac{k}{n} (1 - \frac{k}{n})] =$$

$$\frac{kn - k^2}{n^2}$$

Our total diversity d_p for a given distribution p is

$$d_p = \sum_{k=0}^{n} v_k p_k \tag{1.8}$$

What we want is to have a diversity of $p(1-p)$, exactly the variance of a population of infinite size. What we get is somewhat different. To determine by which factor our result differs from the wanted result, we divide our d_p by $p(1-p)$:

$$\frac{d_p}{p(1-p)} = \sum_{k=0}^{} n \frac{kn - k^2}{n^2} p^k (1-p)^{n-k} \binom{n}{k} =$$

$$\frac{1}{n^2 p(1-p)} \sum_{k=0} nk(n-k)p^k(1-p)^{n-k}\binom{n}{k} =$$

$$\frac{1}{n^2} \sum_{k=1} nk(n-k)p^{k-1}(1-p)^{n-k-1}\binom{n}{k}$$

It can be shown that this can be reduced further. The exact proove will be given elsewhere, here is just the observed result:

$$d = d_p = 1 - \frac{1}{n} \tag{1.9}$$

So it is independant from p, but we still have assumed $|A| = 2$ and $C = 1$. Tests have shown that we get the same result with a random value of C and $|A|$, although the formula in between will be different, especially (4). The formula will be generalized in chapter 3. TODO

According to Shapiro [1] we also know that selecting l individuals from this population of size n will result in a diversity loss (compared to $p(1-p)$) of the same factor $1 - \frac{1}{n}$.

To recap:

We have a distribution p and we generate a new population of the size n. In the optimal case ($n = \text{inf}$) we get the expected variance of $p(1-p)$. We calculated that the real variance is $p(1-p)(1-\frac{1}{n})$ and we know that creating a new population and selecting $\tilde{n}$ individuals from that population (in a flat fitness landscape) will result in the variance $p(1-p)(1-\frac{1}{\tilde{n}})$.

Now, the idea is to not create the new population with p but with a distribution q that fulfills the equation

$$p(1-p) = xq(1-q) \tag{1.10}$$

with x fulfilling the equation:

$$1 - \frac{1}{\tilde{n}} = x(1 - \frac{1}{n}) \Leftrightarrow x = \frac{(\tilde{n}-1)n}{(n-1)\tilde{n}} \tag{1.11}$$

So we have:

$$p(1-p) = q(1-q)\frac{(\tilde{n}-1)n}{(n-1)\tilde{n}} \Leftrightarrow$$

$$-q^2 + q - (-p^2 + p)\frac{(n-1)\tilde{n}}{(\tilde{n}-1)n} = 0 \Leftrightarrow$$

$$q_{1/2} = \frac{1}{2}(1 \pm \sqrt{1 - 4(-p^2 + p)\frac{(n-1)\tilde{n}}{(\tilde{n}-1)n}})$$

For the case of $\tilde{n} = \frac{n}{2}$ (i.e. we select half of the population to generate a new distribution vector) we would get

$$q_{1/2} = \frac{1}{2}(1 \pm \sqrt{1 - 4(-p^2 + p)\frac{(\tilde{n}-1)2\tilde{n}}{(2\tilde{n}-1)\tilde{n}}}) =$$

$$\frac{1}{2}(1 \pm \sqrt{1 - 4(-p^2 + p)\frac{2\tilde{n} - 2}{2\tilde{n} - 1}})$$

For $1 - 4(-p^2 + p)\frac{(n-1)\tilde{n}}{(\tilde{n}-1)n} < 0$ we get a negative value in square root. The border values for p are:

$$1 - 4(-p^2 + p)\frac{(n-1)\tilde{n}}{(\tilde{n}-1)n} = 0 \Leftrightarrow$$

$$p_{1/2} = \frac{1}{2}(1 \pm \sqrt{1 - \frac{(\tilde{n}-1)n}{(n-1)\tilde{n}}})$$

So if our p is within p_1 and p_2

$$\frac{1}{2}(1 - \sqrt{1 - \frac{(\tilde{n}-1)n}{(n-1)\tilde{n}}}) < p < \frac{1}{2}(1 + \sqrt{1 - \frac{(\tilde{n}-1)n}{(n-1)\tilde{n}}})$$

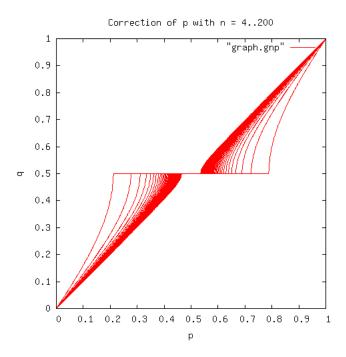we cannot increase the variance any further and have to set $q = 0.5$.

Figure 1.1: For a set of values for n (with $n = 2\tilde{n}$) ranging from 4 (farthest on the left and right) to 200 (nearest to $f(p) = p$)

1.3 Code

Finally we can put our results into a code:

```
if( p < 0.5*(1 - sqrt( (2*ñ-2) / (2*ñ-1) )) )
q = 0.5 * (1 - sqrt( 4*p*(1-p) * (2*ñ-1) / (2*ñ-2) ) );
else
if( p > 0.5*(1 + sqrt( (2*ñ-2) / (2*ñ-1) )) )
q = 0.5 * (1 + sqrt( 4*p*(1-p) * (2*ñ-1) / (2*ñ-2) ) );
else
q = 0.5;
```

or for the general formula with $n \neq 2\tilde{n}$:

```
if( p < 0.5*(1 - sqrt( (ñ-1)*n / ((n-1)*ñ) ) ) )
q = 0.5 * (1 - sqrt( 4*p*(1-p) * (n-1)*ñ / ((ñ-1)*n) ) );
else
if( p > 0.5*(1 + sqrt( (ñ-1)*n / ((n-1)*ñ) ) ) )
q = 0.5 * (1 + sqrt( 4*p*(1-p) * (n-1)*ñ / ((ñ-1)*n) ) );
else
q = 0.5;
```

The main advantage of this code is that only minimal changes to the code have to be made because this way of reducing the diversity loss is problem independant (at least for bitstrings on a flat landscape in UDMA). The code has to be inserted just after determining the distribution p.

1.4 Multiple components

If we set $C > 1$, i.e. bitstrings longer than one bit, we will get similar results. While equation 3 does use C to determine the diversity of a generated population we can look at each component seperately as they are not connected in UMDA (in a flat fitness landscape). All bits belonging to one component in the population are created with an own distribution p that is independant from the other p. Therefor we can handle a problem with $C > 1$ like C seperate problems, each with one component. How interconnected components would affect the diversity out of the scope of this paper and remains to be investigated.

1.5   Tests

1.5.1   General test configuration

UMDA

1.5.2   'Needle in a Haystack' problem

flat fitness landscape

1.5.3   OneMax problem

1.6   Conclusions

1.7   Fields of further research

In the following chapter I will run several tests with different parameters. In the third chapter I will discuss further applications

Bibliography

[1] SHAPIRO, J.L.: *Diversity loss in general estimation of distribution algorithms*, 2006.