View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

ELSEVIER

Theoretical Computer Science 287 (2002) 145–165

Theoretical Computer Science

www.elsevier.com/locate/tcs

Evolutionary computation and Wright's equation

H. Mühlenbein*, Th. Mahnig

Real World Computing Partnership—RWCP, Theoretical Foundation, GMD—Forschungszentrum Informationstechnik Laboratory D-53754 Sankt Augustin, Germany

Abstract

In this paper, Wright's equation formulated in 1931 is proven and applied to evolutionary computation. Wright's equation shows that evolution is doing gradient ascent in a landscape defined by the average fitness of the population. The average fitness W is defined in terms of marginal gene frequencies p_i. Wright's equation is only approximately valid in population genetics, but it exactly describes the behavior of our univariate marginal distribution algorithm (UMDA). We apply Wright's equation to a specific fitness function defined by Wright. Furthermore we introduce mutation into Wright's equation and UMDA. We show that mutation moves the stable attractors from the boundary into the interior. We compare Wright's equation with the diversified replicator equation. We show that a fast version of Wright's equation gives very good results for optimizing a class of binary fitness functions. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithms; Linkage equilibrium; Replicator equation; Search distributions; Factorization of distribution; Wright's equation; Maximum principle; Microscopic and macroscopic evolutionary algorithm

1. Introduction

The purpose of this paper is twofold. First we analyze a powerful evolutionary algorithm using univariate marginal distributions (UMDA) instead of recombination and mutation of strings as done by genetic algorithms. The behavior of a specific instance of this algorithm can be mathematically described by a set of difference equations for the marginal distributions. We later found that these equations have been proposed in population genetics by Wright as early as 1931 [14]. Therefore our second purpose is to relate our analysis to the discussion of these equations in population genetics.

* Corresponding author.
E-mail address: muehlenbein@gmd.de (H. Mühlenbein).

0304-3975/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(02)00098-1

Wright himself [17] gave a short historical overview about the difference equation which later was termed Wright's equation. "Only single-gene distributions were dealt with mathematically in the 1931 paper [14] but these were merely considered to be indications of the sort of thing that is happening in the many dimensions to which the verbal discussion was devoted. In a later paper [15], a formula was given for Δp in cases of multi-factorial heredity and an intermediate optimum. This was taken up more generally in 1937 [16] with the introduction of the symbol W for selective value of the genotypes as a whole. It was assumed that the local population in question was breeding at random and that it was sufficiently accurate that all loci were combined random". For binary alleles the key formula is as follows:

$$\Delta p_i = p_i(t+1) - p_i(t) = p_i(t)(1 - p_i(t))\,\frac{\partial W/\partial p_i}{W}, \tag{1}$$

where i denotes a locus, and p_i is the frequency of $allele = 1$. Wright's extension to multiple alleles is wrong and therefore omitted.

In 1970 Wright [18] remarked: "The appearance of this formula is deceptively simple. Its use in conjunction with other components is not such a gross oversimplification in principle as has sometimes been alleged. Obviously, calculations can be made only from rather simple models, involving only a few loci or simple patterns of interaction among many similarly behaving loci. Apart from application to simple systems, the greatest significance of the general formula is that its form brings out properties of systems that would not be apparent otherwise".

Wright's equation was especially criticized by Fisher [4]. His main argument was that the evaluation of W needs on the order of 2^n terms. The interpretation of Wright's equation is debated till today [3]. For our UMDA the difference equation is exact, whereas it is an approximation in population genetics.

The outline of the paper is as follows. In Section 2 we prove Wright's equation. Then the landscape metaphor in evolutionary computation is discussed. We give a simple formula for computing the average fitness W. In Section 4 we use Wright's original illustration as a numerical example. Then mutation is discussed. In Section 6 the relation between Wright's equation and the replicator equation is investigated. Then numerical results of the different approaches are discussed. We summarize the paper in Section 9 by discussing three royal roads for evolutionary optimization.

2. Univariate marginal distribution algorithm

Let $\mathbf{x} = (x_1, \ldots, x_n)$ denote a vector, $x_i \in \Lambda_i = \{0, 1, 2, \ldots, m_i\}$. We use the following conventions. Capital letters X_i denote variables, small letters x_i assignments. Let a function $f : \boldsymbol{X} \to \mathbb{R}_{\geqslant 0}$ be given. We consider the optimization problem $\mathbf{x}_{\text{opt}} = \text{argmax}\, f(\mathbf{x})$.

Definition. Let $p(\mathbf{x}, t)$ denote the probability of $\mathbf{x}$ in the population at generation t. Then $p_i(x_k, t) = \sum_{\mathbf{x}, X_i = x_k} p(\mathbf{x}, t)$ defines the univariate marginal distributions of variable X_i.

Note that $\sum_{x_k \in \Lambda_i} p_i(x_k, t) = 1$. This means that the univariate marginal distributions are not independent. For notational simplicity we will consider $p_i(0)$ to be the dependent parameter, which can be eliminated, if appropriate. We write $p_i(x_k)$ if just one generation is discussed. We recall two popular recombination/crossover operator used in genetic algorithms.

Definition. Let two strings $\mathbf{x}$ and $\mathbf{y}$ be given. In *one-point crossover* the string $\mathbf{z}$ is created by randomly choosing a crossover point $0 < l < n$ and setting $z_i = x_i$ for $i \leqslant l$ and $z_i = y_i$ for $i > l$. In *uniform crossover* z_i is randomly chosen with equal probability from $\{x_i, y_i\}$.

In order to derive Wright's equation, we have to introduce a special distribution.

Definition. *Robbins' proportions* are defined by the distribution

$$\pi_p(\mathbf{x}, t) := \prod_{i=1}^{n} p_i(x_i, t). \tag{2}$$

A population in Robbins' proportions is called to be in *linkage equilibrium* in population genetics.

In [7,12] we have shown: All complete recombination schemes lead to the same univariate marginal distributions after one step of selection and recombination. If recombination is used for a number of times without selection, then the genotype frequencies converge to linkage equilibrium. This means that *all genetic algorithms are identical if after one selection step recombination is done without selection a sufficient number of times.* This fundamental algorithm keeps the population in linkage equilibrium. Wright also assumed linkage equilibrium to derive his equation.

Instead of performing recombination a number of times in order to converge to linkage equilibrium, one can achieve this in one step by *gene pool recombination* [11]. In gene pool recombination a new string is computed by randomly taking for each loci a gene from the distribution of the selected parents. This means that gene x_i occurs with probability $p^s(x_i)$ in the next population. $p^s(x_i)$ is the distribution of x_i in the selected parents. Thus, new strings $\mathbf{x}$ are generated according to the distribution

$$p(\mathbf{x}, t+1) = \prod_{i=1}^{n} p_i^s(x_i, t). \tag{3}$$

One can simplify the algorithm still more by directly computing the univariate marginal frequencies from the data. Then Eq. (3) can be used to generate new strings. This method is used by the *univariate marginal distribution algorithm* (UMDA).

UMDA

STEP 0: Set $t \Leftarrow 1$. Generate $N \gg 0$ points randomly.

STEP 1: Select $M \leqslant N$ points according to a selection method. Compute the marginal frequencies $p_i^s(x_i, t)$ of the selected set.

STEP 2: Generate N new points according to the distribution $p(\mathbf{x}, t+1) = \prod_{i=1}^{n} p^s(x_i, t)$. Set $t \Leftarrow t + 1$.

STEP 3: If termination criteria are not met, go to STEP 1.

Let $v = \sum_{i=1}^{n}(m_i + 1)$. UMDA formally depends on v parameters, the marginal distributions $p_i(x_k)$. We now consider the average $\bar{f}(t) = \sum_x p(x,t)f(x)$ as a function which depends on $p_i(x_k)$. To emphasize this dependency we write (in accordance with Wright)

$$W(p_1(x_1), p_1(x_2), \ldots, p_n(x_{m_n})) := \bar{f}(t). \tag{4}$$

We can now formulate difference equations, describing the dynamic behavior of $p_i(x_k)$.

Theorem 1. *For infinite populations and proportionate selection UMDA changes the gene frequencies as follows*:

$$p_i(x_k, t+1) = p_i(x_k, t)\,\frac{\bar{f}_i(x_k, t)}{W(t)}, \tag{5}$$

where $\bar{f}_i(x_k,t) = \sum_{\mathbf{x}, X_i = x_k} f(\mathbf{x}) \prod_{j \neq i}^{n} p(x_j, t)$. *Using a formal derivative* $\partial W / \partial p_i(x_k)$, *the equations can also be written as*

$$p_i(x_k, t+1) = p_i(x_k, t) + p_i(x_k, t)\,\frac{\partial W/\partial p_i(x_k) - W(t)}{W(t)}. \tag{6}$$

Furthermore the average fitness W *never decreases.*

$$W(t+1) \geqslant W(t). \tag{7}$$

The theorem has been proven in [7]. Note that the derivatives of W are obtained by formal differentiation of Eq. (4).

Example. $f(x) = \sum_i a_i x_i, \; x_i \in \{0, 1\}$.

After some tedious manipulations one obtains:

$$W(\mathbf{p}) = \sum_i a_i p_i(1),$$

$$\frac{\partial W}{\partial p_i(1)} = a_i + \sum_{j \neq i} a_j p_j(1).$$

This gives the difference equation

$$\Delta p_i(1) = p_i(1,t)(1 - p_i(1,t))\,\frac{a_i}{\sum_i a_i p_i(1,t)}. \tag{8}$$

This equation has been approximately solved in [8].

This example shows that the expressions for W and its derivatives can be surprisingly simple. $W(\mathbf{p})$ can be obtained from $f(x)$ by exchanging x_i with $p_i(1)$. But the formal derivation of $W(p)$ cannot be obtained from the simple $W(p)$ expression! We will investigate the computation of W and its gradient in the following section. Next we formulate and prove Wright's equation. It is obtained by using real derivatives in Euclidian spaces. Thus we have to eliminate one parameter for each locus. In order to minimize possible confusion later we denote the average fitness $\bar{f}(t)$ with parameters $p_i(0)$ eliminated as $\tilde{W}$.

Definition. Let $\bar{\Lambda}_i = \Lambda_i \backslash \{0\}$. If we eliminate $p_i(X_i = 0, t)$ in $W(\mathbf{p})$ by inserting $1 - \sum_{k \in \bar{\Lambda}_i} p_i(x_k, t)$ then we obtain $\tilde{W}(\mathbf{p})$.

We can now formulate the main theorem.

Theorem 2 (Wright's Equation). *For infinite populations and proportionate selection UMDA changes the gene frequencies as follows*:

$$p_i(x_k, t+1) = p_i(x_k, t) + p_i(x_k, t) \times \frac{\partial \tilde{W} / \partial p_i(x_k) - \sum_{j \in \bar{\Lambda}_i} p_i(x_k, t) \partial \tilde{W} / \partial p_i(x_k)}{\tilde{W}(t)}. \tag{9}$$

Proof. Note that

$$\frac{\partial \tilde{W}}{\partial p_i(x_k)} = \bar{f}_i(x_k, t) - \bar{f}_i(0, t).$$

Furthermore we have

$$\sum_{k \in \Lambda_i} p_i(x_k, t) \bar{f}_i(x_k, t) = \tilde{W}(t).$$

We next compute

$$\bar{f}_i(x_k, t) - \tilde{W}(t) = \frac{\partial \tilde{W}(t)}{\partial p_i(x_k)} + \bar{f}_i(0, t) - \sum_{j \in \Lambda_i} p_i(x_j, t) \bar{f}_i(x_j, t),$$

$$\sum_{j \in \Lambda_i} p_i(x_j, t) \bar{f}_i(x_j, t) = \sum_{j \in \bar{\Lambda}_i} p_i(x_j, t) \frac{\partial \tilde{W}(t)}{\partial p_i(x_j)} + \bar{f}_i(0, t).$$

Inserting this equation into the difference Eq. (5) gives the conjecture after some simple manipulations. □

This is the exact formulation of Wright's equation for multiple alleles at n loci. For binary alleles we obtain Eq. (1). The above equations completely describe the dynamics of UMDA with proportionate selection.

There has been a fierce battle between Wright and Fisher concerning the importance of Wright's equation. Fisher especially criticized Wright's average fitness function W. He wrote [4]: "Prof. Wright confuses the number of genotypes, e.g. 3^{1000},[1] (for 1000 loci) which may be distinguished among individuals, with the continuous field of variation of gene frequencies. Even if a potential function, such as W is supposed to be, really existed, the large number of genotypes supplies no reason for thinking that

[1] Diploid organism have on each loci 4 genotypes with 3 independent frequencies.

even one peak, maximal for variations of all gene ratios should occur in this field of variation".

Wright was not able to refute this argument. In 1963 he wrote [17]: "The summation in the formula for $\tilde{W}$ has, however, as many terms as there are kinds of genotypes, 3^{1000} for 1000 pairs of alleles. This, of course, points to a practical difficulty in calculating Δp for more than two or three pairs of interacting factors, unless a regular model is postulated. There was no confusion".

We will show in the next section that both, Fisher and Wright, overlooked that if the fitness function is simple, $\tilde{W}$ can be easily computed. This is discussed next.

3. Average fitness and the landscape metaphor

Wright is also the originator of the landscape metaphor now popular in evolutionary computation and population genetics. Unfortunately Wright used two quite different definitions for the landscape, apparently without realizing the fundamental distinction between them. The first landscape describes the relation between the genotypes and their fitness, while the second describes the relation between the allele frequencies in a population and its mean fitness.

The first definition is just the fitness function $f(x)$ used in evolutionary computation, the second one is the average fitness $\tilde{W}(p)$. The second definition is much more useful, because it lends to a quantitative description of the evolutionary process, i.e. Wright's equation.

For notational simplicity we only derive the relation between $f(x)$ and $\tilde{W}$ for binary alleles. Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ with $\alpha_i \in \{0, 1\}$ be a multi-index. We define $0^0 := 1$:

$$\mathbf{x}^\alpha := \prod_i x_i^{\alpha_i}.$$

Definition. The representation of a binary discrete function using the ordering according to function values is given by

$$f(\mathbf{x}) = f(0, \ldots, 0)(1 - x_1) \cdots (1 - x_n) + \cdots + f(1, \ldots, 1)\, x_1 \cdots x_n. \tag{10}$$

The representation using the ordering according to variables is

$$f(\mathbf{x}) = \sum_\alpha a_\alpha x^\alpha \tag{11}$$

$\max\{|\alpha|_1 = \sum_i \alpha_i\colon\ a_\alpha \neq 0\}$ is called the order of the function.

In both representations the function is linear in each variable x_i. The following lemma is obvious.

Lemma. *The two representations are unique. There exist a unique matrix A of dimension $2^n * 2^n$ such that*

$$a_\alpha = (Af)_\alpha$$

We now use this result for $\tilde{W}$.

Lemma. *$\tilde{W}(\mathbf{p}) := \bar{f}(t)$ is an extension of $f(x)$ to S. There exist two representations for $\tilde{W}(p)$. These are given by*

$$\tilde{W}(\mathbf{p}) = f(0,\ldots,0)(1-p_1)\cdots(1-p_n) + \cdots + f(1,\ldots,1)p_1\cdots p_n, \tag{12}$$

$$\tilde{W}(\mathbf{p}) = \sum_{\alpha} a_\alpha p^\alpha. \tag{13}$$

If the function is given in analytical form (Eq. (11)) and the order of the function is bounded by a constant independent of n, then $\tilde{W}(\mathbf{p})$ can be computed in polynomial time. The proofs in this section are somewhat informal. The above lemma can rigorously be proven by Moebius inversion.

We will now characterize the attractors of UMDA. Let $S_i = \{q_i \mid \sum_{k\in\bar{\Lambda}_i} q_i(x_k) \leqslant 1;\ 0 \leqslant q_i(x_k) \leqslant 1\}$ and $S = \prod_i S_i$ the Cartesian product. For simplicity we consider the binary alleles $x_i \in \{0,1\}$. Then $S = [0,1]^n$.

Theorem 3. *The stable attractors of Wright's equation are at the corners of S, i.e. $p_i \in \{0,1\}$ $i = 1,\ldots,n$. In the interior there are only saddle points or local minima where grad $W(p) = 0$. The attractors are local maxima of $f(x)$ according to one bit changes. Wright's equation solves the continuous optimization problem* argmax$\{\tilde{W}(\mathbf{p})\}$ *in S by gradient ascent.*

Proof. W is linear in p_i. Therefore it cannot have any local maxima in the interior. Points with *grad* $W(p) = 0$ are unstable fixpoints of UMDA.

We next show that boundary points which are not local maxima of $f(x)$ cannot be attractors. We prove the conjecture indirectly. Without loss of generality, let the boundary point be $\hat{p} = (1,\ldots,1)$. We now consider an arbitrary neighbor, i.e. $p^* = (0,1,\ldots,1)$. The two points are connected at the boundary by

$$p(z) = (1-z, 1, \ldots, 1), \quad z \in [0,1].$$

We know that $\tilde{W}$ is *linear* in the parameters p_i. Because $\tilde{W}(p^*) = f(0,1,\ldots,1)$ and $\tilde{W}(\hat{p}) = f(1,\ldots,1)$ we have

$$\tilde{W}(p(z)) = f(1,\ldots,1) + z \cdot [f(0,1,\ldots,1) - f(1,\ldots,1)]. \tag{14}$$

If $f(0,1,\ldots,1) > f(1,\ldots,1)$ then $\hat{p}$ cannot be an attractor of UMDA. The mean fitness increases with z. □

Eq. (13) can also be used to compute the derivative. It is given by

$$\frac{\partial \tilde{W}(p)}{\partial p_i(1)} = \sum_{\alpha \mid \alpha_i = 1} a_\alpha p^{\alpha'} \tag{15}$$

with $\alpha_i' = 0$, $\alpha_j' = \alpha_j$.

The extension of the above lemma to multiple alleles and multivariate distributions is straightforward, but the notation becomes difficult.

4. Wright's simple illustration

From 1963 Wright used the same example to illustrate his theory. Wright's example consists of a diploid organism with four loci with binary alleles $\{A,a;B,b;C,c;D,d\}$. We have to transform this example to a haploid organism. We just map $(A,A),(B,B)$, $(C,C),(D,D)$ to allele 1 and the other three combinations to 0. Furthermore we multiply the fitness values by 8 in order to obtain integer values. Then Wright's fitness function can be mathematically be written as

$$\begin{aligned} f(x_1,x_2,x_3,x_4) = {} & 7(x_1+x_2)+6(x_3+x_4) \\ & -4(x_1x_2+x_1x_3+x_1x_4+x_2x_3+x_2x_4+x_3x_4). \end{aligned} \tag{16}$$

We abbreviate $p_i := p_i(1)$. Then we obtain

$$\begin{aligned} \tilde{W}(p_1,p_2,p_3,p_4) = {} & 7(p_1+p_2)+6(p_3+p_4) \\ & -4(p_1p_2+p_1p_3+p_1p_4+p_2p_3+p_2p_4+p_3p_4). \end{aligned} \tag{17}$$

The derivatives are given by

$$\frac{\partial\tilde{W}}{\partial p_1} = 7-4(p_2+p_3+p_4), \quad \frac{\partial\tilde{W}}{\partial p_2} = 7-4(p_1+p_3+p_4),$$

$$\frac{\partial\tilde{W}}{\partial p_3} = 6-4(p_1+p_2+p_4), \quad \frac{\partial\tilde{W}}{\partial p_4} = 6-4(p_1+p_2+p_3).$$

Now Wright's equation can be used for analysis. Setting $grad\,W = 0$ we obtain a saddle point at $p_1 = p_2 = \frac{5}{12}$ and $p_3 = p_4 = \frac{2}{3}$. Wright gives in his figures some curious numbers about local saddle points, despite he obviously used his formulae not in a mathematical sense.

The local maxima and therefore the attractors of UMDA are the six genotypes with two bits on and two bits off. We have $(0,0,1,1)$ with fitness $8,(1,0,1,0),(1,0,0,1)$, $(0,1,1,0),(0,1,0,1)$ with fitness 9, and $(1,1,0,0)$ with fitness 10. One would be interested to characterize the attractor regions for these six maxima. But this is a formidable task in four dimensions.

In order to get some results, we make a simplification. The equations are symmetric in p_1,p_2 and p_3,p_4. If we start with equal probabilities $p_1 = p_2$ and $p_3 = p_4$, then the dynamics depends on two variables only. But note that with this constraint the four intermediate local maxima are not accessible for the dynamics.

Fig. 1 is a plot of the $\tilde{W}$ landscape. The saddle point $p_1 = \frac{5}{12}$ and $p_3 = \frac{2}{3}$ can be clearly recognized. There are two attractors at the boundary, (0,1) and (1,0), the global optimum. Even using the picture it is very difficult to determine the attractor regions. The following two regions are easy to determine. If $p_1 > \frac{5}{12}$ and $p_3 < \frac{2}{3}$ then

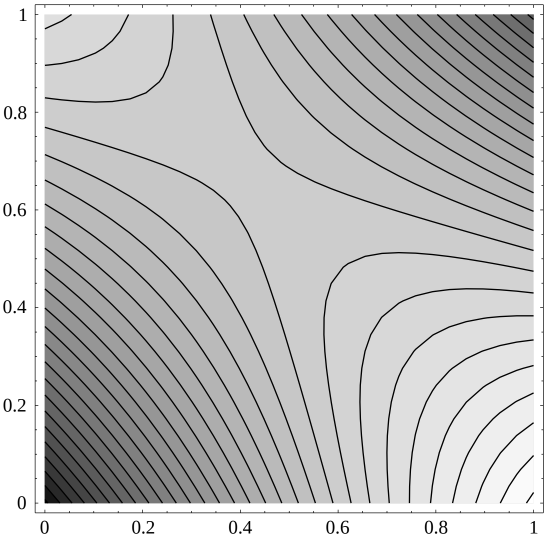

Fig. 1. $\tilde{W}$ landscape for $p_1 = p_2$ and $p_3 = p_4$.

the attractor is (1,0). If $p_1 < \frac{5}{12}$ and $p_3 > \frac{2}{3}$ then the attractor is (0,1). For the remaining area the attractor has to be determined by actual simulation.

But the assumption $p_1 = p_2$ and $p_3 = p_4$ is very restrictive and unstable. Small differences in the initial values may have a dramatic impact. Wright's equation is deterministic. It behaves unstable if the initial points are at the boundary of two attractor regions. At these points the behavior of UMDA and Wright's equation are very different.

Despite its importance for theoretical population genetics. Wright's equation was seldom used quantitatively. To the best of our knowledge, only Barton [1] has numerically applied Wright's equation. Unfortunately he did not use Wright's original example, but a simpler function with only one level of local optima. Furthermore, he computed the average fitness with a complicated approximation valid only for Gaussian fitness values.

We now turn to another problem in evolutionary computation—the introduction of randomness by mutation.

5. Mutation, Bayes prior, and population size

There exist many investigations to compute good mutation rates for genetic algorithms. Our approach allows to investigate the problem from a dynamic perspective. In Wright's equation all local optima are attractors at the boundary. Any evolution will stop at the boundary. This is not desirable. Therefore, Wright included mutation into his equation as a background operator. With mutation the local attractors are not at the boundary any more, but move into the interior.

Wright [15] included mutation with a recurrent symmetric mutation rate of $0 \leqslant \mu < 1$ as follows into his equation:

$$\Delta p_i = p_i(t)(1 - p_i(t)) \frac{\partial \tilde{W}/\partial p_i}{\tilde{W}} - \mu(p_i(t) - (1 - p_i(t))). \tag{18}$$

This is a simplification of the biological process to be modeled. Mutation is selective neutral, i.e. it occurs after selection. Nevertheless, the equation fulfils its goal. Mutation changes the landscape. The attractors are now defined by a *dynamic equilibrium between mutation and selection*. For $\mu \gg 0$ we have a stable attractor at $p_i = 0.5$, for $\mu = 0$ we have the multi-modal landscape with attractors at the boundary discussed before.

Let us take $\mathit{OneMax} = \sum x_i$ as example. Without mutation the only stable attractor is obviously $p_i = 1,\ i = 1, \ldots, n$. With mutation the attractor is obtained by setting $\Delta p_i = 0$ in Eq. (18). The attractor is given by

$$p_i^* = 1 - \frac{\mu}{2\mu + (1/n)}. \tag{19}$$

If we set $\mu = 1/n$ we obtain $p_i = \frac{2}{3}$. For $\mu = 0.5/n$ we have $p_i = 0.75$. These attractors are far away from the global optimum $p_i = 1$. At the attractor the optimum will be generated with probability

$$p(x_{\mathrm{opt}}) = p_i^n.$$

Even for small $n = 20$ the probability is very small, $p = 0.0032$. The mutation rate has to be much smaller. From Eq. (19) we easily obtain the result.

Corollary. *For any $\mu > 0$ we have for* OneMax

$$\lim_{n \to \infty} p(x_{\mathrm{opt}}) = 0.5 \tag{20}$$

The above result shows again that proportionate selection selects very weak. For large n the mutation rate has to be 0. If the selection is stronger, a higher mutation rate can be used. We give a result for truncation selection. For *OneMax* we have earlier derived an approximate difference equation [7]. If we add mutation to the approximation we obtain

$$\Delta p = \frac{I}{n}\sqrt{n p(t)(1 - p(t))} - 2\mu p + \mu.$$

Tournament selection can be approximated similarly. For binary tournament selection we have $I = 1/\sqrt{\pi} = 0.56$. Setting $\Delta p_i = 0$ we obtain the expression

$$p^* = \frac{1}{2} + \frac{I}{2\sqrt{I^2 + 4\mu^2 n}}.$$

For $n = 64$ and $\mu = 1/n$ we obtain $p = 0.959$. For $\mu = 0.5/n$ we obtain $p = 0.988$. Both values are in excellent agreement with actual simulation runs with UMDA. We next compute the probability that at this attractor the optimum is generated.

Theorem 4. *Let the mutation rate be* $\mu = k/n$. *Then for tournament selection and the linear fitness function* OneMax *we have*

$$\lim_{n\to\infty} p(x_{\mathrm{opt}}) = \mathrm{e}^{-k^2/I^2}. \tag{21}$$

Proof. For $\mu = k/n$ and keeping only factors with $1/n$ we obtain

$$\lim_{n\to\infty}\left(\frac{1}{2}+\frac{I}{2\sqrt{I^2+4\mu^2 n}}\right)^n \approx \lim_{n\to\infty}\left(1-\frac{k^2}{I^2 n}\right)^n \approx \mathrm{e}^{-k^2/I^2}. \quad \square \tag{22}$$

Thus, truncation selection can support a mutation rate of $\mu = 1/n$. This mutation rate we have recommended earlier [7]. We will now introduce mutation into UMBA in a concise statistical manner. The technique is called *Bayes prior*. Usually the empirical probabilities are computed by the maximum likelihood estimator. For N samples with $m \leqslant N$ instances of x the estimate is defined by

$$\hat{p}(x) = \frac{m}{N}.$$

For $m = N$ we obtain $p(x) = 1$ and for $m = 0$ we obtain $p(x) = 0$. This leads to the gene fixation problem, because both values are attractors. The fixation problem is reduced if $\hat{p}(x)$ is restricted to an interval $0 < p_{\min} \leqslant \hat{p}(x) \leqslant 1 - p_{\min} < 1$. This is exactly what results from the *Bayesian estimation*. The estimate $\hat{p}(x)$ is the expected value of the posterior distribution after applying Bayes formula to a prior distribution and the given data. For binary variables x the estimate

$$\hat{p}(x) = \frac{m+r}{N+2r} \tag{23}$$

is used with $r > 0$. r is derived from a Bayesian prior. $r = 1$ is the result of the uniform Bayesian prior. The larger r, the more the estimates tend towards $\frac{1}{2}$. The reader interested in a derivation of this estimate in the context of Bayesian networks is referred to [6].

We now incorporate this estimate into UMDA with proportionate selection. The frequencies are changed as follows:

$$p_i(t+1) = \frac{p_i^{\mathrm{s}}(t)N + r}{N+2r},$$

where $p_i^{\mathrm{s}}(t)$ is given by Wright's equation (1). Setting $\gamma = r/N$ we obtain

$$\Delta p_i(t) = p_i(t) + p_i(t)(1-p_i(t))\frac{\partial\tilde{W}}{\tilde{W}} + \frac{2\gamma}{1+2\gamma}$$
$$\left(p_i(t) - p_i(t)(1-p_i(t))\frac{\partial\tilde{W}}{\tilde{W}}\right) + \frac{\gamma}{1+2\gamma}. \tag{24}$$

Note that Eq. (24) is quite similar to Eq. (18) if we set the mutation rate $\mu = \gamma/(1+2\gamma)$. The mutation rate is now multiplied by $p_i(t+1)$ instead of $p_i(t)$.

Table 1
Attractors for different mutation rates, proportionate and tournament selection

Sel.	r/N	p_1	p_2	p_3	p_4	p_1^a	p_2^a	p_3^a	p_4^a
Prop	0.02	0.7	0.7	0.3	0.3	0.92	0.93	0.09	0.08
Prop	0.03	0.1	0.1	0.9	0.9	0.89	0.89	0.12	0.13
Prop	0.04	0.1	0.9	0.1	0.9	0.86	0.84	0.16	0.18
Prop	0.04	0.1	0.1	0.9	0.9	0.86	0.83	0.16	0.15
Prop	0.06	0.1	0.1	0.9	0.9	0.82	0.81	0.22	0.31
Tour	0.04	0.1	0.1	0.9	0.9	0.02	0.02	0.97	0.96
Tour	0.06	0.1	0.1	0.9	0.9	0.94	0.94	0.08	0.07
Tour	0.08	0.1	0.1	0.9	0.9	0.92	0.92	0.12	0.12

We are now able to compute the attractors for Wright's equation, given a Bayes prior r. The attractors can be computed by setting $\Delta p_i = 0$. This defines a nonlinear system of four equations. We compute a numerical example by setting $r = 20$ and $N = 1000$. If we assume $p_1 = p_2$ and $p_3 = p_4$ we obtain three solutions at $(0.158871, 0.889656), (0.349797, 0.723039)$, and $(0.932256, 0.0914656)$. The second solution is unstable, so we have two attractors. Next we increase r from 20 to 40. Now we obtain only one attractor at $(0.864838, 0.171577)$. This attractor is nearby the global optimum! Thus mutation is able to flatten the multi-modal landscape. This can easily be concluded from Eq. (24). For $\gamma \gg 0$ we have a unique attractor at $p_i = 0.5$, for $\gamma = 0$ all local maxima are attractors at the corners.

We compare the analytical results with UMDA simulation runs in Table 1. For a population size of $N = 1000$ the coincidence of the analytical results with the simulation results is very good. We also show numerical results for tournament selection.

Tournament selection can handle a higher mutation rate than proportionate selection. For $r/N = 0.04$ we still have an attractor at the smallest local optimum. For $r/N = 0.06$ this attractor has vanished.

The above investigations will now be used to obtain estimates for the population size and the mutation rate for tournament selection. From statistical arguments N should be some multiple of n for large n (i.e. $n \geqslant 30$) [10]. In order to determine r we observe the following. If r is too low, the attractors are changed only a little. If r is too large, then the attractor is in the interior, far away from the global optimum. But we do not want to move the attractors so far into the interior, that it is highly improbable to generate the optimum. This argument gives our goal. r should be so large that the attractors are far enough away from the attractors. But the attractor nearby the global optimum should be near enough, so that the optimum will be generated with certain probability.

Because it is too difficult to analytically compute the attractors for each given fitness function, we use the result for *OneMax*. This gives our rule of thumb.

Rule of thumb. For UMDA with tournament or truncation selection use a population size of $N \approx 3n$ and a Bayes prior of $r = 1/n$ for a problem of size n.

6. The replicator equation

In this section we investigate the relation between Wright's equation and a popular equation called *replicator equation*. Replicator dynamics is a standard model in evolutionary biology to describe the dynamics of growth and decay of a number of species under selection. Let $S=\{1,2,\dots,s\}$ be a set of species, p_i the frequency of species i in a fixed population of size N. Then the replicator equation is defined on a simplex $S^s=\{p\colon \sum p_i=1,\ 0\leqslant p_i\leqslant 1\}$

$$\frac{\mathrm{d}p_i}{\mathrm{d}t}=p_i(t)\left(f_i(p)-\sum_{i=1}^{s}p_i(t)f_i(p)\right), \tag{25}$$

f_i gives the fitness of species i in relation to the others. The replicator equation is discussed in detail in [5]. For the replicator equation a maximum principle can be shown.

Theorem 5. *If there exists a potential V with $\partial V/\partial p_i=f_i(p)$, then $\mathrm{d}V/\mathrm{d}t\geqslant 0$, i.e. the potential V increases using the replicator dynamics.*

If we want to apply the replicator equation to a binary optimization problem of size n, we have to set $s=2^n$. Thus the number of species is exponential in the size of the problem. The replicator equation can be used for small size problems only.

Voigt [13] had the idea to generalize the replicator equation by introducing continuous variables $0\leqslant p_i(x_k)\leqslant 1$ with $\sum_k p_i(x_k)=1$. Thus $p_i(x_k)$ can be interpreted as univariate probabilities.

Definition. The *diversified replicator equation* DRP is given by

$$\frac{p_i(x_k)}{\mathrm{d}t}=p_i(x_k)\left(f_{ik}(\mathbf{p})-\sum_{x_k}p_i(x_k)f_{ik}(\mathbf{p})\right). \tag{26}$$

For $n=1$ the pure replicator equation is obtained. The DRP equation can be used as the dynamic of a binary optimization problem by setting $m_i=2$ and $x_k\in\{0,1\}$.

Theorem 6. *If there exist potentials V_i with $\partial V_i/\partial p_i(x_k)=f_{ik}(\mathbf{p})$, then $\mathrm{d}V_i/\mathrm{d}t\geqslant 0$, i.e. the, potential V_i increases using the replicator dynamics.*

Proof. One easily shows that

$$\frac{\mathrm{d}V_i}{\mathrm{d}t}=\sum_{k=1}^{m}p_i(x_k)\left(f_{ik}(\mathbf{p})-\sum_{j=1}^{m}p_jf_{ij}(\mathbf{p})\right)^2\geqslant 0. \tag{27}$$

The conjecture follows from $p_i(x_k)\geqslant 0$ and $\sum_{x_k}p_i(x_k)=1$. □

DRP is a differential equation. Voigt [13] proposed the following discrete version.

Definition. The *discrete diversified replicator equation* DDRP is given by

$$p_i(x_k)(t+1) - p_i(x_k)(t) = p_i(x_k)(t)\,\frac{f_{ik}(\mathbf{p}) - \sum_{x_k} p_i(x_k) f_{ik}(\mathbf{p})}{\sum_{x_k} p_i(x_k) f_{ik}(\mathbf{p})}. \tag{28}$$

The name discrete diversified replicator equation was not a good choice. The DDRP is more similar to Wright's equation than to the replicator equation. This is the content of the next theorem.

Theorem 7. *If the average fitness $W(\mathbf{p})$ is used as potential, then Wright's equation and the discrete diversified replicator equation are identical.*

Proof. The average fitness is defined as

$$W(\mathbf{p}) = V(\mathbf{p}) = \sum_x a_x \prod_{i=1}^n p_i(x_i).$$

We compute the derivatives

$$\frac{\partial V(\mathbf{p})}{\partial p_i(1)} = \sum_{x|x_i=1} a_x \prod_{j\neq i}^n p_j(x_j),$$

$$\frac{\partial V(\mathbf{p})}{\partial p_i(0)} = \sum_{x|x_i=0} a_x \prod_{j\neq i}^n p_j(x_j).$$

Obviously,

$$p_i(1)\frac{\partial V}{\partial p_i(1)} + p_i(1)\frac{\partial V}{\partial p_i(1)} = V(\mathbf{p}).$$

The conjecture now follows from the proof of Wright's equation. □

We recently discovered that Baum and Eagon [2] have proved a discrete maximum principle for certain instances of the DDRP.

Theorem 8 (Baum and Eagon [2]). *Let $V(\mathbf{p})$ be a polynomial with nonnegative coefficients homogeneous of degree d in its variables $p_i(x_j)$ with $p_i(x_j) \geqslant 0$ and $\sum_{x_j} p_i(x_j) = 1$. Let $\mathbf{p}(t+1)$ be the point given by*

$$p_i(x_j, t+1) = \frac{p_i(x_j,t)\partial V/\partial p_i(x_j)}{\sum_{x_k} p_i(x_k)\partial V/\partial p_i(x_k)}. \tag{29}$$

The derivatives are taken at $\mathbf{p}(t)$. Then $V(\mathbf{p}(t+1)) > V(\mathbf{p}(t))$ unless $\mathbf{p}(t+1) = \mathbf{p}(t)$.

Eq. (29) is exactly the DDRP with a potential V. Thus the DDRP could be called the Baum–Eagon equation. From the above theorem the discrete maximum principle for Wright's equation follows by setting $V = W$ and $d = n$. Thus the potential is the average fitness, which is homogeneous of degree n.

7. The system dynamics approach to optimization

Theorem 8 shows that both, Wright's equation and the DDRP, maximize some potential. This means that both equations can be used for maximization. But there is a problem: both equations are deterministic. For difficult optimization problems, there exists a huge number of attractors, each with a corresponding attractor region. If the iteration starts at a point within the attractor region, it will converge to the corresponding attractor at the boundary. But if the iteration starts at points which lie at the boundary of two or more attractors, i.e. on the separatrix, the iteration will be confined to the separatrix. The deterministic system cannot decide for one of the attractors.

UMDA with a finite population does not have a sharp boundary between attractor regions. We model this behavior by introducing randomness. The new value $p_i(x_j, t+1)$ is randomly chosen from the interval

$$[(1-c)p_i'(x_j, t+1), (1+c)p_i'(x_j, t+1)],$$

where $p_i'(x_j, t+1)$ is determined by the deterministic equation, c is a small number. For $c = 0$ we obtain the deterministic equation. In order to use the difference equation optimally, we do not allow the boundary values $p_i = 0$ or $p_i = 1$. We use $p_i = p_{\min}$ and $p_i = 1 - p_{\min}$ instead.

A second extension concerns the determination of the solution. All dynamic equations presented use variables, which can be interpreted as probabilities. Thus instead of waiting that the dynamic system converges to some boundary point, we terminate the iteration at a suitable time and generate a set of solutions. Thus, given the values for $p_i(x_j)$ we generate points x according to the UMDA distribution $p(\mathbf{x}) = \prod_{i=1}^{n} p_i(x_i)$.

We can now formulate a family of optimization algorithms, based on difference equations (*DIFFOPT*).

DIFFOPT

STEP 0: Set $t \Leftarrow 0$ and $p_i(x_j, 0) = 0.5$ Input $p_{\min}$.

STEP 1: Compute $p_i'(x_j, t+1)$ according to a dynamic difference equation. If $p_i'(x_j, t+1) < p_{\min}$ then $p_i'(x_j, t+1) = p_{\min}$. If $p_i'(x_j, t+1) > 1 - p_{\min}$ then $p_i'(x_j, t+1) = 1 - p_{\min}$

STEP 2: Compute randomly $p_i(x_j, t+1)$ in the interval $(1-c)p_i'(x_j, t+1), (1+c)p_i'(x_j, t+1)$. Set $t \Leftarrow t+1$

STEP 3: If termination criteria are not met, go to STEP 1.

STEP 4: Generate N solutions according to $p(\mathbf{x}, t) = \prod_{i=1}^{n} p_i(x_i, t)$ and compute *max* $f(x)$ and *argmax* $f(\mathbf{x})$

DIFFOPT is not restricted to Wright's equation or DDRP. We propose a third one. Its rationale is as follows. From the analysis of UMDA we know that Wright's equation models proportionate selection. But this method converges very slowly when approaching the boundary. We have not been able to derive dynamic equations for truncation selection. Therefore we experimented with a number of faster versions of Wright's equation. The following difference equation was ultimately chosen.

Definition. $F-Wright(\alpha)$ (Fast Wright) is defined by the following difference equation:

$$p_i(x_i, t+1) = p_i(x_i, t) + sign(diff) * \exp(\alpha \ln abs(diff)) \tag{30}$$

$$diff = p_i(x_i, t)\,\frac{\partial\tilde{W}/\partial p_i(x_i) - \sum_{y_i \in \bar{A}_i} p_i(y_i, t)\partial\tilde{W}/\partial p_i(y_i)}{\tilde{W}(\mathbf{p})}. \tag{31}$$

If a value outside the interval $(p_{\min}, 1-p_{\min})$ is generated, we just set the value to the corresponding boundary value of the interval. For $\alpha = 1$ we obtain Wright's equation. We usually set $\alpha = 0.5$. The reason for this choice is that we wanted a difference equation which resembles as much as possible *truncation selection*. If we take the fitness function *OneMax*, we obtain for *F-Wright*(0.5) the difference equation

$$p(t+1) - p(t) = \sqrt{p(t)(1 - p(t)/(np(t)))} = \sqrt{(1 - p(t)/n)}. \tag{32}$$

This equation is similar to the approximate equation we have computed for UMDA with truncation selection. Only the multiplication by p is missing. This means that *F-Wright* will normally converge faster than UMDA with truncation selection.

We next evaluate the three difference equations with optimization problems.

8. Optimization of binary functions

The DDRP opens the possibility to use an arbitrary potential. If the potential is not a representation of the average fitness, Wright's equation and DDRP are different. We will demonstrate this with a simple example, a quadratic potential.

Example. $V(\mathbf{p}) = \sum_{ij} a_{ij} p_i(1) p_j(0) + c$

c is chosen such that $V(\mathbf{p}) > 0$. We make the assumption $a_{ii} = 0$. We obtain

$$\frac{\partial V}{\partial p_i(1)} = \sum_j a_{ij} p_j(0),$$

$$\frac{\partial V}{\partial p_i(0)} = \sum_j a_{ji} p_j(1),$$

$$V_i(\mathbf{p}) = p_i(1)\frac{\partial V}{\partial p_i(1)} + p_i(0)\frac{\partial V}{\partial p_i(0)}.$$

Obviously $\sum_i p_i(1) \sum_j a_{ij} p_j(0) = \sum_i p_i(0) \sum_j a_{ji} p_j(1)$. Therefore we obtain.

Proposition. *$V(\mathbf{p}) = 1/2 \sum_i V_i(\mathbf{p})$ if c is suitably chosen.*

The DDRP is given by

$$\Delta p_i(1) = p_i(1)\frac{\sum_{j \neq i} a_{ij} p_j(0) - V_i}{V_i + c_i},$$

c_i has to be chosen that $V_i(\mathbf{p})+c_i>0$. If we eliminate $p_i(0)=1-p_i(1)$ and abbreviate $p_i := p_i(1)$ we obtain

$$\Delta p_i = p_i(1-p_i)\frac{\sum_{j\neq i} a_{ij}(1-p_j)-\sum_{j\neq i} a_{ji}p_j}{V_i+c_i}. \tag{33}$$

We now determine Wright's equation for the same problem. This means we have to find a fitness function, which will give $V(\mathbf{p})=\tilde{W}(\mathbf{p})$.

Example. $f(x)=\sum_{ij} a_{ij}x_i(1-x_j)+c$

c is chosen such that $f(x)>0$. We compute $\tilde{W}(\mathbf{p})$ using our lemma

$$\tilde{W}(\mathbf{p}) = \sum_{ij} a_{ij}p_i(1-p_j)+c. \tag{34}$$

Obviously, $\tilde{W}(\mathbf{p})=V(\mathbf{p})$. Wright's equation is given by

$$\Delta p_i = p_i(1-p_i)\frac{\sum_{j\neq i} a_{ij}(1-p_j)-\sum_{j\neq i} a_{ji}p_j}{\tilde{W}(\mathbf{p})}. \tag{35}$$

We now compare the two difference equations. We assume that $c=c_i=0$ and obtain

$$\Delta p_i = p_i(1-p_i)\frac{\sum_{j\neq i} a_{ij}(1-p_j)-\sum_{j\neq i} a_{ji}p_j}{\tilde{W}(\mathbf{p})},$$

$$\Delta p_i = p_i(1-p_i)\frac{\sum_{j\neq i} a_{ij}(1-p_j)-\sum_{j\neq i} a_{ji}p_j}{p_j\sum_j a_{ij}(1-p_j)+(1-p_i)\sum_j a_{ji}p_j}.$$

The two difference equations differ in the denominator only. The denominator of DDRP is normally smaller than the denominator of Wright's equation. Thus DDRP will converge faster. We will compare three different examples.

Problem 1. $a_{i,i+1}=1$, $a_{i,i-1}=1$. All other values are set to 0.

The two global optima of this problem are $1,0,1,0,\ldots$ and $0,1,0,1,\ldots$ with a fitness value of $n-1$. The fitness function is symmetric. $f(x)$ and $f(\bar{x})$ have the same fitness value. $\bar{x}$ is the inverted x string. We have an unstable attractor at $p_i=0.5$.

Problem 2. $a_{i,i+1}=1$, $a_{i,i-1}=2$, $a_{n-1,n-2}=3$ All other values are set to 0.

Here the matrix a is not symmetric. The value $a_{n-1,n-2}=3$ deceives the system to set $x_{n-1}=1$. But the optimal solution is $x_{\max}=(0,1,0,1,..)$ with $x_{n-1}=0$ for n even. The optimum fitness value is $1.5n-1$.

Problem 3. $a_{i,j}=1$, $j<i$ All other values are set to 0.

Here the maximum is $x_{\max}=(0,0,..0,1..,1,1)$, i.e. the first half of the bits are 0, the second half of the bits are 1. For $n=30$ the optimal value is 225.

In Table 2, numerical results are displayed. For Problem 1 with $n=30$ the optimum is found at least once by all three methods. On the average one bit is wrong. This

Table 2
Numerical results (average over 10 runs). The number in brackets gives the number of times a global optimum has been found

Algorithm	Prob.	n	Iter.	Aver.	Maximum (S)
Wright	1	30	250	26.1	28.2(2)
DDRP	1	30	70	25.9	27.8(1)
F-Wright(0.5)	1	30	20	25.4	27.6(1)
Wright	1	60	500	52.2	55.6(0)
DDRP	1	60	140	53.3	55.6(0)
F-Wright(0.5)	1	60	20	52.8	54.5(0)
Wright	2	30	60	39.6	43.0(0)
DDRP	2	30	70	39.2	43.1(1)
F-Wright(0.5)	2	30	20	40.3	43.0(0)
Wright	2	60	500	85.2	88.0(0)
DDRP	2	60	50	83.3	87.7(0)
F-Wright(0.5)	2	60	20	85.3	88.0(0)
Wright	3	30	250	216.3	225.0(10)
DDRP	3	30	250	167.4	204.4(00)
F-Wright(0.5)	3	30	20	40.3	225.0(10)

behavior can be understood because of the parallel search and the symmetry of the problem. For $n = 60$ we have 3 bits wrong on the average. In problem 2 bit $n-1$ is always set to 1 (because of $a(n-1, n-3) = 3$). Therefore the optimum is missed, which has a 0 at this place. The same behavior is to be observed for $n = 60$. The optimum is missed by one point. A large difference in the performance can be seen for Problem 3. Here the results for the more local DDRP are really bad. DDRP is not able to set the bits correct in the area where all 1 meets all 0. This problem is the simplest for Wright's equation and F-Wright.

Taken all three examples together shows that *F-Wright*(0.5) is the fastest and most efficient algorithm.

In Table 3 numerical results for a microscopic algorithm (a genetic algorithm GA) and UMDA are shown. The results of UMDA with proportionate selection and Wright's equation are fairly similar. The results for Problem 2 are left out because they are similar to Problem 1. Note that no algorithm is able to locate the global optimum for Problem 1 with size $n = 60$. For this problem UMDA has to be extended to use more complex search distributions, which use conditional marginal distributions. The corresponding algorithm we call the factorized distribution algorithm, FDA [9].

9. Three royal roads to optimization

In this section we will try to classify the different approaches presented. Population search methods are based on two components at least—selection and reproduction with variation. In our research we have transformed genetic algorithms to a family

Table 3
Numerical results for UMDA with proportionate selection (p) and truncation selection (tr) and a genetic algorithm with uniform crossover (uc) and one-point crossover (1p).

Algorithm	Prob.	n	N	Iter.	Maximum (S)
UMDA p.	1	30	300	230	27.2(4)
UMDA tr.	1	30	300	90	26.9(2)
GA uc	1	30	300	100	27.4(1)
GA 1p	1	30	300	100	28.0(2)
UMDA p.	1	60	600	400	53.0(0)
UMDA tr.	1	60	600	150	53.3(0)
GA uc	1	60	600	150	55.3(0)
GA 1p	1	60	600	150	58.0(3)
UMDA p.	1	30	300	200	225.0(10)
UMDA tr.	3	30	300	10	225.0(10)
GA uc	3	30	300	30	225.0(10)
GA 1p	3	30	300	30	225.0(10)

of algorithms using search distributions instead of recombination/mutation of strings. The simplest algorithm of this family is the univariate marginal distribution algorithm UMDA.

Wright's equation describes the behavior of UMDA using an infinite population and proportionate selection. The equation shows that UMDA does *not* primarily optimize the *fitness function* $f(x)$, but the *average fitness* of the population $W(\mathbf{p})$ which depends on the continuous marginal frequencies $p_i(x)$. Thus the important landscape for population search is *not* the landscape defined by the fitness function $f(x)$, but the landscape defined by $W(\mathbf{p})$. In mathematical terms Wright's equation transforms the discrete optimization problem into a continuous one. Thus mathematically we can try to optimize $W(\mathbf{p})$ instead of $f(x)$. This later approach we (and other researcher) call the system dynamics approach to optimization.

The two components of population based search methods—selection and reproduction with variation—can work on a microscopic (individual) or a macroscopic (population) level. The level can be different for selection and reproduction. It is possible to classify the different approaches according to the level the components work. The following table shows three classes of evolutionary algorithms, each with a representative member.

Algorithm	Selection	Reproduction
Genetic Algorithm	microscopic	microscopic
UMDA	microscopic	macroscopic
System Dynamics	macroscopic	macroscopic

A genetic algorithm uses a population of individuals. Selection and recombination is done by manipulating individual strings. UMDA uses marginal distributions to create individuals. These are macroscopic variables. Selection is done on a population of individuals, as genetic algorithms do. In the system dynamics approach selection is modeled by a specific dynamic difference equation for macroscopic variables. We believe that a fourth class—macroscopic selection and microscopic reproduction—makes no sense.

Each of the approaches have their specific pros and cons. Genetic algorithms are very flexible, but the standard recombination operator has limited capabilities. UMDA can use any kind of selection techniques, which is used in genetic algorithm. UMDA be extended to an algorithm which uses a more complex factorization of the distribution. This is done by the factorized distribution algorithm FDA. Selection is very difficult to model on a macroscopic level. Wright's equation are valid for proportionate selection only. Other selection schemes lead to very complicated system dynamics equations.

Thus for proportionate selection and gene pool recombination all methods will behave similarly. But each of the methods allows extensions which cannot be modelled with an approach using a different level.

Mathematically especially interesting is the extension of UMDA to PDA with an adaptive Boltzmann annealing schedule. For this algorithm, convergence for a large class of discrete optimization problems can be shown. The interested reader is referred to [9,10].

References

[1] N.H. Barton, The divergence of a polygenic system subject to stabilizing selection, mutation and drift, Genet. Res. Camb. 54 (1989) 57–77.

[2] L.E. Baum, J.A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bull. Amer. Math. Soc. 73 (1967) 360–363.

[3] A.W.F. Edwards, Sewall Wright's equation, Theoret. Population Biol. 57 (2000) 67–70.

[4] R.A. Fisher, Average excess and average effect of a gene substitution, Ann. Eugenics 11 (1941) 53–63.

[5] J. Hofbauer, K. Sigmund, Evolutionary Games and Population Dynamics, Cambridge University Press, Cambridge, 1998.

[6] M.I. Jordan, Learning in Graphical Models, MIT Press, Cambrigde, MA, 1999.

[7] H. Mühlenbein, The equation for the response to selection and its use for prediction, Evolutionary Comput. 5 (3) (1997) 303–346.

[8] H. Mühlenbein, Th. Mahnig, Convergence theory and applications of the factorized distribution algorithm, J. Comput. Inform. Tech. 7 (1999) 19–32.

[9] H. Mühlenbein, Th. Mahnig, FDA—a scalable evolutionary algorithm for the optimization of additively decomposed functions, Evolutionary Comput. 7 (4) (1999) 353–376.

[10] H. Mühlenbein, Th. Mahnig, Evolutionary algorithms: from recombination to search distributions, in: L. Kallel, B. Naudts, A. Rogers (Eds.), Theoretical Aspects of Evolutionary Computing, Natural Computing, Springer, Berlin, 2000, pp. 137–176.

[11] H. Mühlenbein, Th. Mahnig, A. Rodriguez Ochoa, Schemata, distributions and graphical models in evolutionary optimization, J. Heuristics 5 (1999) 215–247.

[12] H. Mühlenbein, H.-M. Voigt, Gene pool recombination in genetic algorithms, in: J.P. Kelly, I.H. Osman (Eds.), Metaheuristics: Theory and Applications, Kluwer Academic Publishers, Dordrecht, 1996, pp. 53–62.

[13] H.-M. Voigt, Evolution and Optimization, Akademie-Verlag, Berlin, 1989.

[14] S. Wright, Evolution in mendelian populations, Genetics 16 (1931) 97–159.

[15] S. Wright, The analysis of variance and the correlations between relatives with respect to deviation from an optimum, J. Genetics 30 (1935) 243–256.

[16] S. Wright, The distribution of gene frequencies in populations, Proc. Nat. Acad. Sci. 24 (1937) 253–259.

[17] S. Wright, Stochastic processes in evolution, in: J. Gurland (Ed.), Stochastic Models in Medicine and Biology, Springer, Berlin, 1964.

[18] S. Wright, Random drift and the shifting balance theory of evolution, in: K. Kojima (Ed.), Mathematical Topics in Population Genetics, Univ. of Wisconsin Press, Wisconsin, 1970.