GENETIC ALGORITHMS: A SURVEY OF SOME MATHEMATICAL MODELS — PART I

Thomas Unger[1]

1. Introduction

Genetic Algorithms (GAs) are a special type of *evolutionary algorithms*, algorithms that simulate biological processes to solve search and optimization problems. They were introduced by John Holland in 1975 [H]. Given a specific problem, potential solutions are typically encoded as bit strings, constituting a *population.* The bit strings are allowed to *reproduce* on the basis of their *fitness*, thus forming a new population. Iterating this process, the population evolves according to a 'natural selection and survival of the fittest' process similar to the one described by Charles Darwin in *The Origin of Species.* If the GA is implemented successfully, the final population should consist of maximally fit individuals, approximating an optimal solution for the problem at hand.

The success or failure of a GA for a certain problem is strongly dependent on the encoding and several parameters that we will introduce later.

GAs have been implemented for a wide variety of problems, both real-world (e.g. scheduling electricity generation [A]) and abstract (e.g. solving NP-complete problems [D]). The bulk of the GA literature is concerned with practical applications. For a very complete bibliography, see [G2], which contains more than 4000 entries !

In this article we present mathematical models that have been introduced to study the behaviour of GAs. A lot of work in

[1]The author gratefully acknowledges financial support from Enterprise Ireland.

this area still has to be done, as a complete understanding of what happens does not exist yet.

2. Terminology

In this section we will give a brief overview of the GA terminology. For more detailed information, the reader can consult [B], [W], the classics [H] and [G1] or the more recent [M].

We will first introduce the relevant concepts as they appear in the literature. Some of them are usually not defined in a very (mathematically) rigorous way. For now we will follow this tradition, so that we can move on to the description of the GA as soon as possible. The reader need not panic however, we will make them precise in Section 5.

Consider the binary alphabet $\Sigma = \{0, 1\}$. (In general other alphabets can be used.) An ordered sequence

$$a = a_{\ell-1}a_{\ell-2}\cdots a_1 a_0,\ a_i \in \Sigma$$

is called an *individual* or a *chromosome* or a *string* of *length* ℓ with *genes* a_i at *locus* i. (In general genes can contain more than one letter of the alphabet. Chromosomes can be more complex, e.g. *diploid* — containing two sequences — instead of *haploid* — consisting of only one sequence.) Note that we read a string from right to left.

Fix ℓ and let $\Omega = \{0, 1\}^\ell$, the set of all possible length ℓ strings over Σ. A *population* of *size* n is a multi-set of n elements of Ω (i.e. a particular string can occur more than once).

Let P be a population of size n consisting of length ℓ strings. A *fitness function* is a map $f : P \to \mathbb{R}^+$, $x \mapsto f(x)$. We call $f(x)$ the *fitness (value)* of the string x. The fitness function is determined by the problem at hand. (For example in an optimization problem it can usually be taken to be the function one wants to optimize.) The choice of the fitness function is one of the factors that determines the success or the failure of the GA.

Selection is an operator that maps a string to multiple copies of itself according to its fitness value. *Crossover* is an operator that maps two strings (*parents*) to two new strings (*offspring*) and that is applied with a probability p_c on individual strings.

Several different crossover operators are in use. The simplest one is *one-point crossover:* given two length ℓ bit strings a and b, a locus i is selected randomly $(0 \leq i \leq \ell - 2)$ and two new strings are formed by swapping the substrings of a and b starting at locus $i+1$, again reading from right to left.

Mutation is an operator that is applied with a probability p_m (usually low, e.g. $p_m = 0.001$) to a string a by picking a locus $0 \leq i \leq \ell - 1$ randomly and replacing the bit a_i by $a_i + 1 \bmod 2$ (i.e. $0 \leftrightarrow 1$).

There are many varieties of selection, crossover and mutation. The particular varieties chosen and the fine-tuning of the probabilities p_c and p_m can have a big influence on the performance of the GA. Other operators have been defined in the literature.

3. The Simple Genetic Algorithm

All GAs can be viewed as modifications of a basic one, the Simple Genetic Algorithm (SGA), that we will describe in this section. Suppose that we are given a clearly defined problem, that candidate solutions are encoded as bit strings of length ℓ, that the population size is n and that a fitness function f is defined. Then the SGA consists of the following steps:

◇ Start with random population $P(0)$ of size n consisting of binary strings of length ℓ.

◇ Until the system stops improving, repeat the following procedure, starting with $t = 0$:

- Consider population $P(t)$.
- Calculate the fitness $f(i)$ of every string i in $P(t)$.
- **Selection.** Select n strings from $P(t)$ according to their relative fitness. These strings constitute an "intermediate population", called the **gene pool**.
- **Recombination.** Construct new population $P(t+1)$ as follows:
  As long as size of $P(t+1) < n$ repeat the following steps:
  - Randomly select 2 parents from the gene pool.

   - **Crossover.** Generate two offspring by means of one-point crossover. If no crossover takes place, form two offspring by cloning the parents.
   - **Mutation.** Mutate the offspring.
   - Place the resulting strings in $P(t+1)$.
- Increment t.

Each iteration of this process is called a *generation.* The entire set of generations is called a *run.*

4. Holland's Schema Theorem

The first attempt to explain rigorously the behaviour of GAs was made by John Holland [H]. We will briefly explain his idea. The set $\Omega = \{0,1\}^\ell$ can be considered as consisting of the vertices of an ℓ-dimensional cube. A given bit string x is an element of several hyperplanes in this ℓ-cube. Holland calls a hyperplane a *schema.* A schema H can contain several bit strings. The *average fitness* of a schema H is the average fitness of all $x \in H$. The idea is that at a given generation, while the GA is *explicitly* evaluating the fitness of the n strings in the population, it is *implicitly* estimating the average fitness of a much larger number of schemata. Holland calls this behaviour *implicit parallelism.*

Consider the alphabet $\Sigma' = \{0,1,*\}$. A schema can be viewed as an element of ${\Sigma'}^\ell$. For example, when $\ell = 3$, the strings 010 and 011 are both elements of the schema $01*$. The *order* $o(H)$ of a schema H is equal to the number of defined bits. (E.g. $1{*}0$ has order 2.) The *defining length* $\delta(H)$ of a schema H is the distance between the outermost defined bits. (E.g. $\delta(0*1) = 3-1 = 2$.)

The approximate dynamics of the increase and decrease in schema instances is described by the Schema Theorem, which roughly states that *short, low-order schemata with above average fitness will receive exponentially increasing numbers of samples over time.* It gives a lower bound on the expected growth of the number of instances of a schema from one generation to the next. Recently, Vose has argued that this theorem is not useful at all [V2].

5. The Infinite Population Model

In this section we will describe Vose and Liepins' formalization of the SGA. They model genetic search directly instead of looking at schemata as in Holland's model. This account is based on [V1], [N] and [M]. The SGA in this section is slightly different from the one presented in Section 3, in that after crossover only one of the two offspring is selected (at random) and the other one discarded. This modification simplifies parts of the formalization.

Again, let $\Omega = \{0,1\}^\ell$, the set of length-ℓ binary strings. Let $N = |\Omega| = 2^\ell$. We can view Ω as the set $\{0,\ldots,N-1\}$ by identifying bitstrings with their decimal value. We can also view Ω as $\mathbb{Z}/2\mathbb{Z} \times \cdots \times \mathbb{Z}/2\mathbb{Z}$, the product of ℓ copies of $\mathbb{Z}/2\mathbb{Z}$, the integers mod 2. This allows us to define two group operations on $\{0,\ldots,N-1\}$: the component-wise sum (denoted $\oplus$) on the product group acts as exclusive-or on $\{0,\ldots,N-1\}$ and the component-wise multiplication (denoted $\otimes$) acts as logical-and on $\{0,\ldots,N-1\}$.

Let n_i^t denote the number of instances of string i in the population at time t. Let $f : \Omega \to \mathbb{R}^+$ denote the fitness function. Suppose the population size is n.

We introduce a vector $p^t \in \mathbb{R}^N$ that represents the population at time t. Its components are defined as follows,

$$p_i^t = \frac{n_i^t}{n},$$

the proportion consisting of string i in the population at time t.

Another vector $s^t \in \mathbb{R}^N$ is defined by its components as follows,

$$s_i^t = \frac{f(i)n_i^t}{\sum_{j=0}^{N-1} f(j)n_j^t},$$

the probability that string i will be selected for recombination (i.e. selected for the gene pool that will be used to construct the population at time $t+1$).

Remarks.

- $p_i^t = s_i^t = 0$ when string i is not in the population at time t.

- p^t and s^t both have at most n non-zero entries.
- Since both p^t and s^t have non-negative entries that sum to 1, they are *stochastic* vectors.

Example. Suppose that $\ell = 2$ and the population consists of 2 copies of 11 and one copy each of 01 and 10. Then $p^t = (0, 0.25, 0.25, 0.5)$. Suppose that fitness is equal to the number of ones in the string. Then $f(00) = 0$, $f(01) = f(10) = 1$ and $f(11) = 2$. Hence $\sum_j f(j)n_j^t = 6$. Hence $s^t = (0, 1/6, 1/6, 4/6) = (0, 0.1667, 0.1667, 0.6667)$.

Under the assumption that the fitness function does not change during the evolution of the population, we have the following

Definition 5.1. The *selection operator* F is defined to be the $N \times N$ diagonal matrix with $F_{ii} = f(i)$, $\forall i \in \{0, \ldots, N-1\}$.

We define a relation $\sim$ on $\mathbb{R}^N \setminus \{0\}$ by $x \sim y$ iff $\exists \lambda > 0$ such that $x = \lambda y$. Clearly $\sim$ is an equivalence relation. Furthermore, for any equivalence class we can always find a representative with norm 1 ($y = x/||x|| \sim x$ and $||y|| = 1$).

Since selection is performed proportional to relative fitness, we expect the following to hold,

$$Fp^t \sim s^t.$$

Indeed, it is easy to show that

$$(Fp^t)_k = \overline{f} s_k^t, \ \forall k \in \Omega.$$

where

$$\overline{f} = \frac{1}{n} \sum_{j=0}^{N-1} f(j) n_j^t$$

is the average fitness of the population at time t.

In the GA literature the term *genetic operator* is often used without being clearly defined. This shortcoming is overcome by

Naudts [N] who formalizes the notion of a genetic operator as follows: consider a map

$$G : \Omega \times P_1 \times \cdots \times P_m \to \Omega$$

where $P_1, \ldots, P_m$ are m parameter sets. Usually $1 \leq m \leq \ell$ and $P_i = \{0, \ldots, \ell - 1\}$. Values for the parameters will be chosen at random immediately before the operator is applied.

Definition 5.2. A map G as defined above is a *genetic operator* acting on *one* string iff for each parameter tuple $(p_1, \ldots, p_m) \in P_1 \times \cdots \times P_m$,

$$G(\cdot, p_1, \ldots, p_m) : \Omega \to \Omega$$

is a bijection.

Next consider a map

$$C : \Omega^2 \times P_1 \times \cdots \times P_m \to \Omega^2$$

where the P_i's are again parameter sets. Let C_1 denote the first projection of C and C_2 the second.

Definition 5.3. A map C defined as above is a *genetic operator* acting on *a couple* of strings iff for each parameter tuple $(p_1, \ldots, p_m) \in P_1 \times \cdots \times P_m$,

(1) $C((\cdot,\cdot), p_1, \ldots, p_m) : \Omega^2 \to \Omega^2$ is a bijection;

(2) $C_1((i,j), p_1, \ldots, p_m) = C_2((j,i), p_1, \ldots, p_m),\ \forall (i,j) \in \Omega^2$.

Example. Crossover is a genetic operator $C : \Omega^2 \times P_1 \to \Omega^2$. When two strings $i = i_{\ell-1} \cdots i_1 i_0$ and $j = j_{\ell-1} \cdots j_1 j_0$ are selected, a crossover point p is selected at random in P_1, $0 \leq p \leq \ell - 1$ and crossover is applied to the strings i and j as follows,

$$C((\cdot,\cdot), p) : \Omega^2 \to \Omega^2,\ (i,j) \mapsto (i', j'),$$

where $i' = i_{\ell-1} \cdots i_{p+1} j_p \cdots j_1 j_0$ and $j' = j_{\ell-1} \cdots j_{p+1} i_p \cdots i_1 i_0$. If $p = \ell - 1$, no crossover occurs and C acts as the identity.

Remarks.

- If a genetic operator which normally acts on one string, is to act on a couple, it is defined to act *independently* on both components.
- The second definition may be generalized to genetic operators acting on tuples containing more than two strings.

Now the recombination process can be formalized as follows: as long as the new population is not full, two strings i and j are selected from the gene pool, and a finite sequence $(H_i)_{i=1}^h$ of genetic operators is applied to them,

$$(o_1, o_2) = H_1 \circ H_2 \circ \cdots \circ H_h(i, j),$$

resulting in a couple (o_1, o_2), called the *offspring* of (i, j). Then one of the two offspring is chosen with a probability of 0.5 to contribute to the next generation.

Definition 5.4. For every $k \in \Omega$ we define an $N \times N$ matrix $r(k)$ with components

$$r_{i,j}(k) = \mathbb{P}(k \text{ results from the recombination process based on parents } i \text{ and } j).$$

These matrices are called *recombination probabilities.* They have the following properties:

- $\sum_{k=0}^{N-1} r_{i,j}(k) = 1, \ \forall i, j \in \Omega;$
- $r_{i,j}(k) = r_{j,i}(k), \ \forall i, j, k \in \Omega.$

In what follows we will require that *recombination consists of a sequence of genetic operators which commute with group translation.* In other words:

- If a genetic operator X acts on one string only, we must have

  $$k \oplus l = X(i) \iff k = X(i \oplus l)$$

  for all parameters of X (left out here for convenience).

- If a genetic operator C acts on 2 strings, we must have for the projections C_1 and C_2 that

$$k \oplus l = C_m(i,j) \iff k = C_m(i \oplus l, j \oplus l) \quad (m = 1, 2).$$

It is easily observed that both one-point crossover and mutation commute with group translation.

The following result is now obvious,

Lemma 5.1. *The recombination probabilities satisfy*

$$r_{i,j}(k \oplus l) = r_{i \oplus l, j \oplus l}(k).$$

Definition 5.5. M is defined to be the $N \times N$ matrix having entries $m_{i,j} = r_{i,j}(0)$.

The following theorem shows that for the purpose of recombination it is sufficient to know the matrix M. That is why we call M a *mixing matrix.*

Theorem 5.1. *The matrix M determines the matrices $r(k)$, is nonnegative and symmetric and satisfies*

$$\sum_k m_{i \oplus k, j \oplus k} = 1, \ \forall i, j \in \Omega.$$

Proof: From the previous lemma we have that

$$r_{i,j}(k) = r_{i \oplus k, j \oplus k}(0) = m_{i \oplus k, j \oplus k}, \qquad \forall i, j, k \in \Omega$$

and the recombination probabilities sum to 1. M is nonnegative since its elements are probabilities and symmetric because the recombination probabilities are symmetric. ■

To get M positive rather than nonnegative we have to suppose that mutation is nonzero.

Definition 5.6. The *Walsh matrix* $W = (w_{i,j})$ is defined by

$$w_{i,j} = \prod_{k=1}^{\ell} r_{k(\lfloor i2^{\ell-k} \rfloor \bmod 2)}(j),$$

where the *Rademacher functions* $r_i : \Omega \to \{-1, 1\}$ are given by

$$r_i(x) = 1 - 2\left(\lfloor \frac{x2^i}{N} \rfloor \bmod 2\right).$$

The Walsh functions also map to $\{-1, 1\}$, are symmetric and orthogonal,

$$\sum_{k=0}^{\ell-1} w_{i,k} w_{j,k} = \begin{cases} N & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}.$$

Furthermore, the rows of the Walsh matrix are group characters,

$$w_{i \oplus j,k} = w_{i,k} w_{j,k}.$$

Vose and Liepins show that conjugation of the positive matrix M by W results in a sparse matrix.

Definition 5.7. The *twist* of M, denoted M_*, is defined as follows,

$$(M_*)_{i,j} = m_{i \oplus j, i}.$$

They also show that conjugation by W triangulates the twist M_* of M.

Lemma 5.2. *Let $\mathbb{E}$ denote expectation, then*

$$\mathbb{E}(p_k^{t+1}) = \sum_{i,j} s_i^t s_j^t r_{i,j}(k).$$

Proof: The expected proportion of string k in the next generation is computed by summing over all possible ways of producing k. If k results from reproduction based on parents i and j, then i (resp. j) is selected for reproduction with probability s_i^t (resp. s_j^t) and k is the result of recombination with probability $r_{i,j}(k)$. ■

If we now take the limit as population size $n \to \infty$, the *law of large numbers* gives us $p_k^{t+1} \to \mathbb{E}(p_k^{t+1})$.

Define permutations σ_j on $\mathbb{R}^N$ by

$$\sigma_j \langle s_0, \ldots, s_{N-1} \rangle^T = \langle s_{j \oplus 0}, \ldots, s_{j \oplus (N-1)} \rangle^T,$$

where vectors (between $\langle , \rangle$) are regarded as columns, and T denotes transpose.

Define the operator $\mathcal{M}$ by

$$\mathcal{M}(s) = \langle (\sigma_0 s)^T M \sigma_0 s, \ldots, (\sigma_{N-1} s)^T M \sigma_{N-1} s \rangle^T.$$

Theorem 5.2. $\mathbb{E}(s^{t+1}) \sim F\mathcal{M}(s^t)$.

Proof:

$$\begin{aligned}
\mathbb{E}(p_k^{t+1}) &= \sum_{i,j} s_i^t s_j^t r_{i,j}(k) \\
&= \sum_{i,j} s_i^t s_j^t r_{i \oplus k, j \oplus k}(0) \\
&= \sum_{i \oplus k, j \oplus k} s_{i \oplus k}^t s_{j \oplus k}^t r_{i,j}(0) \\
&= (\sigma_k s^t)^T M \sigma_k s^t
\end{aligned}$$

Since $s^{t+1} \sim F p^{t+1}$, the result follows. ■

The expected behaviour of a simple GA is therefore determined by two matrices: fitness information appropriate for selection is contained in F and M encodes mixing information appropriate for recombination.

Furthermore, the relation

$$s^{t+1} \sim F\mathcal{M}(s^t)$$

is an exact representation of the limiting behaviour as population size $n \to \infty$.

Based on the previous results Vose and Liepins formalize the SGA as follows:

Definition 5.8. *Simple genetic search* corresponds to the operator $\mathcal{G} = F \circ \mathcal{M}$, where F is the selection operator and M is any mixing matrix satisfying Theorem 5.2 and such that WM_*W is lower triangular. An initial population is modelled by a point $s^0 \in \mathbb{R}^N$, and the transition between generations is determined by $s^{t+1} \sim \mathcal{G}(s^t)$.

This formalization generalizes the recombination induced by mutation and one-point crossover, and regards GAs with finite populations as approximations to the ideal of simple genetic search.

Vose and Liepins give a geometric interpretation of simple genetic search by regarding the operator $\mathcal{G}$ as a map $\mathcal{G} : \mathcal{S} \to \mathcal{S}$, where $\mathcal{S}$ is the set of points with nonnegative coordinates of the unit sphere in $\mathbb{R}^N$, since every equivalence class of $\sim$ has a member of norm 1.

An initial population then corresponds to a point on $\mathcal{S}$, iterates of $\mathcal{G}$ are trajectories on $\mathcal{S}$ and convergence of the genetic algorithm corresponds to a fixed point of $\mathcal{G}$.

The general problem of finding the fixed points of $\mathcal{G}$ was not solved by Vose and Liepins. They did however study the fixed points of F and $\mathcal{M}$ separately. Fixed points of F (selection alone) correspond to populations that have completely converged to strings of *equal fitness*.

Only one class of these fixed points is *stable*: the set of fixed points corresponding to the *maximally fit* strings in the search space. So, we can interpret F as being a *focusing operator* that moves the population towards a state in which only the maximally fit individuals of the initial population are present.

Vose and Liepins then investigate the set of fixed points of $\mathcal{M}$, $\mathcal{M}_{\text{fixed}}$. They prove a sufficient condition for a fixed point to be an attractor:

Theorem 5.3. *Let $x \in \mathcal{M}_{\text{fixed}}$. If the matrix M is positive, then x is asymptotically stable whenever the second largest eigenvalue* *of M_* is less than $1/2$.*

They also determine the group of symmetries of $\mathcal{M}_{\text{fixed}}$:

Theorem 5.4. *For all j, and for every mixing matrix M, $\mathcal{M}(\sigma_j x) = \sigma_j \mathcal{M}(x)$. In particular, $\sigma_j \mathcal{M}_{\text{fixed}} = \mathcal{M}_{\text{fixed}}$, and $v = \langle N^{-1/2}, \ldots, N^{-1/2} \rangle \in \mathcal{M}_{\text{fixed}}$.*

This last theorem implies in particular that the dynamical system on $\mathcal{S}$ corresponding to $\mathcal{M}$ looks the same at each member of the population. In other words, $\mathcal{M}$ is a *diffusing operator.*

Based on these qualitative results they shed light on the phenomenon of *punctuated equilibria* that typically characterizes genetic search: relatively long periods of no improvement punctuated by quick rises in fitness. Intuitively they arise from the combination of the focusing properties of F and the diffusing properties of $\mathcal{M}$. Periods spent near one of the unstable fixed points of F correspond to *stasis* and the periods of rapid improvement can be accounted for by a movement (under the diffusing force of recombination) from the vicinity of one fixed point to another.

In the case of one-point crossover with mutation, Vose and Liepins calculate the matrix M explicitly. Its entries are equal to

$$m_{i,j} = \frac{(1-\mu)^\ell}{2} \left\{ \eta^{|i|} \left(1 - \chi + \frac{\chi}{\ell - 1} \sum_{k=1}^{\ell-1} \eta^{-\Delta_{i,j,k}} \right) + \eta^{|j|} \left(1 - \chi + \frac{\chi}{\ell - 1} \sum_{k=1}^{\ell-1} \eta^{\Delta_{i,j,k}} \right) \right\},$$

where χ is the crossover probability, μ the mutation probability, $\eta = \mu/(1-\mu)$, $|i|$ is the number of 1's in the bit string representation of the integer i and

$$\Delta_{i,j,k} = |(2^k - 1) \otimes i| - |(2^k - 1) \otimes j|.$$

On the basis of several computer runs calculating the spectrum of M_*, they find support for the following

Conjecture. *If $0 < \mu < 0.5$, then*

1. *The second largest eigenvalue of* M_* *is* $1/2-\mu$.
2. *The third largest eigenvalue of* M_* *is* $2\left(1-\frac{\chi}{\ell-1}\right)\left(\frac{1}{2}-\mu\right)^2$.

This conjecture was later proved by Koehler [K], who also showed:

Theorem 5.5. *The entire spectrum of* M_* *is given by*

$$\frac{1}{2}(1-2\mu)^{|i|}\left(1-\frac{\chi\,\mathrm{wid}(i)}{\ell-1}\right),\ i=0,\ldots,2^\ell-1,$$

where $\mathrm{wid}(i)$ *is the difference in position of the last 1 bit and the first 1 bit of* i. *If* i *has a single 1 bit or* $i=0$, *then* $\mathrm{wid}(i)=0$.

Theorems 5.3 and 5.5 imply that every fixed point of M is an attractor when $0<\mu<0.5$. Finally, Vose and Liepins also give a plausible argument for $v=\langle N^{-1/2},\ldots,N^{-1/2}\rangle$ (i.e. all possible strings represented equally) to be the unique fixed point of $\mathcal{M}$. This still has to be proved though.

* * *

In Part II, which will appear in the next issue of this Bulletin, we will discuss a finite population model and statistical mechanical approaches to modelling GAs.

References

[A] C. J. Aldridge, S. McKee and J. R. McDonald, *Genetic algorithm methodologies for scheduling electricity generation*, *in* Progress in Industrial Mathematics at ECMI 96, ed. by M. Brons, M. P. Bendsoe and M. P. Sorensen, 364-371. Teubner: 1997.

[B] D. Beasley, D. R. Bull and R. R. Martin, *An overview of genetic algorithms: Part 1, Fundamentals*, University Computing **15** (1993), 58-69.

[D] K. A. De Jong and W. M. Spears, *Using genetic algorithms to solve NP-complete problems*, *in* Proceedings of the Third International Conference on Genetic Algorithms, ed. by J. D. Schaffer. Morgan Kaufmann: 1989.

[G1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley: 1989.

[G2] D. E. Goldberg, K. Zakrzewski, B. Sutton, R. Gadient, C. Chang, P. Gallego, B. Miller and E. Cantú-Paz, Genetic Algorithms: A Bibliography, IlliGAL Report no. 97011, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997. Available on the web as

ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/97011.ps.Z

[H] J. Holland, Adaptation in Natural and Artificial Systems. MIT Press: 1975.

[K] G. J. Koehler, *A Proof of the Vose-Liepins Conjecture*, Ann. Math. Artificial Intelligence **10** (1994), 409-422.

[M] M. Mitchell, An Introduction to Genetic Algorithms. MIT Press: 1996.

[N] B. Naudts, Modelling Genetic Algorithms. Licentiate thesis, University of Antwerp–UIA, 1995.

[W] D. Whitley, *A Genetic Algorithm Tutorial*, Statistics and Computing **4** (1994), 65-85.

[V1] M. D. Vose and G. E. Liepins, *Punctuated equilibria in genetic search*, Complex Systems **5** (1991), 31-44.

[V2] M. D. Vose, A Critical Examination of The Schema Theorem, Technical Report ut-cs-93-212, University of Tennessee, Knoxville, 1993.

Thomas Unger
Department of Mathematics
University College Dublin
Belfield
Dublin 4
Ireland
e-mail: thomas.unger@ucd.ie