

New Stopping Criterion for Genetic Algorithms

Haldun Aytug
The Belk College of Business Adm.,
MIS/OM Dept., UNCC
9201 University City Blvd.
Charlotte, NC 28223

704-547-4429
704-547-3123 (FAX)
haytug@email.uncc.edu

Gary J. Koehler
Decision and Information Sciences, 351 BUS
The Warrington College of Business Administration
University of Florida
Gainesville, FL 32611

352-846-2090
352-392-5438 (FAX)
koehler@nervm.nerdc.ufl.edu
URL: <http://www.cba.ufl.edu/dis/faculty/koehler/>

December 4, 1996

New Stopping Criterion for Genetic Algorithms

Abstract

Genetic Algorithms have been successfully applied in a wide variety of problems. Although widely used, there are few theoretical guidelines for determining when to terminate the search. One result by Aytug and Koehler provides a loose bound on the number of GA generations needed to see all populations (and hence, an optimal solution) with a specified probability. In this paper we derive a tighter bound. This new bound is on the number of iterations required to achieve a level of confidence to guarantee that a Genetic Algorithm has seen all strings (and, hence, an optimal solution.)

Subject Areas: Genetic Algorithms, Stopping Condition, First Passage Times, Markov Chains.

1. Introduction

Genetic Algorithms (GAs) are general purpose search algorithms based on an evolutionary paradigm. Population members are represented by strings, corresponding to chromosomes. Search starts with a population of randomly selected strings, and, from these, the next generation is created by using genetic operators. At each iteration individual strings are evaluated with respect to a performance criteria and assigned a fitness value. Strings are randomly selected using these fitness values to either survive or to mate to produce children for the next generation. All such strings are subject to mutation. GAs have been used successfully in many domains (see Goldberg [3] for a nice overview).

All of the steps of a GA are well defined except the stopping criterion. Many practitioners use stopping rules like "stop when there is no significant improvement during the last ten iterations" or "stop after k generations" (where k is chosen to maximize the use of a reasonable level of resources available to the practitioner) or some such heuristic.

Aytug and Koehler [1] developed a theoretical bound on the number of iterations required to achieve a level of confidence that guarantees that a GA has found an optimal solution. This bound on GA running time is a function of only the mutation rate, size of the population strings and the population size (see their Corollary 2.12). Aytug, Bhattacharyya and Koehler [2] generalized this bound to non-binary alphabets.

The Aytug and Koehler [1] bound guarantees, with specified probability, that every population has been seen and hence every optimal solution. This is an overkill. They suggested the following possible future research.

"We believe tighter bounds could be derived by establishing a way where all population members (not all populations) are seen at least once with specified probabilities. For example, if after t iterations the sequence of populations seen so far covers all possible strings, then the search should be terminated since there is no more need to look at new populations. This would not necessitate seeing all populations with specified probability and hence should provide a better bound."

In this paper we develop such an approach and bound. Indeed, this new bound is significantly tighter than that in [1]. As in Aytug and Koehler [1], we restrict our attention to GAs operating on finite populations with binary strings. Our model is based

on the theoretical GA model given by Nix and Vose [5] and the concept of first passage time distributions of Markov Chains.

In Section 2 we review background material. In Section 3 we develop our model and derive a bound on the number of GA iterations sufficient to guarantee a desired confidence level. Finally, in Section 4 we present our conclusions and future research directions.

2. Background Material

2.1 GA Model

Let Ω be the set of binary strings of length $\ell > 1$ over which we wish to maximize a function $f()$ using a Genetic Algorithm (GA). We will limit our discussion to simple GA's involving one-point crossover and uniform mutation. $f()$ is assumed positive over the $r = 2^\ell$ points of Ω . Notation used throughout this paper is summarized in Table 1. The GA operates as given in Table 2.

<< insert Table 1 here >>

Algorithm 1: (*Genetic Algorithm*)

Given: String length ℓ , fitness function $f()$, mutation rate $\mu \in (0,1)$, one-point crossover rate $\chi \in (0,1]$ and population size $n \geq 1$.

Initialization: Generate an initial population, population 0. This is usually done by randomly drawing n strings from Ω with replacement.

Step 1: Form a new population as follows. Repeat the following steps until the new population has n members.

- (A) Randomly choose two members from the old population according to their fitness values relative to the sum of all fitness values of the current population. These are called parent strings.
- (B) Let $\text{RAN}(0,1)$ give a uniform random number between zero and one. If $\text{RAN}(0,1) \leq \chi$, derive two new strings by a random single-point crossover. Randomly choose one to add to the new population. Otherwise, randomly choose one of the parents to add to the new population.
- (C) Perform mutation on the new population member. For each bit, if $\text{RAN}(0,1) \leq \mu$, replace the bit by its complement.

Step 2: If stopping conditions are not met, return to Step 1.

Table 2: The Simple Genetic Algorithm

Using results from Vose [6], Nix and Vose [5] have derived a Markov Chain (MC) model for the expected behavior of this GA. The MC has states that represent the possible populations. Each state $i, i \in \{0, \dots, N-1\}$, indexes a vector, ϕ_i , of length r . Let e be a vector of ones of appropriate length and e' its transpose. The ϕ_i vectors are defined by

$$e' \phi_i = n,$$

$$(\phi_i)_j \in \{0, 1, \dots, n\} \quad j = 0, \dots, r-1$$

and

$$\phi_i \succ \phi_j \quad \text{if } i < j.$$

Here \succ means "lexicographically greater than." Nix and Vose [5] showed that

$$N = \binom{n + r - 1}{r - 1}.$$

Example: As an example, suppose $\ell = 2$ and $n = 2$. Then $r = 4$, $N = 10$ and the possible states are:

State ID:	0	1	2	3	4	5	6	7	8	9
State Vector:	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9
	2	1	1	1	0	0	0	0	0	0
	0	1	0	0	2	1	1	0	0	0
	0	0	1	0	0	1	0	2	1	0
	0	0	0	1	0	0	1	0	1	2.

Let $Z' = (\phi_0, \phi_1, \dots, \phi_{N-1})$ be the matrix of the ϕ vectors and let B be the matrix formed from Z by

$$B_{i,j} = \delta(Z_{i,j} \neq 0)$$

where

$$\delta(x) = \begin{cases} 0 & x \text{ false} \\ 1 & x \text{ true} \end{cases}.$$

The probability of transition from population i to population j is given by (see [5]):

$$P_{i,j} = n! \prod_{g=0}^{r-1} \frac{q_{i,g}^{Z_{j,g}}}{Z_{j,g}!}$$

where

$$q_{i,g} = \mathcal{M}\left(\frac{F\phi_i}{e'F\phi_i}\right)_g$$

and F is a diagonal matrix having values $f(0), f(1), \dots, f(r-1)$ along the diagonal. Operator $\mathcal{M}(\cdot)$ of an r dimensional vector is defined by (see [6])

$$\mathcal{M}(x) = \begin{pmatrix} (\sigma_0 x)' M \sigma_0 x \\ \cdot \\ \cdot \\ \cdot \\ (\sigma_{r-1} x)' M \sigma_{r-1} x \end{pmatrix}$$

where the permutation of x , $\sigma_k x$, is defined by

$$\sigma_k X = \begin{pmatrix} X_{k \oplus 0} \\ \cdot \\ \cdot \\ \cdot \\ X_{k \oplus (r-1)} \end{pmatrix}.$$

Here \oplus is the exclusive-or operator. Finally, matrix M is defined by (see [6])

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

with

$$\eta = \frac{\mu}{1-\mu}$$

and

$$\Delta_{g,k,h} = \left| (2^h - 1) \otimes g \right| - \left| (2^h - 1) \otimes k \right|$$

where $|i|$ is the number of non-zero bits of i , $i \in \Omega$, and \otimes means bitwise logical-and.

Under our assumptions on μ , χ , and $f()$, the transition matrix, P , is completely ergodic.

2.2 Stopping Criteria

Aytug and Koehler [1] derived a worst case bound on a GA's running time. A confidence probability, $\alpha \in [0,1)$ is chosen by the user. Aytug and Koehler [1] showed that the minimum number of GA generations necessary to see all populations (and, hence, an optimal solution) with probability α is bounded from above by \bar{t} given in their Corollary 2.12. This is summarized below.

Corollary 2.1: (*Aytug and Koehler [1]*)

A sufficient number of iterations required to find an optimal solution with a desired level of confidence α is bounded above by

$$\bar{t} = \left\lceil \frac{\ln(1-\alpha)}{\ln\left(1 - \min\{(1-\mu)^{\ell_n}, \mu^{\ell_n}\}\right)} \right\rceil.$$

Notice that this bound is independent of the crossover rate and the fitness function.

We now develop a tighter bound. The main improvement is that instead of seeing all populations, we will seek a bound on seeing all strings (with specified probability).

3. A New Worst-Case Bound

We start our GA with an initial population, say population h . The probability that h contains string i is merely $e_h' Be_i$. Either string i is in population h or not. If state h does not contain string i , then the probability of first seeing i after one transition is equal to the sum of transition probabilities from state h to states containing string i . Likewise, the probability of first seeing string i starting in state h after t transitions (i.e., after t new generations) is equal to the sum of probabilities of going through $t-1$ states not containing string i to a state containing string i . Let e_i be the i^{th} unit vector and $d(x)$ be the diagonal matrix formed by using vector x as the diagonal. In matrix notation, the vector of first passage probabilities to string i in $t > 0$ transitions is

$$FP_{i,t} = d(e - Be_i) [Pd(e - Be_i)]^{t-1} PBe_i. \quad (1)$$

The first term has zeros on the diagonal for states containing string i and ones otherwise. The second term gives $k-1$ step transition probabilities through states not containing string i . The final term gives one-step transition probabilities to states containing string i . Clearly

$$FP_{i,0} = Be_i.$$

Let

$$D_i = d(e - Be_i).$$

Then Equation 1 can be written more succinctly as

$$FP_{i,t} = [D_i P]^t Be_i.$$

Thus, the vector of expected first passage probabilities to states containing string i in exactly t or fewer generations is

$$\begin{aligned} \text{SFP}_{i,t} &= \sum_{k=0}^t \text{FP}_{i,t} = \sum_{k=0}^t [D_i P]^k \text{Be}_i . \\ &= \left(I - [D_i P]^{t+1} \right) \left(I - [D_i P] \right)^{-1} \text{Be}_i \\ &= \left(I - [D_i P]^{t+1} \right) e \end{aligned}$$

since

$$\left(I - [D_i P] \right)^{-1} \text{Be}_i = e$$

and

$$\sum_{k=0}^t [D_i P]^k = \left(I - [D_i P]^{t+1} \right) \left(I - [D_i P] \right)^{-1} .$$

Note that $\left(I - [D_i P] \right)$ is easily shown to be non-singular and that $\left(I - [D_i P] \right)^{-1} \text{Be}_i = e$ can be seen by multiplying through by $\left(I - [D_i P] \right)$ and simplifying.

Let a confidence probability, $\alpha \in [0,1)$, be specified and consider the following problem:

Problem 1:

$$\begin{aligned} t^* &= \max_i \min t \\ \text{Subject to} \\ \sum_{k=0}^t [D_i P]^k \text{Be}_i &\geq \alpha e \end{aligned}$$

Problem 1 gives the minimum number of transitions from population zero necessary to see all strings regardless of the starting state. As such, t^* provides an upper bound on the number of generations needed to find an optimal solution with probability α . The proof

for the following Lemma is virtually identical to that given for Lemma 2.5 in [1] and will be stated without proof. The constraint of Problem 1 can be simplified as shown in the following result

Lemma 3.1 (*Equivalent Constraint*)

t satisfies

$$\sum_{k=0}^t [D_i P]^k B e_i \geq \alpha e$$

if and only if

$$(1 - \alpha)e \geq [D_i P]^{t+1} e \quad (2)$$

The following Lemma will prove useful in further simplifying Equation 1.

Lemma 3.2: (*Aytug and Koehler [1]*)

Let Q be a non-negative matrix where $Qe > 0$. Then

$$\left(\max_i e_i' Q e \right)^k \geq \max_i e_i' Q^k e$$

for $k \geq 1$.

A number of the rows of Equation 2 are trivially satisfied, namely the rows corresponding to states having string i . Furthermore, using Lemma 3.2 requires us to ignore these rows. Towards this end, consider the following equivalent reduced system. The right side of Equation 2 can be written as

$$[D_i P]^{t+1} e = D_i [P D_i]^t P e$$

so Equation 2 becomes

$$(1 - \alpha)e \geq D_i [P D_i]^t e.$$

Rows representing states containing string i are always satisfied since their right hand sides have zeros. Also, $[PD_i]^t$ has zero columns corresponding to states containing string i . Let $D_i(0)$ be the matrix formed from D_i by removing the zero rows. Thus, the following is equivalent to Equation 2

$$(1 - \alpha)e \geq [D_i(0)PD_i(0)]^t e. \quad (3)$$

Note that $D_i(0)PD_i(0)' > 0$. Now we can apply Lemma 3.2 to show that Problem 1 is bounded as follows.

Theorem 3.3: (*Bound on t^**)
 t^* is bounded by

$$t^* \leq \bar{t} \equiv \left\lceil \frac{\ln(1 - \alpha)}{\ln\left(1 - \min_{i,j} e_j'(I + D_i P)Be_i\right)} \right\rceil.$$

Proof:

Consider

$$(1 - \alpha)e \geq [D_i(0)PD_i(0)]^t e.$$

From Lemma 3.2 we have

$$\left(\max_j e_j' D_i(0)PD_i(0)' e \right)^t \geq e_j' [D_i(0)PD_i(0)]^t e.$$

Hence

$$(1 - \alpha) \geq \left(\max_j e_j' D_i(0)PD_i(0)' e \right)^t \geq e_j' [D_i(0)PD_i(0)]^t e$$

provides a bound on t . Taking logs of the first two terms, solving for t and rounding up gives a lower bound for a given i . Choosing the largest such bound provides a sufficient condition for satisfying Equation 3, and hence, Equation 1 (using Lemma 3.1). This gives

$$t^* \leq \bar{t} \equiv \left\lceil \max_i \frac{\ln(1-\alpha)}{\ln\left(\max_j e_j' D_i(0) P D_i(0)' e\right)} \right\rceil.$$

At this point, it is convenient to note that

$$\begin{aligned} & \max_i \frac{\ln(1-\alpha)}{\ln\left(\max_j e_j' D_i(0) P D_i(0)' e\right)} \\ &= \ln(1-\alpha) \min_i \frac{1}{\ln\left(\max_j e_j' D_i(0) P D_i(0)' e\right)} \\ &= \frac{\ln(1-\alpha)}{\ln\left(\max_{i,j} e_j' D_i(0) P D_i(0)' e\right)} \end{aligned}$$

and that

$$\begin{aligned} \max_{i,j} e_j' D_i(0) P D_i(0)' e &= \max_{i,j} e_j' D_i P D_i e = \max_{i,j} e_j' D_i P (e - B e_i) \\ &= \max_{i,j} e_j' D_i (e - P B e_i) = \max_{i,j} e_j' (e - B e_i - D_i P B e_i) = 1 - \min_{i,j} e_j' (I + D_i P) B e_i. \end{aligned}$$

□

The following Lemma provides a closed form expression for the min in Theorem 3.3.

Lemma 3.4:

$$\min_a (P B e_i)_a = \begin{cases} 1 - (1 - \mu^\ell)^n & \mu \leq 0.5 \\ 1 - (1 - (1 - \mu)^\ell)^n & \mu \geq 0.5 \end{cases}.$$

Proof:

First we show that

$$\left(\text{PBe}_i\right)_a = 1 - (1 - q_{a,i})^n.$$

We have that

$$\left(\text{PBe}_i\right)_a = 1 - \sum_{j \ni Z_{j,i}=0} P_{a,j} = 1 - n! \sum_{j \ni Z_{j,i}=0} \prod_h \frac{q_{a,h}^{Z_{j,h}}}{Z_{j,h}!}.$$

Using the multinomial theorem [4] we get

$$n! \sum_{j \ni Z_{j,i}=0} \prod_h \frac{q_{a,h}^{Z_{j,h}}}{Z_{j,h}!} = (1 - q_{a,i})^n.$$

So, given that

$$\left(\text{PBe}_i\right)_a = 1 - (1 - q_{a,i})^n$$

then

$$\min_a \left(\text{PBe}_i\right)_a = \min_a \left[1 - (1 - q_{a,i})^n\right] = 1 - (1 - \min_a q_{a,i})^n.$$

Note that $q_{a,i}$ is equivalent to

$$\left(\sum_{m=0}^{r-1} f(m)Z_{a,m}\right)^{-2} \sum_{m=0}^{r-1} \sum_{s=0}^{r-1} f(m)Z_{a,m} f(s)Z_{a,s} M_{m \oplus i, s \oplus i}.$$

Aytug and Koehler [1] have shown that the minimum element of $q_{a,i}$ (over a and i) is μ^ℓ for $\mu \leq 0.5$ and $(1 - \mu)^\ell$ for $\mu \geq 0.5$. This value (for $\mu \leq 0.5$) is attained by creating string 0 from parent strings $r-1, r-1$ or, equivalently, by creating i from parents k and k where $k \oplus i = r-1$. Consequently, for a given i one can always find a state a such that $Z_{a,k} = n$ where k is chosen to satisfy $k \oplus i = r-1$. There is always such a k since Ω omega is closed under \oplus . Then, for any given i there will be a corresponding state a which includes the minimum of $q_{a,i}$. The result for $\mu \geq 0.5$ is similarly obtained. Substituting these yields the desired result.

Combining the results of Theorem 3.3 and Lemma 3.4 gives our main result.

Corollary 3.5: (*Bound on t^**)

t^* is bounded by

$$t^* \leq \bar{t} \equiv \left\lceil \frac{\ln(1-\alpha)}{n \ln \left(1 - \min \left[\mu^\ell, (1-\mu)^\ell \right] \right)} \right\rceil.$$

Notice that this bound tightens as the population size increases, as the string size decreases, and as the confidence probability decreases - all of which satisfies our intuition on a GA's running time. Perhaps not so obvious is that the bound tightens as the mutation rate increases (up to 0.5 and decreases there on). The worst-case number of function evaluations is $n\bar{t}$ which is independent of n . The following table gives runtime bounds for various parameter settings.

String Size	Population Size	Alpha	Mutation Rate	Bound
20	100	0.95	0.2	2.87055E+12
20	100	0.95	0.3	859166582
20	100	0.95	0.4	2724603
20	100	0.95	0.45	258377
20	100	0.95	0.5	31413
20	200	0.95	0.2	1.43527E+12
20	200	0.95	0.3	429583291
20	200	0.95	0.4	1362302
20	200	0.95	0.45	129189
20	200	0.95	0.5	15707

It is also useful to compare our new bound with that discovered in Aytug and Koehler [1]. From Corollary 2.1 we have the old bound of

$$\left\lceil \frac{\ln(1-\alpha)}{\ln \left(1 - \min \left\{ (1-\mu)^{\ell_n}, \mu^{\ell_n} \right\} \right)} \right\rceil.$$

Clearly since $(1 - \mu^{\ell n}) \geq (1 - \mu^{\ell})^n$ the new bound is tighter. The following table gives some comparisons showing how dramatic the improvement of the new bound.

String Size	Population Size	Alpha	Mutation Rate	New Bound	Old Bound	New/Old
5	2	0.95	0.2	4681	29255197	0.00016
5	2	0.95	0.3	616	507329	0.001214
5	2	0.95	0.4	146	28569	0.00511
5	2	0.95	0.45	81	8797	0.009208
5	2	0.95	0.5	48	3067	0.01565
7	3	0.95	0.2	78014	1.4202E+15	5.49E-11
7	3	0.95	0.3	4566	2.8639E+11	1.59E-08
7	3	0.95	0.4	609	681150661	8.94E-07
7	3	0.95	0.45	267	57416961	4.65E-06
7	3	0.95	0.5	128	6282505	2.04E-05

It is interesting to note when \bar{t} is minimal. Without loss of generality, we can restrict our attention to $\mu \leq 0.5$ and solve

$$\min \bar{t} = \frac{\ln(1 - \alpha)}{n \ln(1 - \mu^{\ell})}$$

subject to

$$0 \leq \mu \leq 0.5$$

$$1 \leq n \leq r, n \text{ integer.}$$

Since \bar{t} is decreasing in increasing n and increasing μ we have a minimum value of

$$\frac{\ln(1 - \alpha)}{r \ln(1 - 0.5^{\ell})}.$$

Aytug and Koehler determined that the minimal value of the bound in Corollary 2.1 is

$$\frac{\ln(1 - \alpha)}{\ln(1 - 0.5^{\ell})}.$$

So the new bound's minimum value is $2^{-\ell}$ times the minimum value of the bound in Corollary 2.1.

4. Summary and Future Directions

Using the Vose [6] and Nix and Vose [5] model of simple GAs, a bound for the minimal number of iterations required to see all strings and, hence, an optimal solution, with a specified level of confidence has been derived (Corollary 3.5.) The upper bound attains its minimum at $\mu=0.5$ and $n=r$. This minimal value is $2^{-\ell}$ times the minimum value of the bound found by Aytug and Koehler [1]. The bound obtained in Corollary 3.5 does not take into consideration the fitness or the crossover rate. It is a function of n , μ , ℓ , and the confidence level, α , only. (This results from using the relaxation in Lemma 3.2.) This bound tightens as the population size increases, as the string size decreases, as the confidence probability decreases and as the mutation rate increases. The worst-case number of function evaluations depends only on μ , ℓ , and the confidence level, α .

Although the bound found in this paper is an improvement over the bound in [1], it is still loose and many approaches can be pursued to find better bounds. In this work we sought a worse-case bound on seeing all strings with a given confidence independent of a starting state. One improvement might be to specify a starting state that is better than the worse-case state. Another improvement might be to restrict the target strings to a subset of all the strings, possibly by using outside knowledge about the fitness function. Yet another improvement might come from using a tighter result than that provided by Lemma 3.2.

Another avenue of research might be to obtain bounds on the average time rather than a bound on the time required to obtain a specified confidence. The average time to see a given string is intimately related to the first passage probabilities.

Yet another approach would be to develop bounds on average (as opposed to worse-case) performance. Prior distributions reflecting the generation of an initial population by randomly drawing n strings from Ω with or without replacement are easily specified.

Bibliography

- [1] Aytug, H. and G. J. Koehler, "Stopping Criteria for Finite Length Genetic Algorithms," **ORSA Journal on Computing**, 8, No. 2, 1996, pp. 183-191.
- [2] Aytug, H., S. Bhattacharyya and G. J. Koehler, "A Markov Chain Analysis of General Cardinality Genetic Algorithms with Power of 2 Cardinality Alphabets," forthcoming , " **European Journal of Operational Research**
- [3] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1989, Addison Wesley, Reading, MA.
- [4] Johnson, N. L., A. Kotz and A. W. Kemp, Univariate Discrete Distributions, 1992, John Wiley and Sons, Inc., New York.
- [5] Nix, A. and M. D. Vose, Modeling Genetic Algorithms with Markov Chains," **Annals of Mathematics and Artificial Intelligence**, 5, 1992, pp. 79-88.
- [6] Vose, M. D., "Formalizing Genetic Algorithms", **Proceedings of the IEEE Workshop on Genetic Algorithms, Neural Networks, and Simulated Annealing Applied to Signal and Image Processing**, Glasgow, Scotland, May, 1990.

\oplus	bitwise exclusive-or operator.
\otimes	bitwise and operator.
$ x $	number of nonzero bits of x .
μ	uniform mutation rate with $\mu \in (0,1)$.
η	$\mu / (1 - \mu)$.
χ	one-point crossover rate with $\chi \in (0,1]$.
ℓ	string length.
r	number of strings, $r = 2^\ell$.
Ω	set of all strings, $\Omega = \{0,1,\dots,r-1\}$.
n	population size.
x'	transpose of vector/matrix x .
ϕ_i	r -vector of string frequencies in population i .
N	number of possible populations.
Z	$N \times r$ matrix of ϕ_i vectors. $Z' = [\phi_0, \phi_1, \dots, \phi_{N-1}]$.
B	$N \times r$ matrix formed by setting nonzero components of Z to one.
M	$r \times r$ mixing matrix.
$\delta(x)$	is 0 if x is false, 1 otherwise.
σ_i	i^{th} permutation matrix with $(\sigma_i)_{j,k} = \delta(j = i \oplus k)$.
$\mathcal{M}(\)$	r -vector recombination operator.
$f(i)$	fitness of string i .
F	focusing matrix, a diagonal matrix with $F_{i,i} = f(i)$.
q	$N \times r$ matrix with elements $q_{i,g} = \mathcal{M}(F\phi_i / e'F\phi_i)_g$.
P	transition matrix over state space of populations.
$d(x)$	a diagonal matrix formed using vector x as the diagonal.
e	vector of ones.
e_i	i^{th} unit vector.
D_i	$d(e - Be_i)$.
$D_i(0)$	Matrix formed from D_i by removing rows having all zeros.
$FP_{i,t}$	probability of expected first passage to states containing i in exactly t generations.
$SFP_{i,t}$	probability of expected first passage to states containing i in t generations.

Table 1: Summary of Notation