NEAL'S VIEW OF EVOLUTIONARY FAMILY OF ALGORITHMS

- ANT SYSTEMS ARE PART OF THE FAMILY
- ALL THESE ALGORITHMS BUILD "MODELS"
  - POPULATION OR DISTRIBUTION OF SOLUTIONS
  - SELECT FROM MODEL
    - ALTER IN DIVIDUAL
    - TEST FITNESS
    - UPDATE THE MODEL

FAMILY TREE
"FROM THEORY PERSPECTIVE"

- GAS, HOLLAND, OTHERS
- EVOLUTIONARY STRATEGIES, RECHENBERG, SCHWEFEL
- EVOLUTIONARY PROGRAMMING, FOGEL

EDA
ACO
PSO
"ANTS"
"ANTS"
GENETIC PROGRAMMING
LCS

ALL OF THESE ALGORITHMS CAN BE MODELED WITH THE "VOSE" INFINITE POPULATION MODEL

- MICHAEL VOSE, ALDEN WRIGHT © 1998 MONTANA MAJOR CONTRIBUTORS
6.2 Limiting Distribution of the Simple GA

We shall now apply these ideas to the study of Genetic Algorithms.

3. If not, is the transition matrix primitive (and if so, what is its limiting distribution?)

2. If so, how long does it take to reach them?

1. Are there any absorbing states?
where the matrix \( A \) is denoted by \( A \) by replacing the column of

\[
\begin{bmatrix}
1 - n & 0 \\
1 & -1
\end{bmatrix}
\]

with zeros.

Theorem 4.1: Let \( \{X_n\} \) be a random walk on the set \( \{0, 1, 2, ..., n\} \) with transition kernel \( P \). Suppose \( n \geq 2 \) and \( \alpha \) is the primitive matrix of

\[
\begin{bmatrix}
1 - n & 0 \\
1 & -1
\end{bmatrix}
\]

Then the distribution of \( X_n \) is given by

\[
\frac{\alpha^n}{\alpha^n \cdot \alpha^0} = \alpha^n 
\]

and therefore the two-operation transition matrix is given by

\[
\begin{bmatrix}
\alpha^n \\
\alpha^n \cdot \alpha^0
\end{bmatrix}
\]

where \( \alpha \) is the primitive matrix of

\[
\begin{bmatrix}
1 - n & 0 \\
1 & -1
\end{bmatrix}
\]

Thus, if \( \alpha \) is a non-zero vector, then the probability of \( X_n = x \) is

\[
\frac{\alpha^n}{\alpha^n \cdot \alpha^0} = \alpha^n 
\]

Now suppose mutation is defined in such a way that \( Q < 0 \) for all \( i \neq j \) and \( Q = 0 \) for all \( i = j \).

Then the limiting distribution of the simple GA

\[
\begin{bmatrix}
\alpha^n \\
\alpha^n \cdot \alpha^0
\end{bmatrix}
\]

is given by

\[
\begin{bmatrix}
\alpha^n \\
\alpha^n \cdot \alpha^0
\end{bmatrix}
\]

and the probability of \( X_n \) is

\[
\begin{bmatrix}
\alpha^n \\
\alpha^n \cdot \alpha^0
\end{bmatrix}
\]

Now, recall that having a 1 in the diagonal of the transition matrix

\[
\begin{bmatrix}
\alpha^n \\
\alpha^n \cdot \alpha^0
\end{bmatrix}
\]

CHAPTER 6: GAS AS MARKOV PROCESSES
function is

\[ \frac{df}{d(f)} \]

that acts on the simplex then

\[ V \leftarrow V : f \]

We can write this more compactly if we view the fitness function as a vector

\[ \left( \begin{array}{c} 0 \end{array} \right) \]

and the selection as an operator on a vector

\[ \left( \begin{array}{c} d \end{array} \right) \]

We write this in terms of the population vector and denominator of the selection operator on the population size \( N \) and calculate as follows:

\[ \left( \begin{array}{c} f(a) \end{array} \right) \]

where \( N \) is the population size vector. Dividing the numerator

\[ \left( \begin{array}{c} 1 \end{array} \right) \]

by the denominator gives the probability of any individual \( i \), being selected is

\[ \frac{df_i}{d(f)} \]

We have previously seen that if we use proportional selection, the probability

6.2 Selection

for selection, mutation, and crossover.

We shall now look at the equations for \( d \) and explain the stochastic behavior of the population.

4. Relating the dynamics to the stochastic behavior of the population.

5. Studying the dynamics of \( d \). We shall be interested in fixed points,

\[ 3 = (z) f \]

\[ 1 = (1) f \]

\[ 2 = (0) f \]
6.3 MUTATION

Any non-decreasing probability density function in the integral

This method of defining rank-based selection can be generalized by using

And so that

where

The table shows how the rank probabilities are calculated:

\[
\begin{array}{cccccccc}
\text{rank} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{rank probability} & 0.25 & 0.25 & 0.1 & 0.05 & 0.05 & 0.02 & 0.01 \\
\end{array}
\]

Figure 6.5: Probability density function \(f(x)\) corresponding to the particular set of rank probabilities.
6.4 CROSSOVER

If we combine crossover with mutation we obtain the mixing operator

\[ \frac{d(M \delta|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda)}{d((f)\delta)|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda)} = \psi(d)\delta \]

and so we obtain

\[ \frac{d(M \delta|\Lambda^2W_\Lambda \Lambda)}{d((f)\delta)|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda)} = \left( \frac{d(M \delta|\Lambda^2W_\Lambda \Lambda)}{d((f)\delta)|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda)} \right) \Lambda_\Lambda \]

From this it follows that, for proportional selection

\[ (x) \Lambda_\Lambda \delta \psi = (x \delta \psi) \Lambda_\Lambda \]

vector \( x \) and scalar \( \psi \) and of the properties of quadratic operators such as \( \Lambda_\Lambda \delta \psi \) is that, for any

\[ \frac{d(M \delta|\Lambda^2W_\Lambda \Lambda)}{d((f)\delta)|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda)} = (x) \Lambda_\Lambda \delta \psi \]

If selection is proportional to fitness then as we have seen,

\[ \Lambda_\Lambda \delta \psi = \delta \]

the mixing operation for the reproduction equation.

The full sequence of selection, mutation and crossover gives us the combination of the structure of the search space will be described in the following section.

where \( \Lambda_\Lambda \delta \psi \) is the mutation matrix. Some of the properties of this matrix are

\[ d((f)\delta)|\Lambda^2W_\Lambda \Lambda(f)|\Lambda^2W_\Lambda \Lambda) = \psi(d)\Lambda_\Lambda \]

The mixing operator also turns out to be a quadratic operator afterwords \[299\]. The mixing operator is applied before crossover or selection, it does not matter whether mutation is applied before crossover or not. If the mixing operator is applied first the standard binary representation (\( f \) or \( f' \)) is then the probability that the mixing operator is applied to parents before crossover. For a large class of operators with the integers \( \{0,1\} \) associate the search space with \( \{1,1,0,0,0,0,0,0\} \) and \( 0 \) is the only option. The following table gives the different values

\[ \begin{array}{cccccccc}
0 & 0 & 0 & 3/1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 6/1 & 0 & 2 & 2 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
0 & 6/1 & 0 & 2 & 1 & 1 & 1 & 1 \\
\end{array} \]
Theorem 4.1 (No free lunch) For any set of admissible functions \( f \) and \( \mathcal{F} \),

\[ \int_{\mathcal{F}} f[2]d\mu \leq \int_{\mathcal{F}} f[2]d\nu \]

Theorem 4.1: **No Free Lunch** for any set of admissible functions. 

When the sum is over all possible functions \( f \),

\[ \int_{\mathcal{F}[2]} f[2]d\mu \leq \int_{\mathcal{F}[2]} f[2]d\nu \]

Theorem 4.1: **No Free Lunch** for any set of admissible functions. 

When the sum is over all possible functions \( f \),

\[ \int_{\mathcal{F}[2]} f[2]d\mu \leq \int_{\mathcal{F}[2]} f[2]d\nu \]