EA theory

In general: there is not much (not enough)
Often, either the
- Problem (fitness function), or
- The algorithm

is oversimplified to make the case tractable

In this lecture:
- Schema theorem for GAs
- Building block hypothesis & implicit parallelism for GAs
- Almost sure convergence for any EA
- No Free Luch Theorem (NFL) for any EA

Schemata (1)

Schema (definition):
A schema H in IB^ℓ is a *partial* instantiation of a string in IB^ℓ . Usually the uninstantiated elements are denoted by '$*$', sometimes called "don't care" symbol or "wild card". A schema defines a subset of IB^ℓ :

$$H \in \{0,1,*\}^\ell$$

Example:
H = {1, 0, 0, $*$, 1, $*$, 1, 0, $*$, $*$, $*$}

Further definitions:

- Instance of the schema H: 1 0 0 $*$ 1 $*$ 1 0 $*$ $*$ $*$ → 1 0 0 0 1 0 1 0 1 0 1

- Set of all instances of schema H = (h_1, … , h_ℓ):
I(H) = {(a_1, … , a_ℓ) ∈ IB^ℓ | $h_i \neq * \Rightarrow a_i = h_i$}

Schemata (2)

Further definitions continued:

- Order of the schema: Number of instantiated elements (6 in our example).

$$o(H) = |\{i \mid h_i \in \{0,1\}\}|$$

- Defining length of the schema: length of the sub-string starting at the first and ending at the last instantiated element (7 in our example). Idea: it is the number of possible breakpoints

$$\underbrace{1\,0\,0 * 1 * 1\,0}\,* * *$$

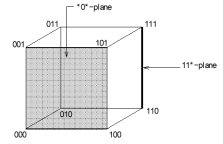
$$d(H) = \max\{i \mid h_i \in \{0,1\}\} - \min\{i \mid h_i \in \{0,1\}\}$$

Schemata (3)

Some numbers:
- In total there are 3^ℓ different schemata.
- Each chromosome (in IB^ℓ) is an instance of 2^ℓ different schemata.
- Thus: at most N · 2^ℓ schemata are represented in a population of size N.

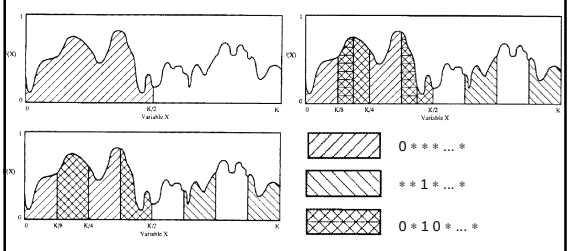
A schema can be viewed as a hyperplane of an n-dimensional space.

Examples in a 3-dimensional (hyper)cube:



Schemata (4)

A function and various (schema) partitions of hyperspace:



Schema Theorem (1)

Theorem (Holland '75):

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\bar{f}} \cdot \left(1 - p_c \frac{d(H)}{l-1}\right) \cdot (1-p_m)^{o(H)}$$

- f to be maximised, $\bar{f}$: mean fitness in population
- l: length of the string
- H: a schema
- d(H): defining length
- o(H): order
- p_m: mutation rate
- p_c: crossover rate
- f(H): (estimated) schema fitness
- m(H, t): expected number of instantiations of H in generation t

Schema Theorem (2)

- Expected number of instantiations of H selected for the gene pool:
$$m(H,t)\cdot\frac{f(H)}{\bar{f}}$$
- Probability that crossover does not occur within the defining length:
$$1-p_c\frac{d(H)}{l-1}$$
- Probability that the schema is not mutated:
$$(1-p_m)^{o(H)}$$

Schema Theorem (3)

Critique on the schema theorem (Bäck '96):

- Most of Holland's approximations are only true for very large numbers (trials and population size).
- Within finite populations, exponentially increasing/decreasing the number of schema instances, leads to entirely filling the population and complete elimination, respectively.
- Not all schemata are represented in a typical population.
- Schemata of large defining length are likely to be destroyed by crossover (even highly fit ones).

The Goldberg example (1)

String number	Initial population (Randomly generated)	x value (Unsigned integer)	$f(x)$ (x^2)	pselect$_i$ $\left(\frac{f_i}{\sum f}\right)$	Expected count $\left(\frac{f_i}{\bar{f}}\right)$	Actual count (From roulette wheel)
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			293	0.25	1.00	1.0
Max			576	0.49	1.97	2.0

Schema Processing

		Strings Representatives	Schema Average Fitness $f(H)$
H_1	1 * * * *	2,4	469
H_2	* 1 0 * *	2,3	320
H_3	1 * * * 0	2	576

The Goldberg example (2)

Mating pool after reproduction (Cross site shown)	Mate (Randomly selected)	Crossover site (Randomly selected)	New population	x Value	$f(x)$ (x^2)
0 1 1 0 | 1	2	4	0 1 1 0 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 1	25	625
1 1 | 0 0 0	4	2	1 1 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 0 0 0	16	256
					1754
					439
					729

Schema Processing

After reproduction			After all operators		
Expected Count	Actual Count	String Representatives	Expected Count	Actual Count	String Representatives
3.20	3	2,3,4	3.20	3	2,3,4
2.18	2	2,3	1.64	2	2,3
1.97	2	2	0.00	1	4

Almost sure convergence (1)

First theoretical result based on Markov chains

Theorem (Eiben et al. '91):

Let - P_t be the population at time t generated by an EA
- *Optima:* set of global optima of a function f

If - $\max_{\bar{x}\in P(t)} f(\bar{x}) \geq \max_{\bar{x}\in P(t-1)} f(\bar{x})$ and
- any point is accessible from any other point

then
$$P[\lim_{t\to\infty} P_t \cap Optima \neq 0] = 1$$

Note: elitist selection and $p_m > 0$ satify the conditions

Almost sure convergence (2)

Critique on the theorem:

- It says nothing about convergence speed
- Theory people: I don't care if it works if only it converges
- Practice people: I don't care if it converges if only it works

Theorem later generalized by Rudolph

Implicit parallelism

- **Implicit parallelism**: a GA with population size N processes more than N different schemata effectively
- Reason: individuals are instantiations of more than one schema
- Effectively processing of a schema:

Sampled at the desirable exponentially increasing rate.

- Why wouldn't a schema be processed effectively?

Schema disruption by genetic operators!

- Holland's estimate: $O(N^3)$ schemata are processed effectively when using a population of size N

The Building Block Hypothesis

Building Block Hypothesis (Holland '75):

GA's are able to detect short, low order and highly fit schemata and combine these into highly fit individuals.

Building blocks are small and good schemata, where:

- small is:
  - short (i.e have a small defining length)
  - of low order, and
- good is: highly fit (estimated fitness in present population).

Implicit parallelism and the Building Block Hypothesis are seen as explanations for the power of GA's.

No Free Lunch Theorem (NFL)

Theorem (Macready & Wolpert'97)

Informal phrasing 1

- For any measure of algorithm performance
- For any two search algorithms
- The aggregate behavior over all possible discrete functions is equivalent

Informal phrasing 2

- Without any structural assumptions on a discrete optimization problem, no algorithm can perform better on average than blind search

NFL (2)

- $F : X \rightarrow Y$ the set of all functions from X to Y (X, Y finite)
- $f \in F$ is any given objective function
- an algorithm will generate samples from $D_m = (X \times Y)^m$
  - m is the sample size
  - $\langle d(x,1,m), \ldots, d(x,m,m) \rangle$ are the points, notations: $d(x,m)$, $d(x)$
  - $\langle d(y,1,m), \ldots, d(y,m,m) \rangle$ are their f values, notation: $d(y,m)$
- an algorithm is a: $d \in D \mid\rightarrow \{ x \mid x \notin d(x) \}$, where $D = U_{m>0} D_m$
- the performance of algorithm a: $P(d(y,m) \mid f,m,a)$ is the conditional probability of getting the sample d(y,m) after m iterations on f

Theorem

$$\Sigma_f P(d(y,m) \mid f,m,a_1) = \Sigma_f P(d(y,m) \mid f,m,a_2)$$

I.e., the quality of generated samples over all objective functions is independent from the algorithm (minima depend on $d(y,m)$ only)

NFL (3)

Remarks

- Discreteness assumption seen as not very restrictive because of inherently discrete computer representations
- "No repetitions" condition ($\{ x \mid x \notin d(x) \}$) is crucial
- An algorithm in this formulation is deterministic, but
  - Pseudorandom number generators are deterministic given a seed
  - Theorem generalizable to stochastic case
- Says nothing about speed
- Does not hold for a subset $G \subset F$
- Thus: there can be differences on problems of type T, etc.