Lecture 1

Evolution in Biology

- study of diversity of life
  - Differences & similarities of organisms
  - Adaptive & non-adaptive characteristics of organisms
- Main features
  - population(s)
  - Reproduction with Inheritance
  - genetic variation
  - natural selection

=> can be understood & represented algorithmically

Evolution as Algorithm (EA)

```
INIT population                          // Choose Population Size, initialization & representation
EVAL quality of each candidate           // Define evaluation function (fitness function)
WHILE not is_terminal() {                // Specify termination criterion
    SELECT candidates for reproduction   // Choose candidate selection mechanism
    RECOMBINE candidates
    MUTATE candidates                    } // Define variation operators (recombine & mutate)
    EVAL new candidates
    SELECT new candidates for next Gen   // Choose survival mechanism
}
```

NOTE: Three alternatives for is_terminal():

1. Satisfying candidate obtained
2. Predefined Limit of iterations / Gens reached
3. Population converged to low level of variety

Toy Example: Canonical GA

Task: find $\max_x f(x)$, for $f(x) = x^2$, $0 \leq x \leq 31$

NOTE: $\max_x f(x) = 31$, $f(31) < x^2$

1. Representation:

⇒ Use unsigned binary integer of length 5: 10110 ⟺ 22

00000 ⟺ 0 & 11111 ⟺ 31

2. Choose Population Size & init method, eval fitness

⇒ Use random initialization, Population size of 4 = n

⇒

Index	Individual	x-value	$f(x) \hat{=}$ fitness
1	01101	13	169
2	11000	24	576
3	01000	8	64
4	10011	19	361

3. Choose Candidate Selection Mechanism:

⇒ We use fitness proportional selection:

1. Calc sum: $\sum_x f(x) = 1170 = S_x$
2. $\forall x: p_x = \frac{f(x)}{S_x}$ ⇒ Probability Dist. ① 0.14 ② 0.49 ③ 0.06 ④ 0.31
3. Expected number $e_x = p_x \cdot n$: ① 0.56 ② 1.96 ③ 0.24 ④ 1.24

⇒ Mating Pool: ① ② ② ④

4. Define Crossover

⇒ One-point Crossover

5. Define Mutation:

⇒ Bit-flip Mutation

6. Choose Survival Selection Mechanism

=> Generational: Place created individuals into new population that replaces the old

=> Repeat until termination condition is satisfied

Convergence of CGA

Convergence $\hat{=}$ Probability that population contains global optimum approaches 1 as $t \to \infty$

- CGA does not converge to global optimum
- CGA with elitist selection does converge

Why Do GAs Work? Schema Theorem

=> Analyzes effect of selection, crossover & mutation on schemas

Schema: E.g. $0*11*$ represents set $\{00110, 01110, 00111, 01111\}$

$*$ $\hat{=}$ Don't care: either 0 or 1

NOTE: 10 has length $L=2$ and belongs to $2^1 = 2^2$ different schemas:

$(10), (*0), (1*), (**)$

Schema Theorem

Schema with above average fitness, small number of fixed bits (order) and a small length ($\hat{=}$ longest distance between 2 fixed positions), is more likely to survive after selection, crossover and mutation

---

Schema Theorem in Words

$N \hat{=}$ population size

Expected nr. of instances of schema H in generation $k+1$ =

N * prob. of selecting schema H at gen k *

prob of surviving crossover at gen k *

prob of surviving mutation at gen k

Order (fixed bits) of Schemas

=> Order of a schema $\hat{=}$ # of fixed bits

E.g: $H = 0*1 \rightarrow o(H) = 2$

$H = 0** \rightarrow o(H) = 1$

$\Rightarrow$ Smaller order $\hat{=}$ more difficult to disrupt

---

Defining Length of Schemas

$\Rightarrow$ Longest Distance between 2 fixed positions

E.g.: $H = *1*01 \rightarrow d(H) = 5-2 = 3$

$H = 0**** \rightarrow d(H) = 1-1 = 0$

$\Rightarrow$ Schema with small defining length is harder to disrupt

---

Schema Instantiation

$\Rightarrow x$ is instance of $H \hat{=} x \in H \hat{=} x$ belongs to schema H

$m(H,k) \hat{=} \#$ of instances of H in Gen k

Fitness of a Schema

$$f(H,k) = \frac{1}{m(H,k)} \sum_{\substack{x \in H \wedge \\ x \in k}} m(x,k)\, f(x)$$

$m(x,k) \hat{=} \#$ of copies of x in k

$f(x) \hat{=}$ fitness of x

E.g.: $k = \{(0,0), (0,0), (0,0), (0,0), (1,0)\}$

$f(0,0) = 1 \quad f(1,0) = 0.5$

Avg. fitness: $\frac{1}{m(H,k)} \sum_{\substack{x \in H \wedge \\ x \in k}} m(x,k) f(x)$

$$\bar{f} = \frac{1}{5}(4 \cdot 1 + 1 \cdot 0.5) = \frac{4.5}{5} = 0.9$$

Probability of Selecting Schema in CGA

$\Rightarrow$ We used fitness proportional selection

$P_s(x) \hat{=}$ probability of selecting x

$$= \frac{f(x)}{\sum_{i=1}^{N} f(x_i)}$$

$$p(H,k) = \sum_{\substack{x \in H \wedge \\ x \in k}} m(x,k)\, P_s(x)$$

$$= \frac{1}{N} m(H,k) \frac{f(H,k)}{\bar{f}}$$

$\Rightarrow$ Schemes with fitness > pop. avg. are likely to appear more in next Gen!

Probability of Surviving Crossover in CGA

=> We used 1-point crossover => crossover point $\hat{=} C$

=> survival $\hat{=}$ either parent is $\in H \wedge$ offspring $\in H$

=> parents $= X_1, X_2$

If $(X_1 \in H \vee X_2 \in H) \wedge C$ in $d(H)$: H is destroyed unless the other parent repairs it

Probability $P_{C \text{ in } d(H)} = d(H)/(l-1)$

Example: $H = *1**0$, $l = 5$, $d(H) = 5-2 = 3$

$P_{C \text{ in } d(H)} = \frac{d(H)}{(l-1)} = \frac{3}{4} = 0.75$

$D_c(H) \hat{=}$ probability of schema getting destroyed

$P_c \hat{=}$ probability of crossover

$$D_c(H) \leq P_c \frac{d(H)}{(l-1)}$$

Example: $P_c = 0.8$, $l = 100$

If $d(H) = 3 \rightarrow D_c(H) \leq 0.8 \frac{3}{99} = 0.024$

If $d(H) = 50 \rightarrow D_c(H) \leq 0.8 \frac{50}{99} = 0.404$

NOTE: We use $\leq$ cause the other parent could theoretically repair the destroyed schema

$S_c(H) \hat{=}$ min. probability that H survives crossover

$$= 1 - D_c(H) \geq 1 - P_c \frac{d(H)}{(l-1)}$$

=> Schemas with small $d(H)$ are more likely to survive cross over

Probability of Schemes Surviving Mutation in CGA

=> Assume: $P_m \hat{=}$ probability that bit-flip mutation is applied, for each position

↳ For H to survive, all fixed bits must stay unchanged

recall: $O(H)$ = order of H = # of fixed bits

$(1 - P_m) \hat{=}$ prob. of bit not getting flipped

$S_m(H) \hat{=}$ prob of H surviving mutation $= (1 - P_m)^{O(H)}$

=> Schemas with low order are more likely to survive?

Schema Theorem in formulas

$$\mathbb{E}[\text{\# of instances of H in } k+1] > N * p(H,k) * S_c(H) * S_m(H)$$

$$\mathbb{E}[m(H, k+1)] \geq m(H,k) \frac{f(H,k)}{\bar{f}} \left(1 - P_c \frac{d(H)}{l-1}\right) (1 - P_m)^{O(H)}$$

=> A schema with above average fitness, short defining length, and low order is more likely to survive

Building Block Hypothesis

NOTE: Schema theorem identifies the building blocks of a good solution

Q: Why do GAs work?

A: Building block hypothesis (BBH):

=> GAs create stepwise better solutions through selecting, crossing & mutating building blocks

building Block ≙ short, low-order, high-fitness schema

Arguments against BBH

- Superposition of fit schemata possibly creates larger schemata which are less likely to survive
- In populations of realistic size, the observed fitness of instances of a schema may be far from the average fitness used in the def. of schema fitness, even in initial pop.

When Does BBH not hold?

=> BBH doesn't hold if there is no useful information for guidance of sub-optimal solutions

=> In sparse reward settings e.g. chess or needle-in-haystack

$$f(x_1,\dots,x_n) = \begin{cases} 1 & \text{if } (x_1,\dots,x_n) = (0,\dots,0) \\ 0 & \text{else} \end{cases}$$

Extensions of CGA

Selection Operator: Tournament selection with param k:

=> Select k individuals randomly

=> Individual with best fitness is selected as parent

→ Allows to adjust selection pressure through k

Population Update:

Steady-State: Add/Remove 1 individual each gen.

Elitism: Pass best chromosome(s) to next gen

Adaptive Pop. Size

- Individuals receive a lifespan at birth
  - ↳ higher fitness → higher lifespan
- age increases per generation
- every gen. select part of population for reproduction randomly

Parallelization

=> Use multiple populations with migration scheme

Hybridation through local search (Memetic Algorithms)

=> Every Gen. apply local search to the individuals

Memetic Algorithm ≙ Search among locally optimal solutions

GA for Travelling-Salesman Problem