Ch9. Genetic algorithms 9.1 - 9.4

S. Visa

Genetic algorithm (GA)

- J. Holland, U. Michigan (60's) → develop computers that could adapt to any environment using biological evolution as example
- = search space H (large!) to find a best (good) h → "population" of hypotheses h that "evolves" according to a "survival of the fittest" technique
- "best" = optimizes a fitness function (e.g. accuracy in classification using h)
- Not guaranteed to find optimal solution
- Analogy to biological evolution

Components in GA

- Size of population to be maintained (how many h)
- Fitness function
- Threshold to indicate acceptable fitness level (to terminate the algorithm)
- Parameters to compute next generation:
  - Fraction of population replaced at each generation
  - Mutation type and rate

Algorithm

GA($Fitness, Fitness\_threshold, p, r, m$)

- *Initialize:* $P \leftarrow p$ random hypotheses
- *Evaluate:* for each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$
  1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S.
  $$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$
  2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate:* Invert a randomly selected bit in $m \cdot p$ random members of P_s
  4. *Update:* $P \leftarrow P_s$
  5. *Evaluate:* for each h in P, compute $Fitness(h)$
- Return the hypothesis from P that has the highest fitness.

Example:

- p = 8 (no. of h in population)
- r = ½ (fraction of p to be replaced by crossover)
- m = 1/8 (mutation rate)
- 2. Crossover: from 2 pairs of h obtain through crossover 4 offspring
  - ➔ Population = 4 best fit h + 4 crossover children
- 3. Mutate: randomly select 1/8 of population; for each invert one randomly selected bit
- OBS. Probability that h is selected is proportional to its own fitness!
- GOAL: breed a population of high fitness h

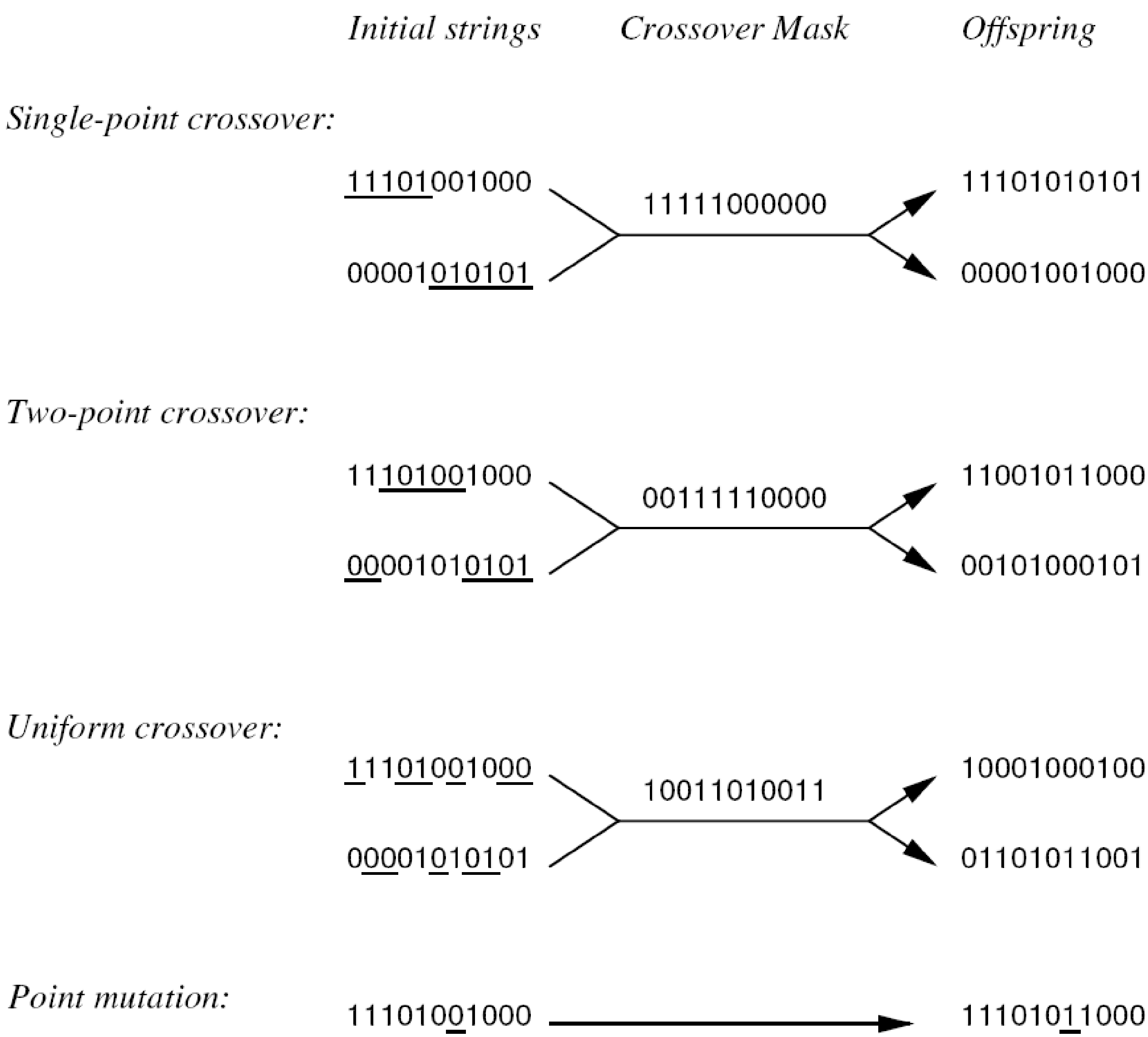
Representing hypotheses

- By bit strings

- E.g. if rule as bit string
  - Assume:
    - attr. Wind can take 2 values ➔ 2 bits: 10, 01, 11, 00
    - attr. Outlook can take 3 values ➔ 3 bits: 100, 111, …
  - IF (Outlook = overcast OR rain) AND (Wind=strong) then Play=yes
    - **➔h1 = <011 10 1>**
  - IF (Wind=strong) then Play=no
    - **➔h2 = <111 10 0>**

- **Critical issue in using successfully GA**

Operators for GA



Crossover and mutation

- Crossover
  - Produces two offspring from 2 parents by copying selected bits from each parent

- Mutation
  - Flips a bit
  - Produces one offspring from a single parent
  - Usually used after crossover
  - Rare in nature! ➔ rate of mutation in GA: 0.01 - 0.001
  - Often has harmful results
  - Still, can spin you out of a local minima

Selecting most fit h

- Roulette wheel selection $\Pr(h_i) = \frac{Fitness(h_i)}{\Sigma_{j=1}^{p} Fitness(h_j)}$
  - ➔ may lead to crowding
    - = highly fit individuals quickly reproduce ➔ very similar individuals take over a large fraction of population
    - Reduces diversity

- Other selection methods
  - Tournament selection
    - Pick h1, h2 randomly
    - With probability Pr(h) (from above) select most fit
  - Fitness sharing
    - Fitness measure of an individual is reduced by presence of other similar individuals in population

GA for concept learning - Example

- GABIL (GA Batch Learning) system – DeJong(‘93)
  - r=0.6, m=0.001, p=100 …1000
  - Two-point crossover
  - Representation: bit-string repres. of individual rules are concatenated

    IF $a_1 = T \wedge a_2 = F$ THEN $c = T$; IF $a_2 = T$ THEN $c = F$

    represented by

$$\begin{array}{ccccccc} a_1 & a_2 & c & & a_1 & a_2 & c \\ 10 & 01 & 1 & & 11 & 10 & 0 \end{array}$$

- Learn Boolean concepts represented by a disjunction of rules (e.g. breast cancer diagnosis)

- Results comparable to C4.5 (~91% accuracy)

Crossover with variable length bitstrings

Start with

$$\begin{array}{lllllll} & a_1 & a_2 & c & a_1 & a_2 & c \\ h_1: & 10 & 01 & 1 & 11 & 10 & 0 \\ \\ h_2: & 01 & 11 & 0 & 10 & 01 & 0 \end{array}$$

1. choose crossover points for h_1, e.g., after bits 1, 8
2. now restrict points in h_2 to those that produce bitstrings with well-defined semantics, e.g., $\langle 1,3\rangle$, $\langle 1,8\rangle$, $\langle 6,8\rangle$.

if we choose $\langle 1,3\rangle$, result is

$$\begin{array}{llll} & a_1 & a_2 & c \\ h_3: & 11 & 10 & 0 \end{array}$$

$$\begin{array}{llllllllll} & a_1 & a_2 & c & a_1 & a_2 & c & a_1 & a_2 & c \\ h_4: & 00 & 01 & 1 & 11 & 11 & 0 & 10 & 01 & 0 \end{array}$$

Schema theorem

- Characterizes evolution of population in terms of the no. of instances representing each schema

- Schema = string of 0,1,* (don't care)
  - Schema: 0**1*
  - Instance of above schema: 0**01**1**0**, 0**11**1**1**, …

- ➔ characterize population by no. of instances representing each possible schema

- m(s,t) = no. of instances of schema s in pop. at time t

- Obs. An individual may belong to (represent) several schemas

- Evolution of a schema depends on
  - Selection
  - Crossover
  - Mutation

Selection step analysis

- $\bar{f}(t)$ = average fitness of pop. at time t
- $m(s,t)$ = instances of schema s in pop at time t
- $\hat{u}(s,t)$ = ave. fitness of instances of s at time t

Probability of selecting h in one selection step

$$\begin{aligned} \Pr(h) &= \frac{f(h)}{\sum_{i=1}^{n} f(h_i)} \\ &= \frac{f(h)}{n\bar{f}(t)} \end{aligned}$$

Probabilty of selecting an instance of s in one step

$$\begin{aligned} \Pr(h \in s) &= \sum_{h \in s \cap p_t} \frac{f(h)}{n\bar{f}(t)} \\ &= \frac{\hat{u}(s,t)}{n\bar{f}(t)} m(s,t) \end{aligned}$$

Expected number of instances of s after n selections

$$E[m(s,t+1)] = \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t)$$

- Expected number of instances of schema s at t+1
  - Proportional to avg. fitness of instances of s at t
  - Inversely proportional to the avg. fitness of all population p at t
- ➔ schemas with above average fitness will be represented in next generation with increasing frequency

Schema theorem

Effect of selection

Effect of 1-pt. crossover

Effect of mutation

$$E[m(s,t+1)] \geq \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t) \left(1 - p_c \frac{d(s)}{l-1}\right) (1-p_m)^{o(s)}$$

- $m(s,t)$ = instances of schema s in pop at time t
- $\bar{f}(t)$ = average fitness of pop. at time t
- $\hat{u}(s,t)$ = ave. fitness of instances of s at time t
- p_c = probability of single point crossover operator
- p_m = probability of mutation operator
- l = length of single bit strings
- $o(s)$ number of defined (non "*") bits in s
- $d(s)$ = distance between leftmost, rightmost defined bits in s

- Lower bound on expected frequency of schema s (considering selection, crossover, and mutation steps)
- ➔ More fit schemas will grow in influence, especially schemas having many *

Obs. on GA

- Bit string encoding (and evaluation function design) → critical!!!!!
- Not necessary binary strings → may use characters from an alphabet
- Easy to implement
- Requires no complex math tools (e.g. derivatives in NN)
- Only changes from problem to problem
  - Evaluation function
  - Number of bits in string
- Not guaranteed to find optimal solution
- "second best way" to solve a problem
- Applied in optimization problems
  - Circuit layout
  - Choosing net topology of a NN
  - TSP

Why GA works - schemas

- Use schema to characterize the evolution