IE 607 Heuristic Optimization

Genetic Algorithm

Origins

- John Holland, CS/EE, U. of Michigan, ideas of adaptive or reproductive plan (1962)
- Ken DeJong, John's student, now CS at George Mason U., function optimization (1975)
- David Goldberg, John's student, now at U. of Illinois at Urbana Champaign (1983 Ph.D. & 1989 book)
- Bagley (1967) game-playing program; Rosenberg (1967) simulated biological process; Cavicchio (1970) subroutine selection & pattern recognition

Key Ideas

Population evolves mainly through sexual reproduction (crossover) with mutation a secondary operator

GA Terminology

encode problem as bit string – ***chromosome*** (also called ***strings***, ***vectors***, or ***solutions*** in some occasions)

each variable – ***gene***

each component(bit) of variable – ***allele***; i.e., possible values of a variable

location of allele – ***locus***; i.e., position of a variable in a string

total package of strings – ***structure***

e.g. an animal’s eye color *gene*, its ***locus*** - position 10, and its ***allele*** value - blue eyes.

GA Terminology (cont.)

genotype – coded string processed by the algorithm; i.e., actual structure

phenotype – decoded solution; i.e., physical expression of the structure

GA Terminology (cont.)

epistasis – non linearity (independence) of alleles; i.e., the interaction between genes such that the contribution of a gene to the fitness depends on the value of other genes in the chromosome

e.g. for echo-location, bats must be able to generate ultrasonic squeaks, *and* have a hearing system capable of detecting the echoes. Therefore, the genes for good hearing can only increase the "fitness" of a bat if it also has genes for squeak production.

GA Terminology (cont.)

fitness – objective function value

→Each chromosome encodes a solution to the problem, and its fitness value is related to the value of the objective function for that solution.

GA Terminology (cont.)

SCHEMA (*pl*. schemata) – specified alleles (rest of the chromosome wild cards represented by *); i.e., defines subsets of similar chromosomes, or as hyperplanes in n-dimensional space.

· schema order (o) – number of non wild card alleles

· schema length (δ) – distance from first to last non wild card alleles

· $(k+1)^l$ schema for alphabets of cardinality k (i.e., number of alphabet characters) and chromosome length l

GA Terminology (cont.)

<u>Example 1</u>: 0, 1, * (i.e., k = 2)

if l=5 → $(k+1)^l = 3^5 = 243$ different similarity templates

<u>Example 2</u>: chromosome: 0 1 0 1 0 0 0 1

schema: * 1 0 * * 0 * *

order is 3 and length is 4

schema: 0 * * * * * * * *

order is 1 and length is 0

schema: * 1 0 1 0 0 0 *

order is 6 and length is 5

Schema Theorem

(Fundamental Theorem of GA): good schemata are sampled over evolution with exponential increases

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

where *m* is the number of schema *H*, *t* is a generation, $f = \frac{\sum f_j}{n}$ is the mean fitness of population, *f(H)* is the

mean fitness of strings containing *H*, p_c is the crossover probability, p_m is the mutation probability, *l* is the chromosome length, δ*(H)* is the length of *H* and *o(H)* is the order of *H*. *m(H,t)* denotes that at a given generation *t*, there are *m* examples of a particular schema *H* contained within the population.

Schema Theorem (cont.)

Reproduction:

$$E\{m(H,t+1)\} = E\{m(H,t)\} \cdot n \cdot \frac{f(H)}{\sum f_j} = E\{m(H,t)\} \cdot \frac{f(H)}{\sum f_j / n}$$

$$= E\{m(H,t)\} \cdot \frac{f(H)}{\bar{f}}$$

crossover: $p_d = \frac{d(H)}{l-1}, p_s = 1 - p_d \Rightarrow p_s \geq 1 - p_c \frac{d(H)}{l-1}$

mutation: a single allele survives with probability of $1\text{-}p_m$, and each of the mutations is independent. The probability of surviving mutation $(1\text{-}p_m)^{o(H)}$, for $p_m << 1 \rightarrow 1\text{-}o(H)p_m$

Schema Theorem (cont.)

Schema Theorem favors highly fit (above average), short length, low order schemata, which are termed ***building blocks*** → receive exponentially increasing trials in subsequent generations

Implication of Schema Theorem – *implicit parallelism* as each chromosome has multiple schema

Canonical GA

encode problem as bit string

set p_c, p_m, ps, g_{max}

randomly generate ps solutions

calculate fitness

until g = g_{max} {

for 1 to ps / 2 {

select 2 parents considering fitness (biased Roulette Wheel Selection)

crossover with p_c probability to produce 2 children

mutate children with p_m probability at each allele }

replace parents with children }

return

note: p_c is generally very large and p_m is generally very small.

Example of GA

max x^2 $-10 \leqq x \leqq 10$

pick chromosome length (l) = 5 and ps = 4

precision of bit encoding: $\frac{a-b}{2^l-1} = \frac{10-(-10)}{2^5-1} = 0.645$

decoding = $b + \frac{a-b}{2^l-1} \cdot \sum_{j=0}^{l-1} s_{l-j} 2^j$ where $x = (s_1, \ldots, s_l)$

randomly generate initial solutions:

x_1 = 01110 = (.645)(2+4+8) – 10 = -0.97 → x_1^2 = 0.94

x_2 = 01111 = (.645)(1+2+4+8) – 10 = -0.325 → x_2^2 = 0.11

x_3 = 10011 = (.645)(1+2+16) – 10 = 2.26 → x_3^2 = 5.09

x_4 = 00100 = (.645)(4) – 10 = -7.42 → x_4^2 = 55.06

sum of fitness 61.20

Example of GA (cont.)

p(x_1 is selected) = 0.94 / 61.20 = 0.02
p(x_2 is selected) = 0.11 / 61.20 = 0.00
p(x_3 is selected) = 5.09 / 61.20 = 0.08
p(x_4 is selected) = 55.06 / 61.20 = 0.90

select x_1 and x_4 for single point crossover and randomly select locus 2 as crossover point:

0 1 1 1 0 parent 1
0 0 1 0 0 parent 2

0 1 1 0 0 child 1, fitness = 5.11
0 0 1 1 0 child 2, fitness = 37.58

Example of GA (cont.)

mutate children with $p_m = 0.10$

1 1 1 0 0 mutated child 1, fitness = 64.96

0 0 1 1 0 mutated child 2, fitness = 37.58

select 2 more parents and produce 2 more children

replace old population with the 4 new children

continue until termination criteria is met

return best solution of final population

Variations

a. encoding

bit string

real number

permutation (also called path or order)

matrix

parse tree

mixed

Variations (cont.)

b. initial population

random

seeding

Variations (cont.)

c. selection for parents

biased roulette wheel

rank based roulette wheel

tournament

deterministic selection – top half/truncation selection; elitist selection

Variations (cont.)

d. crossover

single point

2-point or multi-point

uniform

biased uniform

partial-mapped (PMX)

order (OX)

position-based

order-based

cycle (CX)

Variations (cont.)

d. crossover (cont.)

subtour exchange

heuristic

arithmetic

intermediate

simplex

geometrical

fitness-based scan

etc.

Variations (cont.)

e. mutation

bit flip

inversion

insertion

2 opt (reciprocal exchange)

heuristic

etc.

Variations (cont.)

f. population maintenance

replace parents with children

keep best of pooled population

elitism

Variations (cont.)

g. termination

maximum generations

population homogeneity

non improvement of best

Advanced Variations

a. **local search** options

memetic algorithm

Lamarckian – change chromosome

Advanced Variations (cont.)

b. speciation and **sharing**

discourage solution similarity

share fitness over same niche

Advanced Variations (cont.)

c. fitness scaling

static / dynamic linear

sigma truncation

power law

logarithmic

normalizing

penalized

Advanced Variations (cont.)

d. deception

gray codes

messy coding

floating-point coding

edge encoding

random keys

Advanced Variations (cont.)

e. parallel implementations

solutions in parallel

populations in parallel

Advanced Variations (cont.)

f. use of **search feedback**

change p_c, p_m

change ps

change local search

Advanced Variations (cont.)

g. multi-criteria optimization

multiple populations

sequential evolution

Pareto optimality through ranking