Evolutionary Algorithms

1

Evolutionary Algorithms (EA)

group of search and optimization algorithms based on

- Mendelian genetics
- Darwinian theory of evolution

which are

- heuristic
- stochastic
- population based

2

Evolutionary Algorithms

- genetic algorithms (Holland 1975)
- genetic programming (Koza 1989)
- evolutionary strategies (Rechenberg 1973)

Evolutionary Algorithms

- genetic algorithms
- genetic programming
- evolutionary strategies
- evolutionary programming
- differential evolution
- grammatical evolution
- memetic algorithm

EA Outline

```
Algorithm EA:
INITIALIZE population randomly
CALCULATE_FITNESS of each individual
while not STOP_CRITERIA do
      SELECT parents
      RECOMBINE pairs of parents
      MUTATE offspring
      CALCULATE_FITNESS of offspring
      REPLACE (some) parents by offspring
end_do
```


EA Terminology

- gene
- chromosome
- fitness
- individual (contains a chromosome and has a fitness) = candidate solution
- parents, offspring
- population
- generations = iterations

Components of an EA

- representation for candidate solutions
- a population of candidate solutions
- method for creating an initial population of candidate solutions
- evaluation function to rate candidate solutions
- parent selection mechanism
- genetic operators to alter composition of population of candidate solutions
- replacement mechanism
- various parameters to control a run

Parameters of an EA

- number of generations
  - or some other stopping criteria
- population size, i.e. no. of individuals in a population
- probability of applying variation operators
- chromosome length (depends on problem instance)
- other parameters

The Simple Genetic Algorithm (SGA)

9

Simple GA -SGA

- a.k.a., Canonical GA
- Operators of a SGA
  - selection
  - crossover
  - mutation

10

SGA

```
generate initial population
repeat
    evaluate individuals
    perform reproduction
    select pairs
    recombine pairs
    apply mutation
until end_of_generation
```


Representation & Encoding

- population size constant
- individual has one chromosome
- chromosome length constant
- individual has a fitness value
- binary genes (0/1)
- generational

Initial Population

random initial population

each gene value for each individual determined
randomly to be either *0* or *1*
with equal probability

Fitness Evaluation

- fitness function
  - objective function(s)
  - constraints
- shows fitness of individual
  - degree to which the candidate solution meets the objective
- apply fitness function to individual

Example Problem: One-Max

Objective: maximize the number of **1**s in a string of length 5, composed only of **1**s and **0**s

→ *population size* = 4

chromosome length = 5

fitness function = no. of genes that are **1**

15

Example Population

individual 1:

chromosome = 11001

fitness = 3

individual 2:

chromosome = 00001

fitness = 1

individual 3:

chromosome = 11111

fitness = 5

individual 4:

chromosome = 01110

fitness = 3

16

Reproduction

- consists of
  - selection
    - mating pool (size same as population)
    - possibly more than one copy of some individuals
  - crossover
  - mutation

17

Selection: Roulette-Wheel Selection

- fitness proportionate selection
  - expected no. of representatives of each individual is proportional to its fitness

$$prob_i = \frac{fitness_i}{\sum_j fitness_j} \quad , \; j = 1, \dots, pop.size$$

18

Selection: Roulette-Wheel Selection

```
begin
  set current_member=1;
  while (current_member ≤ m)do
    pick uniform r.v. r from [0,1];
    set i=1;
    while (ai < r) do
      set i=i+1;
    od
    set mating_pool[current_member]=parents[i];
    set current_member=current_member+1;
  od
end
```

m: population size, $a_i = \sum_1^i P_{sel}(i)$, $i = 1,2, \dots , m$

Selection: Tournament Selection

- ordinal based
- roulette wheel selection uses info on whole population
  - info may not be available
    - population too large
    - population distributed on a parallel system
    - maybe no universal fitness definition (e.g., game playing, evolutionary art, evolutionary design)

Selection: Tournament Selection

- relies on an ordering relation to rank any n
- individuals
- most widely used approach
- tournament size k
  - if k large, more of the fitter individuals get a chance

Selection: Tournament Selection

```
begin
  set current_member=1;
  while (current_member ≤ m)do
    pick k individuals randomly;
    select best from k individuals;
    denote this individual i;
    set mating_pool[current_member]=i;
    set current_member=current_member+1;
  od
end
```

m: population size , k: tournament size

Example: Roulette Wheel Selection

Current Population:

i1: 11001, 3
i2: 00001, 1
i3: 11111, 5
i4: 01110, 3

Probability of each individual being selected:

prob(i1) = 3/12 = 0.25
prob(i2) = 1/12 = 0.08
prob(i3) = 5/12 = 0.42
prob(i4) = 3/12 = 0.25

Expected copies of each individual in pool:

i1: (3/12*4) 1
i2: (1/12*4) 0
i3: (5/12*4) 2
i4: (3/12*4) 1

Assume:
wheel is turned 4 times
1 copy of i1
2 copies of i3
1 copy of i4
is copied into mating pool

Recombination

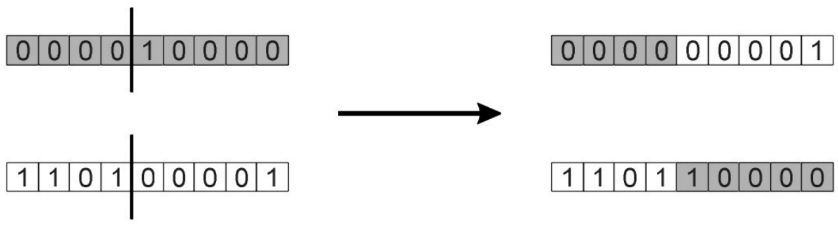
- process for creating new individuals (offspring)
  - from two or more parents
- term used interchangeably with crossover
  - mostly refers to 2 parents
- crossover rate **pc**
  - typically in range [0.5,1.0]
  - acts on parent pair

Recombination

- two parents selected randomly
- a r.v. drawn from [0,1)
- if value < pc , then two offspring created through recombination
- else, two offspring created asexually
  - i.e., copy of parent

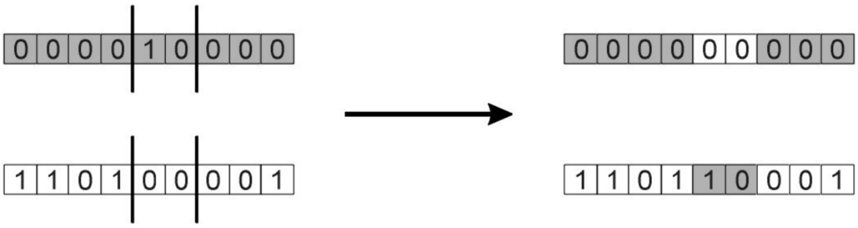
25

One-Point Crossover



26

Two-Point Crossover



Can be N point crossover. In the above example: N=2

27

Example Pairing

<u>Current mating pool:</u>

mate 1: 11001 (i1)

mate 2: 11111 (i2)

mate 3: 11111 (i3)

mate 4: 01110 (i4)

<u>Assume:</u>
As a result of random drawing
(mate 1, mate 3)
and
(mate 2, mate 4)
are paired off for reproduction.

<u>Pairs:</u>

<u>Pair 1:</u>	**<u>Pair 2:</u>**
11001	11111
11111	01110

28

Example One-Point Crossover

Assume: pc= 1.0

for pair 1:
crossover site: 3
110 | 01 ⟶ 11011
111 | 11 ⟶ 11101

for pair 2:
crossover site: 1
1 | 1111 ⟶ 11110
0 | 1110 ⟶ 01111

the new individuals:

i1: 11011
i2: 11101
i3: 11110
i4: 01111

Mutation

- bitwise mutation
- probability of mutation: pm
  - a.k.a., mutation rate
  - equal probability for each gene
  - chromosome of length L; expected no. of changes: L*pm
  - typically chosen to be small
    - depends on nature of problem

Bitwise Mutation

31

Example Mutation

Assume:
as a result of random draws,
1st gene of i1
and
4th gene of i3
are found to undergo mutation

i1: 11011 → 01011
i3: 11110 → 11100

32

Generational SGA

- non-overlapping populations
- offspring replace parents
- elitism may be used
  - no. of elite individuals *e*
  - elite individuals replace worst individuals in the new population (if they are better)

33

Example New Population (no elitism)

individual 1:
chromosome = 01011
fitness = 3

individual 2:
chromosome = 11101
fitness = 4

individual 3:
chromosome = 11100
fitness = 3

individual 4:
chromosome = 01111
fitness = 4

34

Stopping Criteria

- main loop repeated until stopping criteria met
  - for a predetermined no. of generations √
  - until a goal is reached
  - until population converges

Example Stopping Criteria

- *number of generations*= 250
  - loop repeated 250 times
  - best individual at each generation (*lbest*) found
  - overall best individual (*gbest*) becomes solution