Lecture 02 - Genetic Representation, Search Operators, Selection Schemes and Selection Pressure

Introduction to Evolutionary Computation

Xin Yao (x.yao@cs.bham.ac.uk)

CERCIA, School of Computer Science

October 5, 2009

Module homepage

`http://www.cs.bham.ac.uk/~pkl/teaching/2009/ec/`

Outline

A Simple Evolutionary Algorithm

Simple Evolutionary Algorithm

Generate the initial population $P(0)$ at random, and set $t \leftarrow 0$.
repeat
    Evaluate the fitness of each individual in $P(t)$.
    Select parents from $P(t)$ based on their fitness.
    Obtain population $P(t+1)$ by
        applying crossover and mutation to parents.
    Set $t \leftarrow t+1$.
until termination criterion satisfied.

- Basic idea from natural evolution and population genetics.
- Survival of the fittest.

Exploration and Exploitation

Exploration of new parts of search space

- Mutation operators
- Recombination operators

Exploitation of promising genetic material

- Selection mechanism

Mutation operators for bitstrings

The mutation operator introduces small,
random changes to an individual's chromosome.

Local Mutation

- One randomly chosen bit is flipped.

Global Mutation

- Each bit flipped independently with a given probability p_m, called the *per bit mutation rate*, which is often $1/n$, where n is the chromosome length.

$$\mathbf{Pr}\left[k \text{bits flipped}\right] = \binom{n}{k} \cdot p_m^k \cdot (1 - p_m)^{n-k}.$$

Mutation rate

- Note the *difference* between per bit (gene) and per chromosome (individual) mutation rates.

Recombination operators - One point crossover

The recombination operator generates an offspring individual whose chromosome is composed from the parents' chromosomes.

Crossover rate

- probability of applying crossover to parents

One point crossover between parents x and y

Randomly select a crossover point p in $\{1, 2, ..., n\}$.
Offspring 1 is $x_1 \cdots x_p \cdot y_{p+1} \cdots y_n$.
Offspring 2 is $y_1 \cdots y_p \cdot x_{p+1} \cdots x_n$.

Example

Parent x: 101011 | 1010	Offspring 1: 101011 | 1110
Parent y: 010100 | 1110	Offspring 2: 010100 | 1010

Recombination operators - Multi-point crossover

k-point crossover between parents x and y

Randomly select k crossover points $p_1 < \cdots < p_k$ in $\{1, 2, ..., n\}$.
Offspring 1 is $x_1 \cdots x_{p_1} \cdot y_{p_1+1} \cdots y_{p_2} \cdot x_{p_2+1} \cdots x_{p_3} \cdots$ etc.
Offspring 2 is $y_1 \cdots y_{p_1} \cdot x_{p_1+1} \cdots x_{p_2} \cdot y_{p_2+1} \cdots y_{p_3} \cdots$ etc.

Example (2-point crossover)

Parent x: 101 | 011 | 1010 Offspring 1: 101 | 100 | 1010

Parent y: 010 | 100 | 1110 Offspring 2: 010 | 011 | 1110

Recombination operators - Uniform crossover

Uniform crossover between parents x and y

Select a bitstring z of length n uniformly at random.
for all i in 1 to n
    if $z_i = 1$ then bit i in offspring 1 is x_i else y_i.
    if $z_i = 1$ then bit i in offspring 2 is y_i else x_i.

Example

$z =$1010001110

Parent x:	1010111010	Offspring 1:	1111001010
Parent y:	0101001110	Offspring 2:	0000111110

Selection and Reproduction

Selection *emphasizes* the better solutions in a population

- One or more copies of good solutions.
- Inferior solutions are much less likely to be selected.
- Not normally considered a search operator, but influences search significantly

Selection can be used either before or after search operators.

- When selection is used before search operators, the process of choosing the next generation from the union of all parents and offspring is sometimes called *reproduction*.

Generational gap of EA

- refers to the overlap (i.e., individuals that did not go through any search operators) between the old and new generations.
- The two extremes are *generational* EAs and *steady-state* EAs.
- 1-elitism can be regarded as having a generational gap of 1.

Fitness Proportional Selection

Probability of selecting individual x from population P is

$$\mathbf{Pr}\left[x\right] = \frac{f(x)}{\sum_{y \in P} f(y)}.$$

- Use raw fitness in computing selection probabilities.
  Does not allow negative fitness values.
- Also known as roulette wheel selection.

Weaknesses

- Domination of "super individuals" in early generations.
- Slow convergence in later generations.

Fitness scaling often used in early days to combat problem

- Fitness function f replaced with a *scaled* fitness function $\tilde{f}$.

Fitness Scaling 1/2

Simple scaling

$$\tilde{f}(x) := f(x) - f_{\min,\omega}, \quad \text{where}$$

- ω is *scaling window*
- $f_{\min,\omega}$ is lowest observed fitness in last ω generations

Sigma scaling

$$\tilde{f}(x) := \min\{0, f(x) - (\bar{f} - c \cdot \sigma_f)\}, \quad \text{where}$$

- c is a constant, e.g. 2
- $\bar{f}$ is average fitness in current population
- σ_f is the standard deviation of the fitness in the current population

Fitness Scaling 2/2

Power scaling

$$\tilde{f}(x) := f(x)^k, \quad \text{where } k > 0.$$

Exponential scaling

$$\tilde{f}(x) := \exp(f(x)/T), \quad \text{where}$$

- $T > 0$ is the temperature, approaching zero.

Ranking Selection

1. Sort population from best to worst according to fitness:

$$x_{(\lambda-1)},\ x_{(\lambda-2)},\ x_{(\lambda-3)},\ \ldots,\ x_{(0)}$$

2. Select the γ-ranked individual $x_{(\gamma)}$ with probability $\mathbf{Pr}[\gamma]$, where $\mathbf{Pr}[\gamma]$ is a ranking function, e.g.
   - linear ranking
   - exponential ranking
   - power ranking
   - geometric ranking

Linear ranking

Population size λ, and rank $\gamma, 0 \leq \gamma \leq \lambda - 1$, (0 worst)

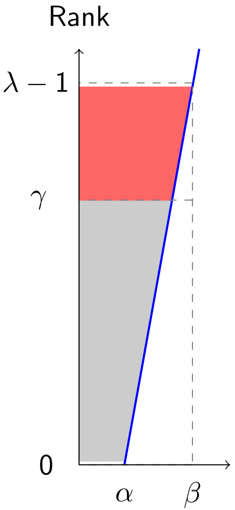
Linear ranking

$$\mathbf{Pr}_{\text{linear}}[\gamma] := \frac{\alpha + (\beta - \alpha) \cdot \frac{\gamma}{\lambda - 1}}{\lambda}$$

where $\sum_{\gamma=0}^{\lambda-1} \mathbf{Pr}_{\text{linear}}[\gamma] = 1$ implies $\alpha + \beta = 2$ and $1 \leq \beta \leq 2$.

In expectation

- best individual reproduced β times
- worst individual reproduced α times.



Other ranking functions

Power ranking

$$\mathbf{Pr}_{\mathsf{power}}[\gamma] := \frac{\alpha + (\beta - \alpha) \cdot \left(\frac{\gamma}{\lambda - 1}\right)^{k}}{C},$$

Geometric ranking

$$\mathbf{Pr}_{\mathsf{geom}}[\gamma] := \frac{\alpha \cdot (1 - \alpha)^{\lambda - 1 - \gamma}}{C},$$

Exponential ranking

$$\mathbf{Pr}_{\mathsf{exp}}[\gamma] := \frac{1 - e^{-\gamma}}{C},$$

where C is a normalising factor and $0 < \alpha < \beta$.

Tournament Selection

Tournament selection with tournament size k

Randomly sample a subset P' of k individuals from population P.
Select the individual in P' with highest fitness.

- Often, tournament size $k = 2$ is used.

$(\mu + \lambda)$ and (μ, λ) selection

Origins in Evolution Strategies.

$(\mu + \lambda)$-selection

Parent population of size μ.
Generate λ offspring from randomly chosen parents.
Next population is μ best among parents and offspring.

(μ, λ)-selection (where $\lambda > \mu$)

Parent population of size μ.
Generate λ offspring from randomly chosen parents.
Next population is μ best among offspring.

Selection pressure

Degree to which selection emphasizes the better individuals.
How can selection pressure be measured and adjusted?

Take-over time τ^* [Goldberg and Deb, 1991, Bäck, 1994].

1. Initial population with unique fittest individual x^*.
2. Apply selection operator reapeadly with no other operators.
3. τ^* is number of generations until population consists of x^* only.

Higher take-over time $\rightarrow$ lower selection pressure.

Fitness prop.	$\tau^* \approx \frac{\lambda \ln \lambda}{c}$	assuming fitness $f(x) = \exp(cx)$
Linear ranking	$\tau^* \approx \frac{2 \ln(\lambda - 1)}{\beta - 1}$	$1 < \beta < 2$
Tournament	$\tau^* \approx \frac{\ln \lambda + \ln \ln \lambda}{\ln k}$	tournament size k
(μ, λ)	$\tau^* = \frac{\ln \lambda}{\ln(\lambda / \mu)}$	

Summary

- Exploration and exploitation
- Mutation operators
- Recombination operators
- Selection mechanisms
- Selection pressure

Main References

Bäck, T. (1994).
Selective pressure in evolutionary algorithms: A characterization of selection mechanisms.
In *Proceedings of the 1st IEEE Conf. on Evolutionary Computation*, pages 57–62. IEEE Press.

Goldberg, D. E. and Deb, K. (1991).
A comparative analysis of selection schemes used in genetic algorithms.
In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann.