

Genetic Algorithms

Dr. Mahmoud Nabil Mahmoud
mnmahmoud@ncat.edu

North Carolina A & T State University

September 6, 2021

Outline

- 1 Practical Consideration
 - Objective Function vs Fitness Function
 - Fitness Scaling
 - Coding
 - A Multiparameter, Mapped, Fixed-point Coding
 - Discretization
 - Constraints
- 2 Revisited: Selection, Crossover, Mutation
 - Selection operator
 - Crossover operator
 - Mutation operator

Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Mapping objective function

Objective Function

Is what we are trying to optimize $g(x)$. It can be minimization or maximization.

Fitness Function

Is how we evaluate each individual in the population $f(x)$.

- We need to study how to convert from $g(x)$ to $f(x)$

Mapping objective function

How to convert minimization Objective to maximization Fitness?

Mapping objective function

How to convert minimization Objective to maximization Fitness?

$$f(x) = \begin{cases} C_{max} - g(x) & \text{if } g(x) < C_{max} \\ 0 & \text{otherwise} \end{cases}$$

C_{max} : the largest g value observed thus far the largest g value in the current population

Mapping objective function

Assuming again our problem is a maximization problem. But $g(x)$ some time output a negative values. How to deal with Negative values of $g(x)$?

Mapping objective function

Assuming again our problem is a maximization problem. But $g(x)$ some time output a negative values. How to deal with Negative values of $g(x)$?

$$f(x) = \begin{cases} C_{min} + g(x) & \text{if } g(x) + C_{min} > 0 \\ 0 & \text{otherwise} \end{cases}$$

C_{min} : absolute value the worst u value in the current or last k generations

Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Fitness Scaling

Premature convergence

Premature convergence means that a population for an optimization problem converged too early, resulting in being suboptimal.

- At the beginning of the GA run, there may be a very high fitness individual i , that biases search towards i . (Biased Selection Pressure)

Fitness Scaling

Premature convergence

Premature convergence means that a population for an optimization problem converged too early, resulting in being suboptimal.

- At the beginning of the GA run, there may be a very high fitness individual i , that biases search towards i . (Biased Selection Pressure)
- This results in a loss of diversity and the survival of the fittest becomes a random walk

Fitness Scaling

Premature convergence

Premature convergence means that a population for an optimization problem converged too early, resulting in being suboptimal.

- At the beginning of the GA run, there may be a very high fitness individual i , that biases search towards i . (Biased Selection Pressure)
- This results in a loss of diversity and the survival of the fittest becomes a random walk
- Near the end of a run, when the population is converging, there may also not be much separation among individuals in the population.

Fitness Scaling

Premature convergence

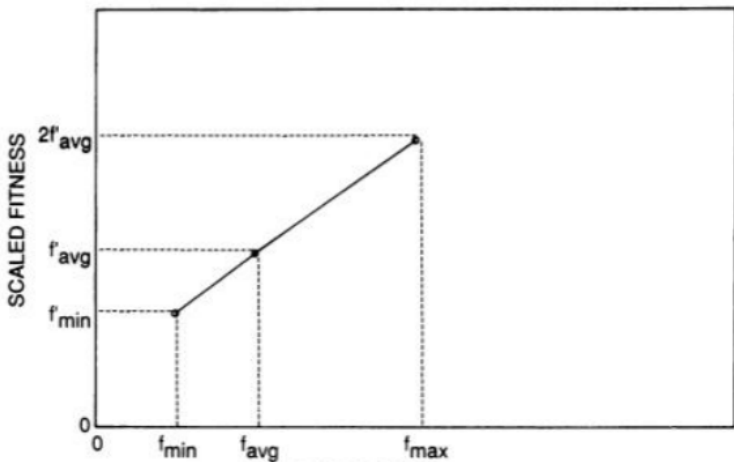
Premature convergence means that a population for an optimization problem converged too early, resulting in being suboptimal.

- At the beginning of the GA run, there may be a very high fitness individual i , that biases search towards i . (**Biased Selection Pressure**)
- This results in a loss of diversity and the survival of the fittest becomes a random walk
- Near the end of a run, when the population is converging, there may also not be much separation among individuals in the population.
- Neither is desirable. Thus we may want to scale the fitness so that selection pressure remains the same throughout the run.

Fitness Scaling

- One useful scaling procedure is linear scaling of fitness.

$$f' = af + b$$



Fitness Scaling

- We want to maintain a certain relationship between the maximum fitness individual in the population and the average population fitness. Let this be expressed by the following constraint equations:

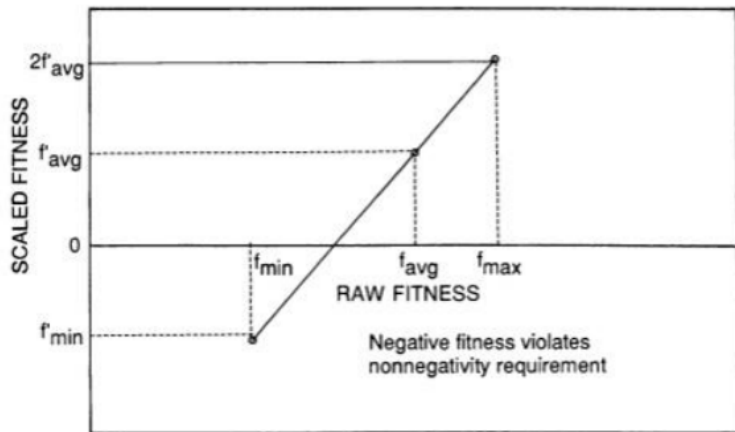
$$\begin{aligned}f'_{max} &= f_{avg} * C_s \\ f'_{avg} &= f_{avg}\end{aligned}$$

- Where f' is the scaled maximum fitness, f_{avg} is the average fitness of the population, and C_s is a scaling constant that specifies the expected number of copies of the best individual in the next generation.
- Increasing C_s will increase selection **pressure** (bias towards best individual and quicker convergence)
- Decreasing C_s will decrease selection pressure.

Example

Assume a population with $f_{avg} = 30$, $f_{min} = 10$, $f_{max} = 100$, and given a selection pressure is 2, calculate the values for a and b?

Fitness Scaling



Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- **Coding**
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Coding

- Although binary codings have given us some flexibility, it do not provide the variety of options we require to tackle the spectrum of problems we face in science, business, and engineering.
- **Principle of meaningful building blocks (similarities)**
Select a coding so that short, low order schemata are relevant to the underlying problem

Coding

- Although binary codings have given us some flexibility, it do not provide the variety of options we require to tackle the spectrum of problems we face in science, business, and engineering.
- **Principle of meaningful building blocks (similarities)**
Select a coding so that short, low order schemata are relevant to the underlying problem

Fundamental theory suggests that small alphabets are good because they maximize the number of schemata available for genetic processing

Example

Assume we would like to optimize $f(x) = x^2$ in range $[0, 31]$ finding x that gives

TABLE 3.1 Comparison of Binary and Nonbinary String Populations

Binary String	Value X	Nonbinary String	Fitness
0 1 1 0 1	13	N	169
1 1 0 0 0	24	Y	576
0 1 0 0 0	8	I	64
1 0 0 1 1	19	T	361

Number of possible individuals if we used binary coding is 31 each with length $\ell = 5$

Example

Assume we would like to optimize $f(x) = x^2$ in range $[0, 31]$ finding x that gives

TABLE 3.2 Binary and Nonbinary Coding Correspondence

Coding Correspondence Table

Binary	Nonbinary
0 0 0 0 0	A
0 0 0 0 1	B
.	.
.	.
.	.
1 1 0 0 1	Z
1 1 0 1 0	1
1 1 0 1 1	2
.	.
.	.
.	.
1 1 1 1 1	6

Number of possible individuals if we used binary coding is 31 each with length $\ell = 1$

Example

- Assuming alphabet cardinality k
- Remember number of possible schemata is $(k + 1)^\ell$
- The different alphabet cardinalities require different string lengths.
- Binary alphabet gives the max number of schemata
- $(2 + 1)^\ell > (k + 1)^\ell$

TABLE 3.1 Comparison of Binary and Nonbinary String Populations

Binary String	Value X	Nonbinary String	Fitness
0 1 1 0 1	13	N	169
1 1 0 0 0	24	Y	576
0 1 0 0 0	8	I	64
1 0 0 1 1	19	T	361

TABLE 3.2 Binary and Nonbinary Coding Correspondence

Coding Correspondence Table

Binary	Nonbinary
0 0 0 0 0	A
0 0 0 0 1	B
.	.
.	.
1 1 0 0 1	Z
1 1 0 1 0	1
1 1 0 1 1	2
.	.
.	.
1 1 1 1 1	6

Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

A Multiparameter, Mapped, Fixed-point Coding

- Some time we would like our decoding of a binary individual to be in a specified interval $[U_{min}, U_{max}]$ instead of $[0, 2^l]$
- The precision of this mapped coding may be calculated:

$$\pi = \frac{U_{max} - U_{min}}{2^l - 1}$$

SINGLE U_1 PARAMETER ($l_1 = 4$)

0 0 0 0 \rightarrow U_{min}
 1 1 1 1 \rightarrow U_{max}
 others map linearly in between

MULTIPARAMETER CODING (10 parameters)

0 0 0 1	0 1 0 1	1 1 0 0	1 1 1 1
U_1	U_2	U_9	U_{10}

Ex.

Find the real values of $x = 0110111$ after mapping it into the range $(-127, 128)$?

Outline

1 Practical Consideration

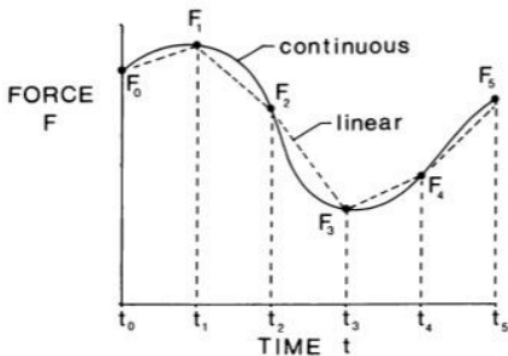
- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- **Discretization**
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Discretization

Suppose the following optimization problem: finding a force as a function of time $f(t)$ to minimize the time of travel of a bicycle between two points.



Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Constraints

- Constraints are usually classified as equality or inequality in the optimization problem.

Ex.

$$\begin{array}{ll} \text{minimize} & g(x) \\ \text{subject to} & b_i(x) > 0 \quad i = \{0, 1, \dots, n\} \end{array}$$

- Naive solution is we can assign a fitness 0 if any constraints is violated for a given individual
- Many practical problems are highly constrained where finding a feasible point is almost as difficult as finding the best solution.

Constraints

- The idea is to degrade the fitness ranking in relation to the degree of constraint violation.
- This is done using a **penalty method**

$$\text{minimize } g(x) + r \sum_{i=1}^n \phi[b_i(x)]$$

where $\phi[.]$ is a penalty function

- r is the weight of violating a constraint.

Outline

- 1 Practical Consideration
 - Objective Function vs Fitness Function
 - Fitness Scaling
 - Coding
 - A Multiparameter, Mapped, Fixed-point Coding
 - Discretization
 - Constraints
- 2 Revisited: Selection, Crossover, Mutation
 - Selection operator
 - Crossover operator
 - Mutation operator

Outline

1 Practical Consideration

- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

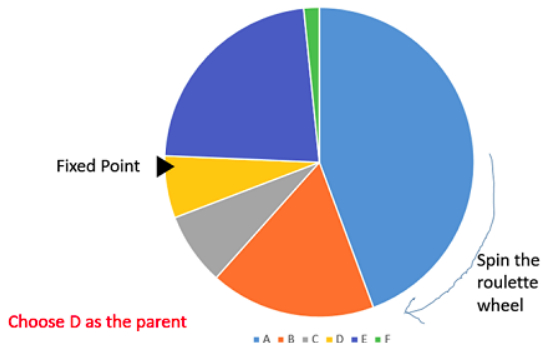
- Selection operator
- Crossover operator
- Mutation operator

Selection Operator

- Parent selection is very crucial to the **convergence rate of the GA** as good parents drive individuals to a better and fitter solutions.
- **Maintaining good diversity** in the population is extremely crucial for the success of a GA.

Roulette Wheel Selection

- In a roulette wheel selection, the circular wheel is divided as described before.
- A fixed point is chosen on the wheel circumference as shown and the wheel is rotated



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

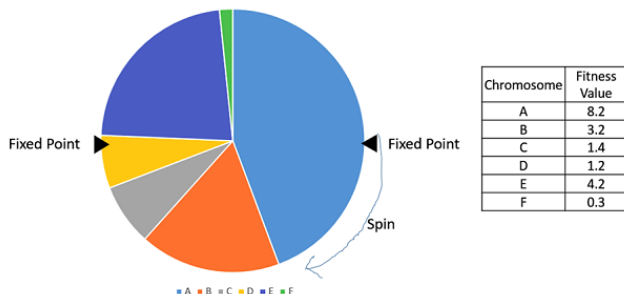
Implementing the roulette wheel selection

Implementation wise, we use the following steps

- 1 Calculate $S =$ the sum of a fitnesses.
- 2 Generate a random number between 0 and S .
- 3 Starting from the top of the population, keep adding the fitnesses to the partial sum P , till $P < S$.
- 4 The individual for which P exceeds S is the chosen individual.

Stochastic Universal Sampling (SUS)

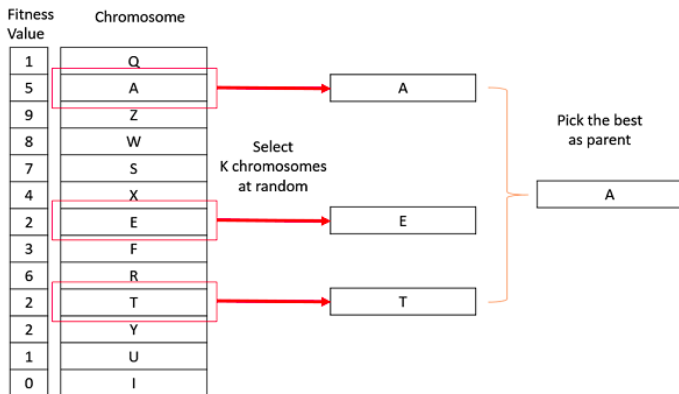
Stochastic Universal Sampling is quite similar to Roulette wheel selection, however instead of having just one fixed point, we have multiple fixed points.



Encourages the highly fit individuals to be chosen at least once.
(i.e., Adjust selection Pressure away from highly fit)

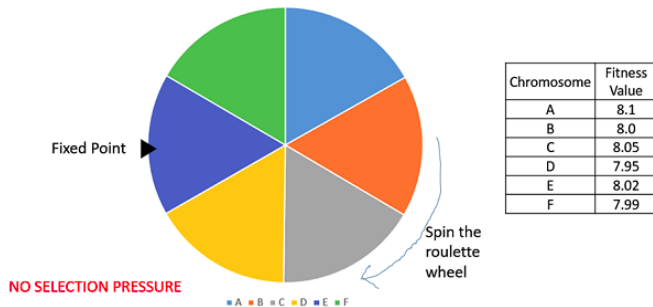
Tournament Selection

In K-Way tournament selection, **we select K individuals** from the population at **random** and select the **best** out of these to become a parent.



Rank Selection

Mostly used when the individuals in the population have very close fitness values (this happens usually at the end of the run).



Rank Selection

- 1 We remove the concept of a fitness value while selecting a parent.
- 2 Every individual in the population is ranked according to their fitness.
- 3 The selection of the parents depends on the rank of each individual and not the fitness.

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

Outline

1 Practical Consideration

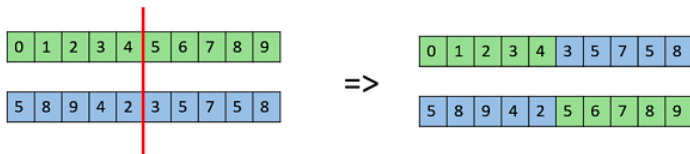
- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- **Crossover operator**
- Mutation operator

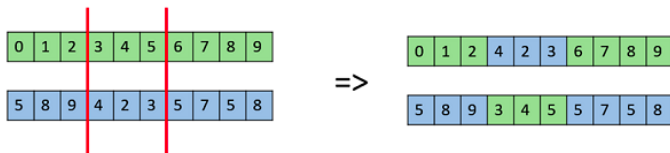
One Point Crossover

In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.



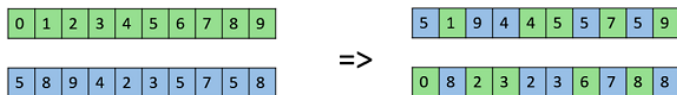
Multi Point Crossover

Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



Uniform Crossover

We treat each gene separately. In this, we essentially flip a coin for each chromosome to decide whether or not it will be included in the off-spring.



We can also bias the coin to one parent, to have more genetic material in the child from that parent.

Outline

1 Practical Consideration

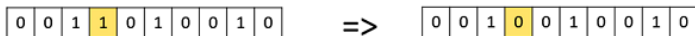
- Objective Function vs Fitness Function
- Fitness Scaling
- Coding
- A Multiparameter, Mapped, Fixed-point Coding
- Discretization
- Constraints

2 Revisited: Selection, Crossover, Mutation

- Selection operator
- Crossover operator
- Mutation operator

Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

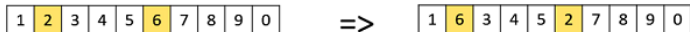


Random Resetting

Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

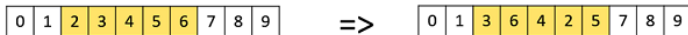
Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.



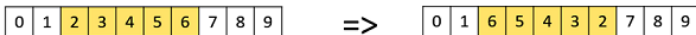
Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.



Inversion Mutation

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.



References

- Goldenberg, D.E., 1989. Genetic algorithms in search, optimization and machine learning.
- Michalewicz, Z., 2013. Genetic algorithms + data structures = evolution programs. Springer Science & Business Media



Questions 

