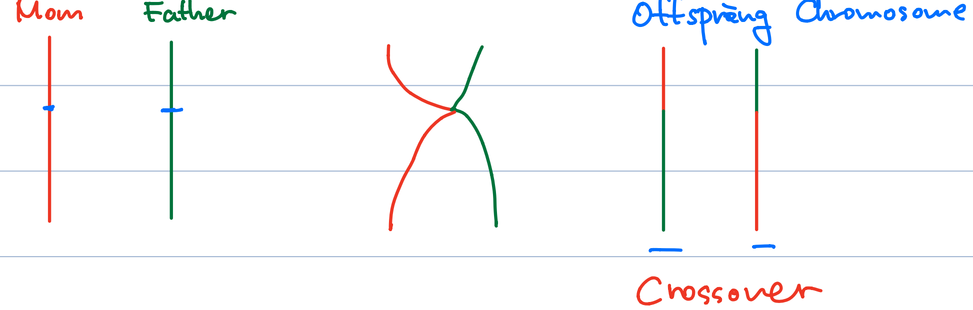
Overview

- Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA.
- Genetic algorithms are based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations:
  - Individuals → Model in a population compete for resources and mates.
  - Those individuals most successful in each 'competition' will ↳ choose better Model. produce more offspring than those individuals that perform poorly.
  - Genes from 'good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
  - Thus each successive generation will become more suited to their environment.

Chromosome

23 x 2



* Crossover

* Mutation.

(Evolution Theory)

* Selection

Genetic Algorithm

Search Space

(0 : not included in model
1 : included in model.)

27 variables => ~~the~~
make sequences using model inclusion index.

- A population of individual is maintained within search space for a genetic algorithm, each representing a possible solution to a given problem.

Chromosome
↪ Compete.

- Each individual is coded as a finite length vector of components, or variables, in terms of some alphabet, usually the binary alphabet $\{0, 1\}$.
- To continue the genetic analogy these individuals are likened to chromosomes and the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables).
- A fitness score is assigned to each solution representing the abilities of an individual to 'compete'. The individual with the optimal (or generally near optimal) fitness score is sought.

Search Space

- The genetic algorithm aims to use selective 'breeding' of the solutions to produce 'offspring' better than the parents by combining information from the chromosomes.

  1 individual chromosome
  2 individual
  3 individual

  Compete with fitted values

- The genetic algorithm maintains a population of n chromosomes (solutions) with associated fitness values. Parents are selected to mate, on the basis of their fitness, producing offspring via a reproductive plan.

  n individual

- Consequently highly fit solutions are given more opportunities to reproduce, so that offspring inherit characteristics from each parent.
- As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size. Individuals in

  # individual = n.

  the population die and are replaced by the new solutions, eventually creating a new generation once all mating opportunities in the old population have been exhausted.

Implementation Details

After an initial population is randomly generated, the algorithm evolves the through three operators:

1. *selection* which equates to survival of the fittest;
2. *crossover* which represents mating between individuals;
3. *mutation* which introduces random modifications.

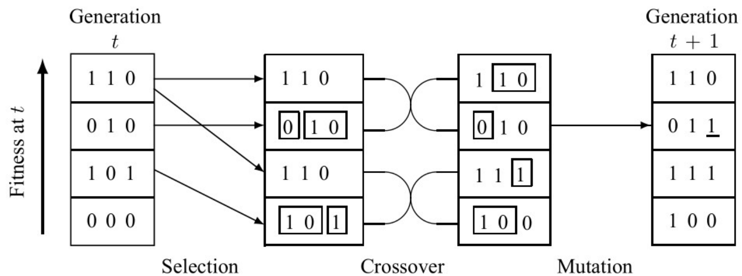


FIGURE 3.5 An example of generation production in a genetic algorithm for a population of size $P = 4$ with chromosomes of length $C = 3$. Crossovers are illustrated by boxing portions of some chromosomes. Mutation is indicated by an underlined gene in the final column.

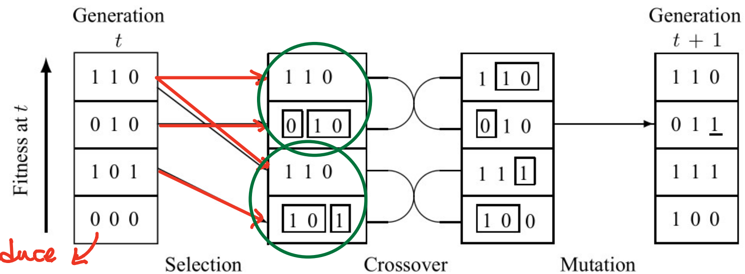
Selection Operator

fitted value (f(θ))

- Key Idea: Give preference to better individuals, allowing them to pass on their genes to the next generation.
- The goodness of each individual depends on its fitness.
- Fitness may be determined by a function $f(\theta)$.

population size : 4
Chromosome : 3

1. 4 man
2. 3 female



cannot produce offspring

FIGURE 3.5 An example of generation production in a genetic algorithm for a population of size $P = 4$ with chromosomes of length $C = 3$. Crossovers are illustrated by boxing portions of some chromosomes. Mutation is indicated by an underlined gene in the final column.

Crossover Operator

Determine where we crossover.

- Prime distinguished factor of genetic algorithm from other optimization techniques.

Choice 1-2
Choice 2-3) Crossover Site ⇒ Determine randomly.

- Two individuals are chosen from the population using the selection operator.
- A crossover site along the bit strings is randomly chosen.
- The values of the two strings are exchanged up to this point.

reproduce 4 individual.

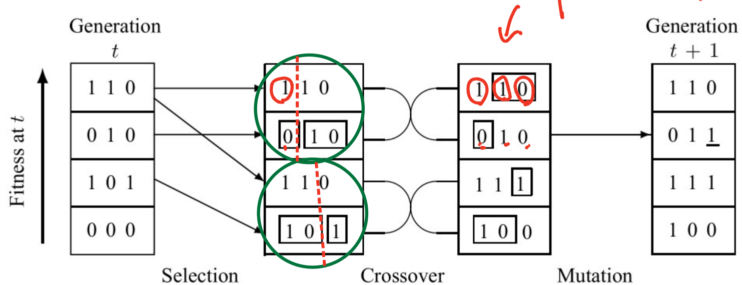


FIGURE 3.5 An example of generation production in a genetic algorithm for a population of size $P = 4$ with chromosomes of length $C = 3$. Crossovers are illustrated by boxing portions of some chromosomes. Mutation is indicated by an underlined gene in the final column.

Crossover Operator

- If $S_1 = 000000$ and $S_2 = 111111$ and the crossover point is 2 then $S_1' = 110000$ and $S2' = 001111$.
- The two new offspring created from this mating are put into the next generation of the population.
- By recombining portions of good individuals, this process is likely to create even better individuals.

better-fitted value.

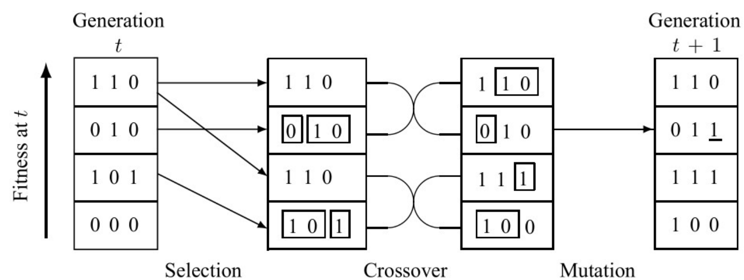


FIGURE 3.5 An example of generation production in a genetic algorithm for a population of size $P = 4$ with chromosomes of length $C = 3$. Crossovers are illustrated by boxing portions of some chromosomes. Mutation is indicated by an underlined gene in the final column.

Mutation Operator

individual Chromosome 000000000000

gene

- With some low probability, a portion of the new individuals will have some of their bits flipped.
  mutation rate (adding randomness)
- Its purpose is to maintain diversity within the population and inhibit premature convergence.
  ⇒ allow uphill move in simulated annealing
  ⇒ make it search wider parameter space
  the convergence of
- Mutation alone induces a random walk through the search space. ⇒ avoid the local optimum
- Mutation and selection (without crossover) create a parallel, noise-tolerant, hill-climbing algorithms.

select. mutation site randomly

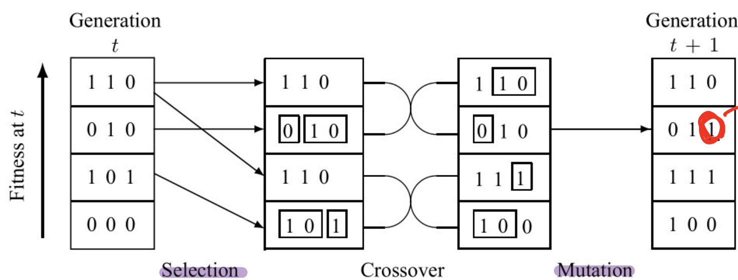


FIGURE 3.5 An example of generation production in a genetic algorithm for a population of size $P = 4$ with chromosomes of length $C = 3$. Crossovers are illustrated by boxing portions of some chromosomes. Mutation is indicated by an underlined gene in the final column.

Effect of Genetic Operator

- Using selection alone will tend to fill the population with copies of the best individual from the population.
- Using selection and crossover operators will tend to cause the algorithms to converge on a good but sub-optimal solution.
- Using mutation alone induces a random walk through the search space.
- Using selection and mutation creates a parallel, noise-tolerant, hill climbing algorithm.

Algorithm

1. Randomly initialized population
2. Determine fitness of population.
3. Repeat
    1. Select parents from population.
    2. Perform crossover on parents creating population.
    3. Perform mutation of population.
    4. Determine fitness of population.

Example: Baseball Salaries

hw

100 generation

Population size: 20

- One hundred generations of size $P = 20$ were used.
- Binary inclusion-exclusion alleles were used for each possible predictor yielding chromosomes of length $C = 27$.
- The starting generation consisted of purely ~~random individuals.~~
- A rank-based fitness function was used. One parent was selected with probability proportional to this fitness.

  20/sum(1:20) best   19/sum(1:20) second
- The other parent was selected independently, purely at random.
- Breeding employed simple crossover.
- A 1% mutation rate was randomly applied independently to each locus.

Example: Baseball Salaries

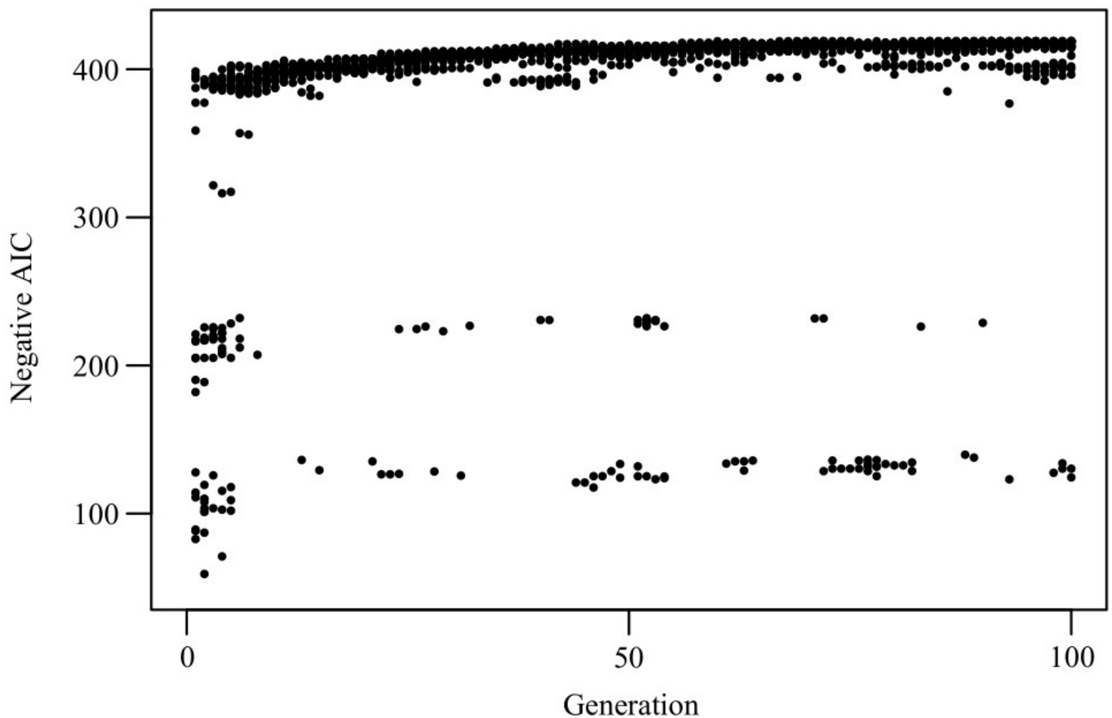


FIGURE 3.6 Results of a genetic algorithm for Example 3.5.