Carl Stevenson

Unit 4 Notes

* Gregor Mendel -genetics
  + have 2 sources of parental dna
    - -male and female
* First steps
  + design a representation of the problem
  + design a fitness function
    - evaluates the goodness of the games
* Basic genetic algorithm
  + start with initial population
  + check for fitness
  + breed fit individuals
* How to choose individuals to breed?
  + Most fit?
  + Ranked by fitness?
  + Most diverse?
* Standard method
  + F(i) = Q(i)/sum(j)Q(j)
  + randomly select individuals with relative quality
  + have them reproduce
  + Similar to Parallel Hill Climbing
* Fitness landscape
  + differs from heuristic state space search
    - no estimates of 'effort' to move to new state
    - no strict ordering of states
  + Has peaks, valleys, discontinuities
    - both real and introduced
    - binary vs gray coding
* Quality rank method
  + rank each with fixed probability
  + favors fit individual
  + has some diversity, but strongly favors the fit
* Cake example
  + How much flour and how much sugar to make a cake?
  + Has a sweet spot in the middle favoring a certain ratio
* Rank space method
  + D(i) = sum 1/d(i)^2
  + Uses distance or (diversity) to decide how to pick
  + D=diversity ranking
  + Encourages diversity
  + Combine the genetic methods, encourages diversity and the fit
* New Individuals?
  + During reproduction, there are two operators:
    - Crossover
    - Mutation
  + During meiosis, chromosomes interact with each other, swap random segments of code
  + Mutation occurs rarely.
    - Sometimes, good mutations happens
* Binary vs Gray Coding
  + binary, increasingly large gaps between bits as it gets larger
  + gray codes have lower hamming distances
* Reproduction
  + population>fitness function>selection>crossover>mutation>population
* Genesis System
  + Genes can be bit sequence or multiple floats
  + Can specify or randomly initialize population
  + Selection by standard or ranked method
* Using Genesis for blackjack
  + Fitness function:
    - Use bit vector strategy
    - Play blackjack against dealer
    - Report losing percentage
* Genetic Programming
  + Represent the problem
    - Expression tree
  + Two operators
    - Crossover
    - Mutation
      * Changing operators from one to another
      * Changing operand to another from the problem
  + Fitness function
    - Test on sample instances, compute the similarity to the desired value
    - Determines the fitness of the genetic