Genetic Algorithms

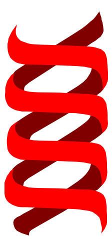
- What are genetic algorithms?
- Genetic algorithm components
- Example problem
- Common questions about genetic algorithms
- Example applications
- Important references for genetic algorithms
- Ten summary points
- Demonstrations with Evolver

How large is the decision space?

- **If we were to look at every alternative, what would we have to do? Of course, it depends.....**

- **Think: enzymes**
  - Catalyze all reactions in the cell
  - Biological enzymes are composed of amino acids
  - There are 20 naturally-occurring amino acids
  - Easily, enzymes are 1000 amino acids long
  - 20^1000 = (2^1000)(10^1000) ≈ 10^1300

- **A reference number, a benchmark:**



10^80 ù **number of atomic particles in universe**

How large is the decision space?

- **Problem: Design an icon in black & white How many options?**
  - Icon is 32 x 32 = 1024 pixels
  - Each pixel can be on or off, so 2^1024 options
  - 2^1024 ≈ (2^20)^50 ≈ (10^6)^50 = 10^300
- **Police faces**
  - 10 types of eyes
  - 10 types of noses
  - 10 types of eyebrows
  - 10 types of head
  - 10 types of head shape
  - 10 types of mouth
  - 10 types of ears
  - but already we have 10^7 faces

How large is the decision space?

- **Arranging n things**
- **Putting 100 things in the right order**
  - 100!
- **How many ways?**
  - n! = n(n-1)(n-2).....1
  - ? (fact 3)
  - 6
  - ? (fact 5)
  - 120
  - ? (fact 100)
  - 93326215443944152681699238856266700490715968
    26438162146859296389521759999322991560894146
    397615651828625369792082722375825118521091686
    4000000000000000000000000

Heuristic Search

- **Decision spaces are typically extremely large**
- **Some algorithms are available for effectively searching certain “nice” decision spaces (e.g., linear programming)**
- **But most problems are “not so nice” and we lack programmed solution techniques**
- **Yet humans routinely solve such problems**
- **Amazing! What’s going on?**

Heuristic Search

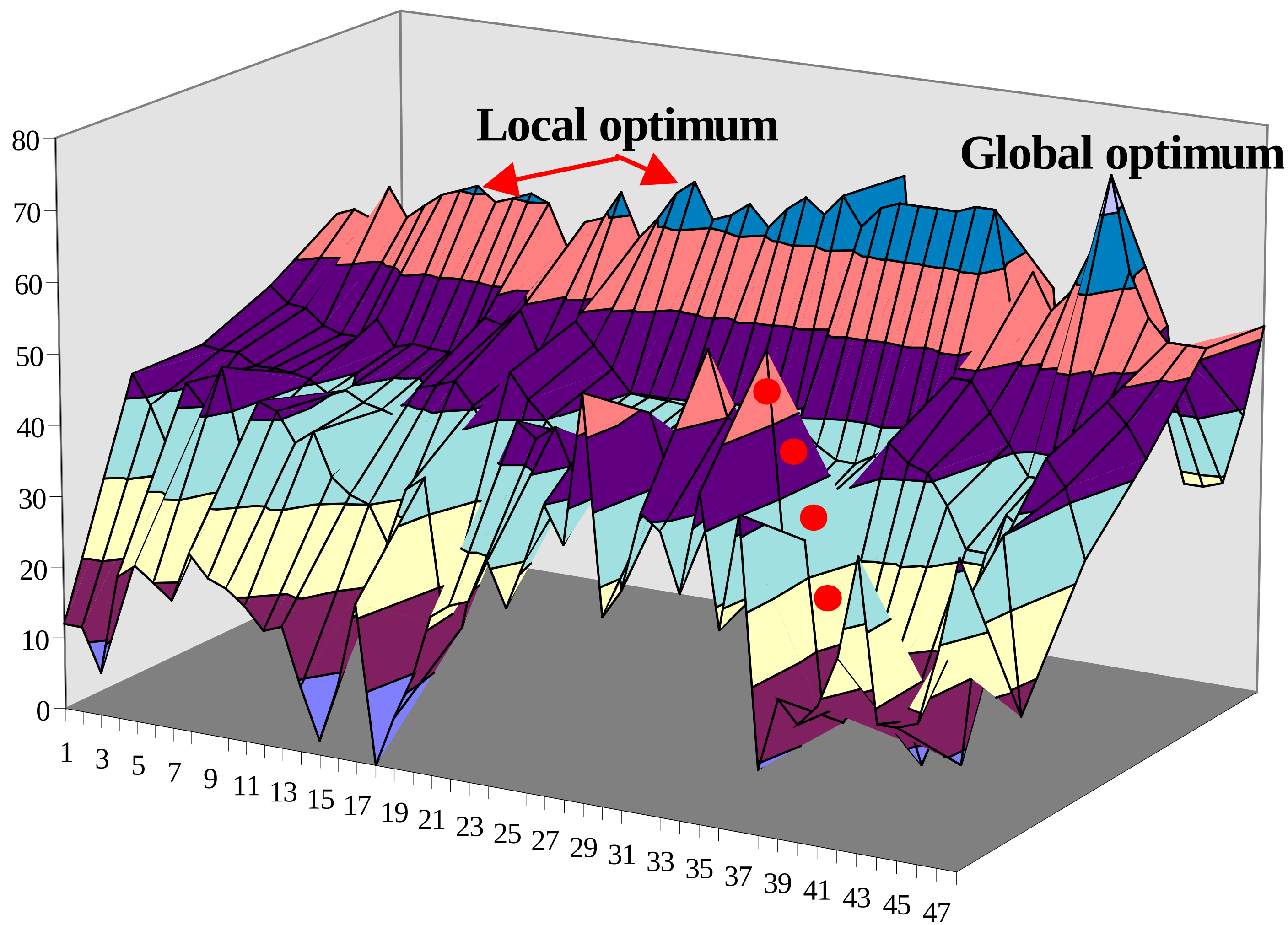
- **Build rule-based expert systems**
  - Performance so far not super impressive (somewhat impressive)
  - Doesn't show what's needed. Only shows that there exist such rules, not how they are found or how cognition could work. (rule-governed vs rule-described)
  - Expensive and very time-consuming in general

- **Build programs that acquire rules automatically**
  - Genetic algorithms
  - Performance so far is very impressive (e.g., suspect ID)
  - Still time-consuming, but can hope for a general architecture

What are genetic algorithms?

- **Genetics and evolution**
  - survival of the fittest gene pool - natural selection
  - meiosis involves exchange of genetic material - crossover
  - maintain genetic diversity - mutation
- **Adaptive systems**
  - finite automata / machine learning / history of GAs
- **Efficient search algorithm for complex problems**
  - randomly generate a population of potential solutions
  - calculate fitness of each potential solution
  - allow best individuals to breed - selection and crossover (p<.6)
  - allow low probability mutations (p<.007) to maintain diversity
  - check for convergence of populations in the gene pool

Genetic algorithms vs hill climbing



Genetic Algorithm Components

- **Selection**
  - determines how many and which individuals breed
  - premature convergence sacrifices solution quality for speed
- **Crossover**
  - select a random crossover point
  - successfully exchange substructures
  - 00000 x 11111 at point 2 yields 00111 and 11000
- **Mutation**
  - random changes in the genetic material (bit pattern)
  - for problems with billions of local optima, mutations help find the global optimum solution
- **Evaluator function**
  - rank fitness of each individual in the population
  - simple function (maximum) or complex function

Genetic Algorithms: Example Problem

Annual sales for Avoiding Extinction by JWI Publishers is 20,000 copies. Books are sold for $30.

JWI Publishers have a variable cost of $6 per book associated with producing the book.

JWI Publishers have two fixed cost components. Overhead, royalties and other costs total $350,000. Setup cost per printing is $6,000. Thus, 4 quarterly printing would cost 4 x $6,000 = $24,000.

Genetic Algorithms: Example Problem

Annual sales for Avoiding Extinction by JWI Publishers is 20,000 copies. Books are sold for $30.

JWI Publishers have a variable cost of $6 per book associated with producing the book.

JWI Publishers have two fixed cost components. Overhead, royalties and other costs total $350,000. Setup cost per printing is $6,000. Thus, 4 quarterly printing would cost 4 x $6,000 = $24,000.

EOQ model yields

$$N_{unit} = \sqrt{\frac{2PU}{C}} = \sqrt{\frac{2 * \$6,000 * 20,000}{\$6}}$$

$$N_{unit} = 6,325 books / order$$

Genetic Algorithms: Example Problem

Quantity	6,326
Annual Book Sales	20,000
Number of Setups	3.16
Setup Cost	$6,000
Selling Price	$30
Variable Book Costs	$6
Total Revenues	$600,000
Variable Costs	$18,978
Fixed Costs	$18,969
Other Costs	$350,000
Profit	$212,052.67

Genetic Algorithms: Example Problem

- **Choose the problem representation**
  - 14 digits binary string allows an order size from 1 to 16,384 (2^{14})
- **Initialize the population**
  - randomly generate 100 - 200 individual strings of length 14
- **Calculate fitness for each individual**
  - convert string to decimal and determine profit with that order size
  - 00100010011010 = 2,202

Total Revenue	**$600,000**
Variable costs	**6 * 2,202 / 2**
Setup costs	**20,000 / 2,202 * $6,000**
Fixed costs	**$350,000**
Profit	**$188,898**

Genetic Algorithms: Example Problem

Perform selection

- long run survival of the fittest
- short run merely nudges population towards better performers
- replace the worst strings (bottom 5%) with copies of the best strings (top 5%), thus it would take a minimum of 20 generations before all strings are replaced - slow convergence.

Perform crossover

- randomly select two parents from the new population
- randomly determine whether to crossover (p = .6)
- if crossover, randomly select a crossover point (1-13)
- example:

  00100010011010 (2,202) x 11011001000111 (13,895)

  at 3 yields

  11000010011010 (3,655) x 00111001000111 (12,442)

Genetic Algorithms: Example Problem

Perform mutation

- bit by bit, string by string, randomly determine whether to mutate each bit using a very low probability (p=.007). If mutation rate is too high, it will prevent convergence.
- if mutation should occur, change 0 to 1 or vice versa.

Check convergence

- bias is one measure of agreement among the population
- bias assumes values between 50 and 100 percent
- bit bias
  - if 100 strings have 0 in position 1 and 100 have a 1, then the bit bias is 50%
  - a 75:25 split or a 25:75 split has a 75% bias
  - a 90:10 split or a 10:90 split has a 90% bias
- string bias is the average bias for each bit over all strings
- a population with a average bias of 95% has converged

Common Questions about Genetic Algorithms

- **Can a GA converge to a poor solution?**

  YES! Poor problem representation, premature convergence, a poor fitness evaluation algorithm, or luck of the random numbers could generate a poor solution

- **How do you know whether the GA solution is optimal or near optimal?**

  If you knew how to find the optimal solution, you would not need to use a GA. There is no guarantee that a GA will find an optimal solution. GAs find a good solution that is “better” than others.

- **Are neural networks better than GAs?**

  Neural networks require less structural knowledge. However, the type and number of node connections and hidden layers make it difficult to interpret relationships in a neural network.

  GAs require a starting framework to setup the problem representation and calculate fitness

Genetic Algorithms: Example Applications

Criminal suspect recognition using ***Faceprints***

- Developed by psychology department of New Mexico State University
- Uses GAs to aid witnesses in the identification of criminal suspects
- Easy to identify criminal from photo but hard to describe features
- Difficult to generate even from computer library of visual features
- GA approach:
  - randomly generate 20 faces on a computer screen
  - witness rates each face on a 10 point scale
  - GA generates additional faces from 5 building blocks: eyes, mouth, nose, hair and chin. Each is a 7 bit string.
  - The five features are coded as a 35-bit binary string consisting of five 7-bit parameters (34 billion faces are possible)
  - Witness evaluates successive generations with 10 point scale
  - Convergence often occurs after 20 generations

Genetic Algorithms: Example Applications

Currency trading

- Prediction Company, Santa Fe New Mexico, developed GAs for currency trading
- left-hand sides of the rule predict when time-series data enter specific regions
- right-hand sides of the rule predict whether the time-series will go up or down
- objective function measures mean-squared error

Complex data base queries

- Boolean operators, multiple algebraic and logical relationships create large complex search spaces
- Bennett K., Ferris M.C., & Ioannidis Y.E., (1989) developed a novel encoding of chromosomes to represent a binary tree query graph.
- System was able to find solutions when traditional search methods yielded too many non-relevant articles

Genetic Algorithms: Example Applications

Inter-office fiber-optic networks

- US West designs expansion plans for large networks
  - old method began with designer intuition and experience
  - multiple rings of interconnected fiber-optic cable with a maximum of 48 nodes per ring
  - tested design with a network simulation tool
- GA approach used a mutation operator to determine whether to expand the existing ring or add a new ring
  - first, fitness uses computer simulation to test performance
  - second, fitness uses cost to rank best performers
- Saves $1-$10 million per design
  - design time cut from 2 person-months to 2 person-days
  - total 6 years savings could exceed $100 million

Genetic Algorithms: Summary

- **Field is not new. Holland's work began in 1970s.**
- **Most of the work has been done in computer science & engineering - not business applications**
- **Translate problem into a string representation - often binary numbers (11000)**
- **Difficult to perform translation for some problems**
- **Little knowledge at startup - randomly generated population of individuals**

Genetic Algorithms: Summary

- **Must be able to calculate fitness of each individual in the population**
- **Potential solutions with greater fitness have greater priority in subsequent generations**
- **Crossover is similar to mating and mutation transforms a stable population to maintain diversity of the search process**
- **Steps 6-8 repeat. Eventually the population converges and the fittest solution survives**
- **GAs are not an optimization technique but often find good solutions for large complex problems**

Important References for Genetic Algorithms

- **Holland, J.H. 1975. Adaptation in natural and artificial systems. Ann Arbor, MI: The University of Michigan Press.**
  - classic technical book with lots of theorems and proofs
- **Goldberg, D.E. 1989. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.**
  - graduate textbook for a machine learning course - code in Pascal
- **Davis, L. (ed) 1991. Handbook of genetic algorithms. New York: Van Nostrand Reinhold.**
  - tutorial and case applications with code in C or Lisp

Important References for Genetic Algorithms

- **Koza, J. 1992. Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press.**
  - application of programs as bits rather than 1s and 0s
- **Bauer, R.J. Jr. 1994. Genetic algorithms and investment strategies. New York: Wiley.**
  - examples of GAs used for trading bonds and stocks
- **Karjalainen, R. and Allen F. 1994. Using genetic algorithms to find technical trading rules. Wharton working paper**