Genetic Algorithms

G5BAIM
Artificial Intelligence Methods
Graham Kendall
gxk@cs.nott.ac.uk - http://www.cs.nott.ac.uk/~gxk

Acknowledgement: Some of these slides are based on those from Wendy Williams

Genetic Algorithms

G5BAIM  Genetic Algorithms
Charles Darwin 1809 - 1882

"A man who dares to waste an hour of life has not discovered the value of life"
Genetic Algorithms

Classes of Search Techniques

- Search techniques
  - Calculus-based techniques
  - Indirect methods
  - Direct methods
  - Evolutionary algorithms
  - Evolutionary strategies
  - Genetic algorithms
- Guided random search techniques
- Enumerative techniques
- Simulated annealing
- Dynamic programming
- Calculus-based techniques
- Newton
- Fibonacci
- Evolutionary strategies
- Genetic algorithms
- Centrally parallel
- Distributed
- Steady-state
- Generational
- Parallel
- Sequential

An Abstract Example

Distribution of Individuals in Generation 0

Distribution of Individuals in Generation N
Genetic Algorithms

Components of a GA
A problem to solve, and ...

- Encoding technique \((\text{gene, chromosome})\)
- Initialization procedure \((\text{creation})\)
- Evaluation function \((\text{environment})\)
- Selection of parents \((\text{reproduction})\)
- Genetic operators \((\text{mutation, recombination})\)
- Parameter settings \((\text{practice and art})\)

Genetic Algorithms

- Based on survival of the fittest
- Developed extensively by John Holland in mid 70’s
- Based on a population based approach
- Can be run on parallel machines
- Only the evaluation function has domain knowledge
- Can be implemented as three modules; the evaluation module, the population module and the reproduction module.
- Solutions (individuals) often coded as bit strings
- Algorithm uses terms from genetics; population, chromosome and gene
Genetic Algorithms

The GA Cycle

Artificial Intelligence Methods

- reproduction
- modification

population

children

parents

modified children

evaluated children

deleted members

discard

Genetic Algorithms

Population

Artificial Intelligence Methods

Chromosome Representation

- Bit strings (0101 ... 1100)
- Real numbers (43.2 -33.1 ... 0.0 89.2)
- Permutations of element (E11 E3 E7 ... E1 E15)
- Lists of rules (R1 R2 R3 ... R22 R23)
- Program elements (genetic programming)
- ... any data structure ...
Genetic Algorithms

Reproduction

Parents are selected at random with selection chances biased in relation to chromosome evaluations.

Chromosome Modification

Modifications are stochastically triggered

Operators:
  - Mutation
  - Crossover (recombination)
The evaluator decodes a chromosome and assigns it a fitness measure.
The evaluator is the only link between a classical GA and the problem it is solving.

**Genetic Algorithms**

**Evaluation**

- evaluated children
- modified children
- evaluation

**Genetic Algorithms**

**Deletion**

- population
- discarded members
- discard

*Generational* GA: entire populations replaced with each iteration

*Steady-state* GA: a few members replaced each generation
Genetic Algorithms

A Simple Example

The Traveling Salesman Problem:

Find a tour of a given set of cities so that
• each city is visited only once
• the total distance traveled is minimized

Genetic Algorithms

Representation

Representation is an ordered list of city numbers known as an order-based GA.

1) London  3) Dunedin  5) Beijing  7) Tokyo
2) Venice  4) Singapore  6) Phoenix  8) Victoria

CityList1  (3 5 7 2 1 6 4 8)
CityList2  (2 5 7 6 8 1 3 4)
Genetic Algorithms

Mutation

Mutation involves reordering of the list:

Before: (5 8 7 2 1 6 3 4)

After: (5 8 6 2 1 7 3 4)
Genetic Algorithms

TSP30 (Performance = 941)

TSP30 (Performance = 800)
Genetic Algorithms

TSP30 (Performance = 652)

Artificial Intelligence Methods

TSP30 Solution (Performance = 420)

Artificial Intelligence Methods
### Genetic Algorithms

#### Issues for GA Practitioners

Choosing basic implementation issues:
- representation
- population size, mutation rate, ...
- selection, deletion policies
- crossover, mutation operators

**Termination Criteria**
- Performance, scalability
- Solution is only as good as the evaluation function (often hardest part)

#### Benefits of Genetic Algorithms

- Concept is easy to understand
- Modular, separate from application
- Supports multi-objective optimization
- Good for “noisy” environments
- Always an answer; answer gets better with time
- Inherently parallel; easily distributed
### Benefits of Genetic Algorithms (cont.)

- Many ways to speed up and improve a GA-based application as knowledge about problem domain is gained
- Easy to exploit previous or alternate solutions
- Flexible building blocks for hybrid applications
- Substantial history and range of use

### When to Use a GA

- Alternate solutions are too slow or overly complicated
- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved by using a GA
- Want to hybridize with an existing solution
- Benefits of the GA technology meet key problem requirements
Genetic Algorithms

Some GA Application Types

<table>
<thead>
<tr>
<th>Domain</th>
<th>Application Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>gas pipeline, pole balancing, missile evasion, pursuit</td>
</tr>
<tr>
<td>Design</td>
<td>semiconductor layout, aircraft design, keyboard configuration, communication networks</td>
</tr>
<tr>
<td>Scheduling</td>
<td>manufacturing, facility scheduling, resource allocation</td>
</tr>
<tr>
<td>Robotics</td>
<td>trajectory planning</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>designing neural networks, improving classification algorithms, classifier systems</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>filter design</td>
</tr>
<tr>
<td>Game Playing</td>
<td>poker, checkers, prisoner’s dilemma</td>
</tr>
<tr>
<td>Combinatorial</td>
<td>set covering, travelling salesman, routing, bin packing, graph colouring and partitioning</td>
</tr>
</tbody>
</table>

GA Algorithm

- Initialise a population of chromosomes
- Evaluate each chromosome (individual) in the population
  - Create new chromosomes by mating chromosomes in the current population (using crossover and mutation)
  - Delete members of the existing population to make way for the new members
  - Evaluate the new members and insert them into the population
- Repeat stage 2 until some termination condition is reached (normally based on time or number of populations produced)
- Return the best chromosome as the solution
Genetic Algorithms

GA Algorithm

1. Initialization
2. Evaluation
3. Selection
4. Recombination
5. Mutation
6. Replacement
7. Repeat steps 2–6 until a terminating condition is met.

### Artificial Intelligence Methods

1. Initialization
   The initial population of candidate solutions is usually generated randomly across the search space. However, domain-specific knowledge or other information can be easily incorporated.
3. Selection
Selection allocates more copies of those solutions with higher fitness values and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. The main idea of selection is to prefer better solutions to worse ones, and many selection procedures have been proposed to accomplish this idea, including roulette-wheel selection, stochastic universal selection, ranking selection and tournament selection, some of which are described in the next section.
Genetic Algorithms

GA Algorithm

1. Initialization
2. Evaluation
3. Selection
4. Recombination
5. Mutation
6. Replacement
7. Repeat steps 2–6 until a terminating condition is met.

4. Recombination
Recombination combines parts of two or more parental solutions to create new, possibly better solutions (i.e., offspring). There are many ways of accomplishing this (some of which are discussed in the next section), and competent performance depends on a properly designed recombination mechanism. The offspring under recombination will not be identical to any particular parent and will instead combine parental traits in a novel manner.

5. Mutation
While recombination operates on two or more parental chromosomes, mutation locally but randomly modifies a solution. Again, there are many variations of mutation, but it usually involves one or more changes being made to an individual's trait or traits. In other words, mutation performs a random walk in the vicinity of a candidate solution.
**Genetic Algorithms**

**GA Algorithm**

1. Initialization
2. Evaluation
3. Selection
4. Recombination
5. Mutation
6. Replacement
7. Repeat steps 2–6 until a terminating condition is met.

**6. Replacement**
The offspring population created by selection, recombination, and mutation replaces the original parental population. Many replacement techniques such as elitist replacement, generation-wise replacement and steady-state replacement methods are used in GAs.

---

**Genetic Algorithms**

**GA Algorithm - Evaluation**

- Responsible for evaluating a chromosome
- Only part of the GA that has any knowledge about the problem.
  The rest of the GA modules are simply operating on (typically) bit strings with no information about the problem
- A different evaluation module is needed for each problem
<table>
<thead>
<tr>
<th>Genetic Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA Algorithm - Population Module</td>
</tr>
<tr>
<td>• Responsible for maintaining the population</td>
</tr>
<tr>
<td>• Initialisation</td>
</tr>
<tr>
<td>– Random</td>
</tr>
<tr>
<td>– Known Solutions</td>
</tr>
<tr>
<td>• Deletion</td>
</tr>
<tr>
<td>– Delete-All: Deletes all the members of the current population and replaces them with the same number of chromosomes that have just been created</td>
</tr>
<tr>
<td>– Steady-State: Deletes ( n ) old members and replaces them with ( n ) new members; ( n ) is a parameter</td>
</tr>
<tr>
<td>But do you delete the worst individuals, pick them at random or delete the chromosomes that you used as parents?</td>
</tr>
<tr>
<td>– Steady-State-No-Duplicates: Same as steady-state but checks that no duplicate chromosomes are added to the population. This adds to the computational overhead but can mean that more of the search space is explored</td>
</tr>
</tbody>
</table>
Genetic Algorithms

GA Parent Selection - Roulette Wheel

- Sum the fitnesses of all the population members, TF
- Generate a random number, m, between 0 and TF
- Return the first population member whose fitness added to the preceding population members is greater than or equal to m

Roulette Wheel Selection

Genetic Algorithms

GA Parent Selection - Tournament

- Select a pair of individuals at random. Generate a random number, R, between 0 and 1. If \( R < r \) use the first individual as a parent. If the \( R \geq r \) then use the second individual as the parent. This is repeated to select the second parent. The value of \( r \) is a parameter to this method

- Select two individuals at random. The individual with the highest evaluation becomes the parent. Repeat to find a second parent
Genetic Algorithms

GA Fitness Techniques

- Fitness-Is-Evaluation: Simply have the fitness of the chromosome equal to its evaluation
- Windowing: Takes the lowest evaluation and assigns each chromosome a fitness equal to the amount it exceeds this minimum.
- Linear Normalization: The chromosomes are sorted by decreasing evaluation value. Then the chromosomes are assigned a fitness value that starts with a constant value and decreases linearly. The initial value and the decrement are parameters to the techniques

Artificial Intelligence Methods

Genetic Algorithms

GA Population Module - Parameters

- Population Size
- Elitism
Genetic Algorithms

GA Reproduction - Crossover Operators

One Point Crossover in Genetic Algorithms

Uniform Crossover in Genetic Algorithms

Partially Matched Crossover (PMX) in Genetic Algorithms

Order Based Crossover

Cycle Crossover

Artificial Intelligence Methods

Genetic Algorithms

GA Example

- Crossover probability, PC = 1.0
- Mutation probability, PM = 0.0
- Maximise \( f(x) = x^3 - 60x^2 + 900x + 100 \)
- \( 0 \leq x \leq 31 \)
- \( x \) can be represented using five binary digits

\[ f(x) = x^3 - 60x^2 + 900x + 100 \]
Genetic Algorithms

GA Example

• Generate random individuals

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Binary String</th>
<th>x</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁</td>
<td>11100</td>
<td>28</td>
<td>212</td>
</tr>
<tr>
<td>P₂</td>
<td>01111</td>
<td>15</td>
<td>3475</td>
</tr>
<tr>
<td>P₃</td>
<td>10111</td>
<td>23</td>
<td>1227</td>
</tr>
<tr>
<td>P₄</td>
<td>00100</td>
<td>4</td>
<td>2804</td>
</tr>
</tbody>
</table>

TOTAL 7718
AVERAGE 1929.50

• Choose Parents, using roulette wheel selection
• Crossover point is chosen randomly

<table>
<thead>
<tr>
<th>Roulette Wheel</th>
<th>Parent Chosen</th>
<th>Crossover Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>4116</td>
<td>P₃</td>
<td>N/A</td>
</tr>
<tr>
<td>1915</td>
<td>P₂</td>
<td>1</td>
</tr>
</tbody>
</table>
Genetic Algorithms

GA Example - Crossover

<table>
<thead>
<tr>
<th>Artificial Intelligence Methods</th>
<th>Chromosome 1</th>
<th>Chromosome 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁</td>
<td>1 0 1 1 1</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>P₂</td>
<td>0 1 1 1 1</td>
<td>0 1 1 1 1</td>
</tr>
<tr>
<td>C₁</td>
<td>1 1 1 1 1</td>
<td>0 0 1 1 1</td>
</tr>
<tr>
<td>C₂</td>
<td>0 0 1 1 1</td>
<td>0 1 1 0 0</td>
</tr>
<tr>
<td>P₃</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₄</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GA Example - After First Round of Breeding

- The average evaluation has risen
- P₂, was the strongest individual in the initial population. It was chosen both times but we have lost it from the current population
- We have a value of x=7 in the population which is the closest value to 10 we have found

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Binary String</th>
<th>x</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₁</td>
<td>11111</td>
<td>31</td>
<td>131</td>
</tr>
<tr>
<td>P₂</td>
<td>00111</td>
<td>7</td>
<td>3803</td>
</tr>
<tr>
<td>P₃</td>
<td>00111</td>
<td>7</td>
<td>3803</td>
</tr>
<tr>
<td>P₄</td>
<td>01100</td>
<td>12</td>
<td>3998</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>11735</td>
<td></td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td>2933.75</td>
<td></td>
</tr>
</tbody>
</table>
Genetic Algorithms

GA Example - Question?

- Assume the initial population was 17, 21, 4 and 28. Using the same GA methods we used above (PC = 1.0, PM = 0.0), what chance is there of finding the global optimum?

Genetic Algorithms

GA Example - Mutation

- A method of ensuring premature convergence does not occur
- Usually set to a small value
- Dynamic mutation and crossover rates
Genetic Algorithms

Try Previous Exam Question

- Question and answer are available from the course web site

<table>
<thead>
<tr>
<th>Artificial Intelligence Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic Algorithms Representation Examples</td>
</tr>
<tr>
<td>• Both of these papers are available on the course web site</td>
</tr>
</tbody>
</table>
Genetic Algorithms

<table>
<thead>
<tr>
<th>G5BAIM</th>
<th>Artificial Intelligence Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graham Kendall</td>
<td><em>End of Genetic Algorithms</em></td>
</tr>
</tbody>
</table>

Artificial Intelligence Methods

3/7/2010