History Of Genetic Algorithms

- "Evolutionary Computing" was introduced in the 1960s by **I. Rechenberg**.

- John Holland wrote the first book on Genetic Algorithms 'Adaptation in Natural and Artificial Systems' in 1975.

- In 1992 **John Koza** used genetic algorithm to evolve programs to perform certain tasks. He called his method "Genetic Programming".

Darwin's Theory of Evolution

"problems are solved by an evolutionary process resulting in a best (fittest) solution (survivor) ,

-In Other words, the solution is evolved"

1. **Inheritance – Offspring acquire characteristics**
2. **Mutation – Change, to avoid similarity**
3. **Natural Selection – Variations improve survival**
4. **Recombination - Crossover**

Chromosome

All Living organisms consists of cells. In each cell there is a same set of Chromosomes.

Chromosomes are strings of DNA and consists of genes, blocks of DNA.

Each gene encodes a trait, for example color of eyes.

Reproduction

During reproduction, recombination (or crossover) occurs first. Genes from parents combine to form a whole new chromosome. The newly created offspring can then be mutated. The changes are mainly caused by errors in copying genes from parents.

The fitness of an organism is measure by success of the organism in its life (survival)

Operation of Genetic Algorithms

Two important elements required for any problem before a genetic algorithm can be used for a solution are

Method for representing a solution

ex: string of bits, numbers, character

Method for measuring the quality of any proposed solution, using fitness function

ex: Determining total weight

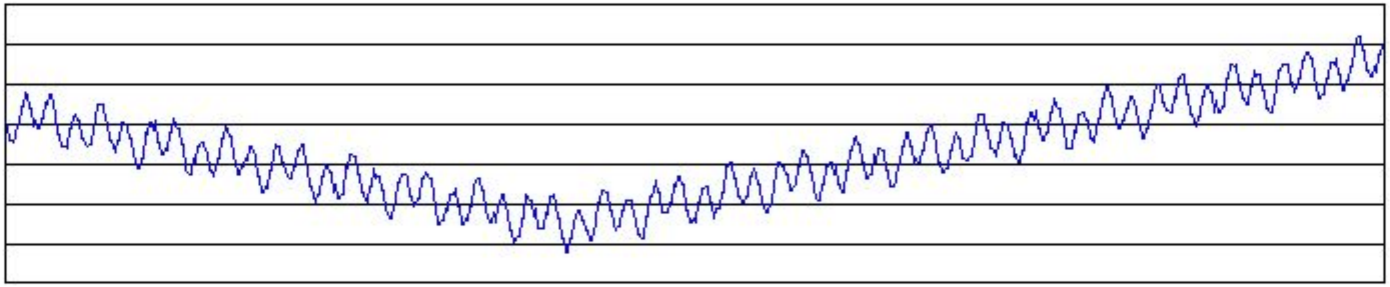
Sequence of steps

1. Initialization
2. Selection
3. Reproduction
4. Termination

Initialization

Initially many individual solutions are randomly generated to form an initial population, covering the entire range of possible solutions (the search space)

Each point in the search space represents one possible solution marked by its value(fitness)



There are no of ways in which we would find a suitable solution and they don't provide the best solution. One way of finding solution from search space is Genetic Algorithms.

Selection

A proportion of the existing population is selected to bread a new bread of generation.

Reproduction

Generate a second generation population of solutions from those selected through genetic operators: crossover and mutation.

Termination

A solution is found that satisfies minimum criteria

Fixed number of generations found

Allocated budget (computation, time/money) reached

The highest ranking solution’s fitness is reaching or has reached

Simple Example for Genetic Algorithms

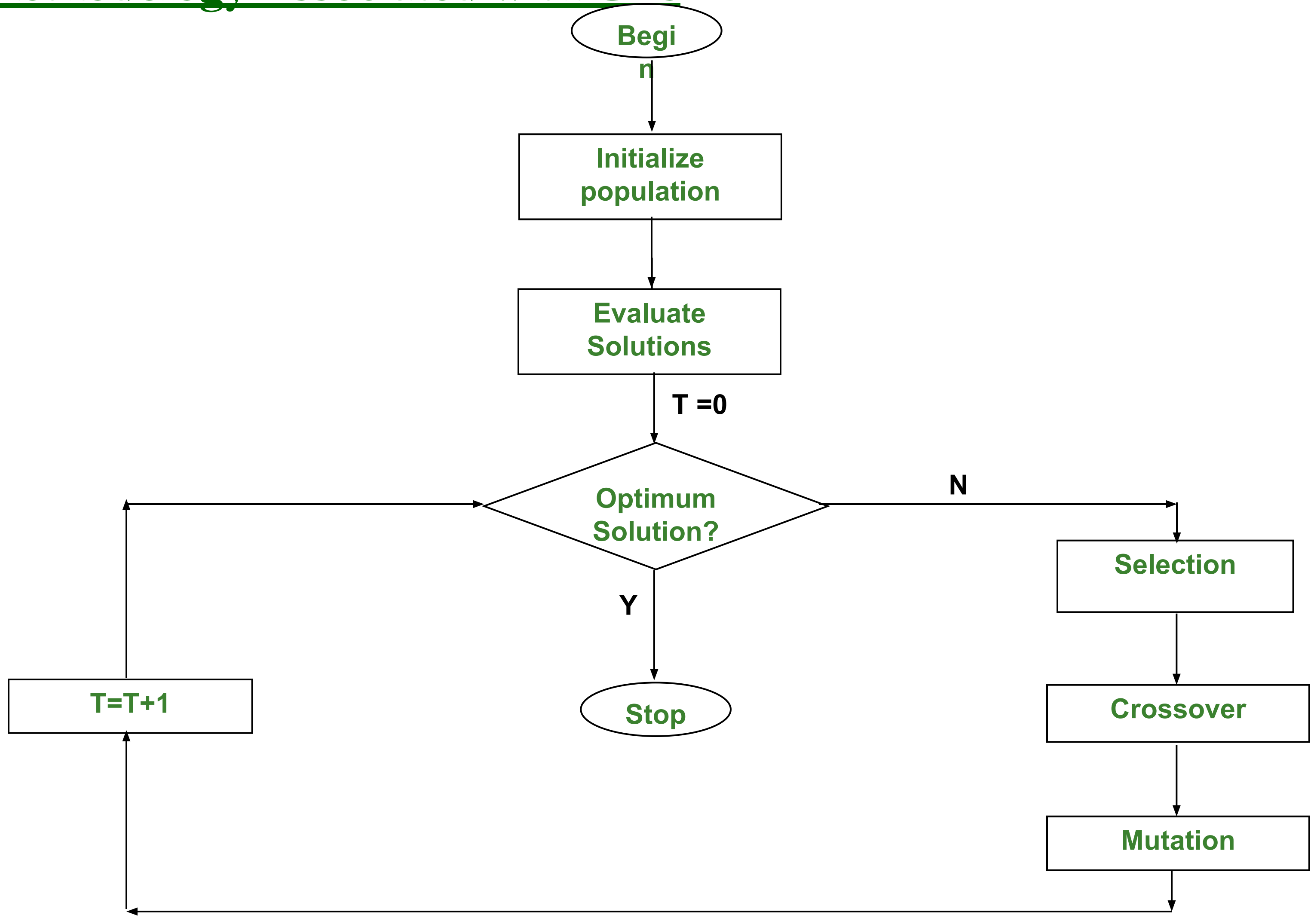
NP Complete problems

Problems in which it is very difficult to find solution, but once we have it, it is easy to check the solution.

Nobody knows if some faster algorithm exists to provide exact answers to NP-problems. An example of alternate method is the genetic algorithm.

Example: Traveling salesman problem.

Methodology Associated with GAs



A Single Loop thru a Number of Evolving Populations

```
Simple_Genetic_Algorithm()
   {
   Initialize the Population;
   Calculate Fitness Function;

   While(Fitness Value != Optimal Value)
      {
      Selection;//Natural Selection, Survival Of Fittest
      Crossover;//Reproduction, Propagate favorable characteristics
      Mutation;//Mutation
      Calculate Fitness Function;
      }
   }
```

Nature Vs Computer - Mapping

Nature	Computer
Population	Set of solutions.
Individual	Solution to a problem.
Fitness	Quality of a solution.
Chromosome	Encoding for a Solution.
Gene	Part of the encoding of a solution.
Reproduction	Crossover

Encoding Using String

Encoding of chromosomes is the first step in solving the problem and it depends entirely on the problem heavily

The process of representing the solution in the form of a string of bits that conveys the necessary information.

Just as in a chromosome, each gene controls a particular characteristic of the individual, similarly, each bit in the string represents a characteristic of the solution.

Encoding Methods

Binary Encoding – Most common method of encoding. Chromosomes are strings of 1s and 0s and each position in the chromosome represents a particular characteristic of the problem.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Encoding Methods (contd.)

Permutation Encoding – Useful in ordering problems such as the Traveling Salesman Problem (TSP). Example. In TSP, every chromosome is a string of numbers, each of which represents a city to be visited.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Encoding Methods (contd.)

Value Encoding – Used in problems where complicated values, such as real numbers, are used and where binary encoding would not suffice.

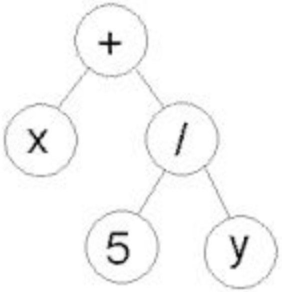
Good for some problems, but *often necessary to develop some specific crossover and mutation techniques for these chromosomes.*

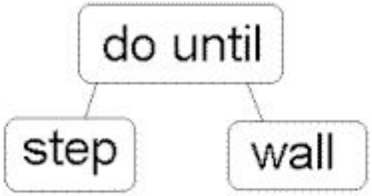
Chromosome A	1.235 5.323 0.454 2.321 2.454
Chromosome B	(left), (back), (left), (right), (forward)

Encoding Methods (contd.)

Tree Encoding – This encoding is used mainly for evolving programs or expressions, i.e. for Genetic programming.

In tree Encoding every chromosome is a tree of some objects, such as functions or commands in a programming language.





In this example, we find a function that would approximate given pairs of values for a given input and output values. Chromosomes are functions represented in a tree.

Operation of Genetic Algorithms

Operation of Genetic Algorithms

- Initialization
- Selection
- Reproduction
- Termination

Initialization

- Initially many individual solutions are randomly generated to form an initial population.

- The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions.

- Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space).

Selection Methods

There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation (epoch). Listed are some of the most commonly used:

Roulette-Wheel Selection, Tournament Selection, Elitist Selection, Rank Selection, Hierarchical Selection

Fitness Function

A fitness function quantifies the optimality of a solution (chromosome) so that that particular solution may be ranked against all the other solutions

- It depicts the closeness of a given '*solution*' to the desired result.
- Watch out for its speed.
- Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions.

Example Of Selection

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count
1	0 1 1 0 1	13	169	0.14	0.58
2	1 1 0 0 0	24	576	0.49	1.97
3	0 1 0 0 0	8	64	0.06	0.22
4	1 0 0 1 1	19	361	0.31	1.23
Sum			1170	1.00	4.00
Average			293	0.25	1.00
Max			576	0.49	1.97

$\text{Prob}_i = f(i) / \sum_i f(i)$

$\text{Expected count} = N * \text{Prob}_i$

Example referred from “Goldberg ’89” -- www.cs.vu.nl/~gusz/ecbook

Roulette Wheel Selection (Fitness-Proportionate Selection)

- In *fitness proportionate selection*, fitness level is used to associate a probability of selection with each individual chromosome.
- In a search space of ‘N’ chromosomes, we spin the roulette wheel N times.
- The “fittest” get through. *(However not all are guaranteed to get through)*

Strings that are fitter are assigned a larger slot and hence have a better chance of appearing in the new population.

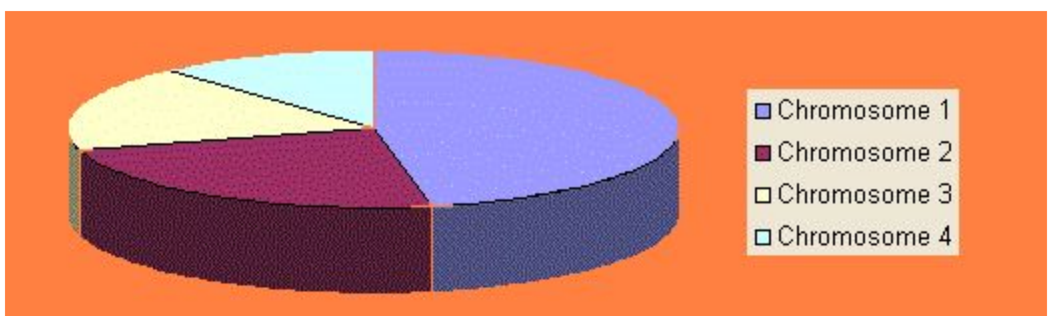


Image referred from ‘Overview of Genetic Algorithms’ -- http://www.tjhsst.edu/~ai/AI2001/GA.HTM

Tournament Selection

Runs a "tournament" among a few individuals chosen at random from the population and selects the winner (the one with the best fitness) for **crossover**

- Two entities are picked out of the pool, their fitness is compared, and the better is permitted to reproduce.
- Selection pressure can be easily adjusted by changing the tournament size.
- Deterministic tournament selection selects the best individual in each tournament.
- Independent of Fitness function.

ADVANTAGE: Decreases computing time, Works on parallel architecture.

Tournament Selection (Pseudo Code)

```
TS_Procedure_nonDeterministic
{
    1.  choose k (the tournament size) individuals from the population at random

    2.  choose the best individual from pool/tournament with probability p

    3.  choose the second best individual with probability p*(1-p)

    4.  choose the third best individual with probability p*((1-p)^2) and so on...
}
```

Reference: wikipedia

Elitism

The best chromosome (or a few best chromosomes) is copied to the population in the next generation.

- Elitism can very rapidly increase performance of GA.
- It is an "*Optimist*" technique.
- A variation is to eliminate an equal number of the worst solutions.

Rank Selection

Rank selection first ranks the population and then every chromosome receives fitness from this ranking.

- Selection is based on this ranking rather than absolute differences in fitness.
- The worst will have fitness 1, second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

ADVANTAGE: Preserves genetic diversity (by preventing dominance of “fitter” chromosomes).

Hierarchical Selection

Individuals go through multiple rounds of selection each generation.

- Lower-level evaluations are faster and less discriminating, while those that survive to higher levels are evaluated more rigorously.

ADVANTAGE: Efficient usage of computing time (By weeding out non-promising candidate chromosomes).

Crossover

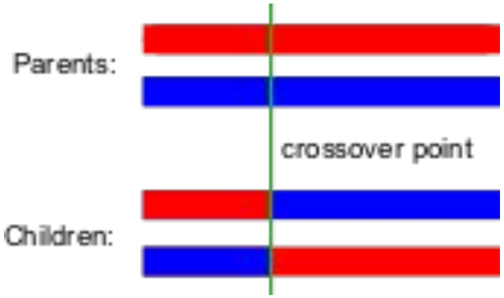
crossover *is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next.*

- *Two strings are picked from the mating pool at random to cross over.*
- *The method chosen depends on the Encoding Method.*

Crossover

- **Single Point Crossover-** A crossover point on the parent organism string is selected. All data beyond that point in the organism string is swapped between the two parent organisms.

- Characterized by **Positional Bias**



Crossover

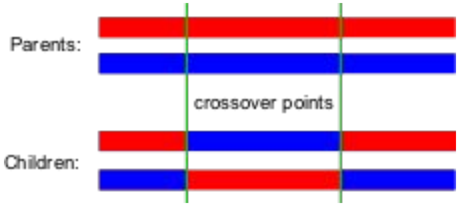
- **Single Point Crossover**

Chromosome1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Reference: Gold berg '89 slides

Crossover

- **Two-Point Crossover-** This is a specific case of a N-point Crossover technique. Two random points are chosen on the individual chromosomes (strings) and the genetic material is exchanged at these points.

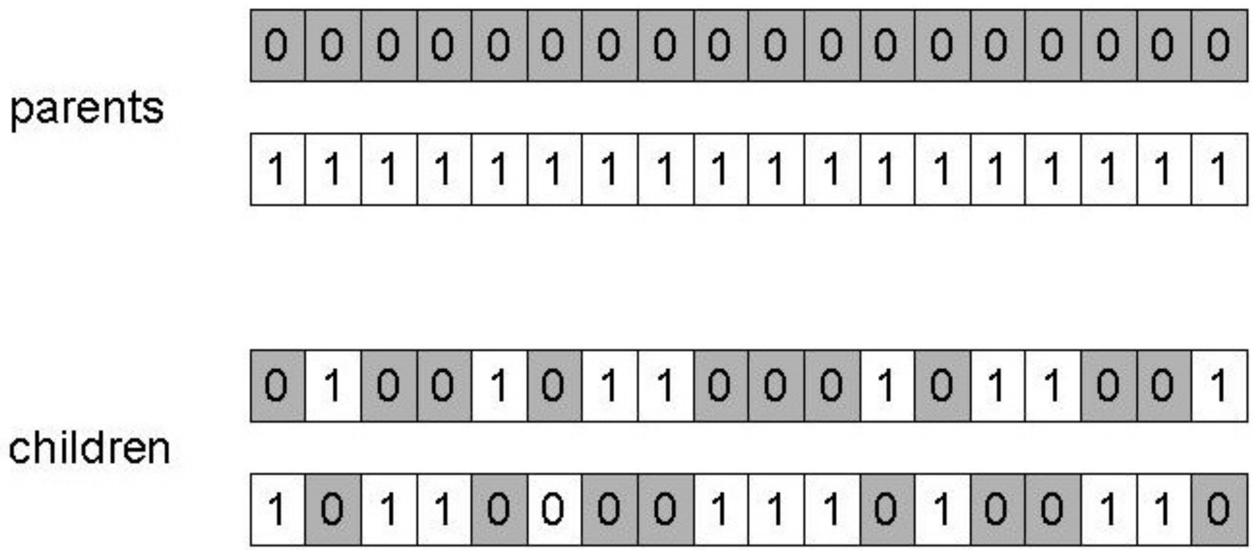


Chromosome1	11011 | 00100 | 110110
Chromosome 2	10101 | 11000 | 011110
Offspring 1	10101 | 00100 | 011110
Offspring 2	11011 | 11000 | 110110

Reference: Gold berg '89 slides

Crossover

- **Uniform Crossover-** Each gene (bit) is selected randomly from one of the corresponding genes of the parent chromosomes.
- Use tossing of a coin as an example technique.



Crossover (contd.)

- Crossover between 2 good solutions **MAY NOT ALWAYS** yield a better or as good a solution.
- Since parents are good, probability of the child being good is high.
- If offspring is not good (poor solution), it will be removed in the next iteration during "Selection".

Mutation

Mutation- *is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.*

Mutation Probability- determines how often the parts of a chromosome will be mutated.

- A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified.

Reference: Gold berg ’89 slides

Example Of Mutation

- **For chromosomes using Binary Encoding, randomly selected bits are inverted.**

Offspring	**11011 00100 110110**
Mutated Offspring	**11010 00100 100110**

Reference: Gold berg '89 slides

Recombination

The process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out.

- The primary objective of the recombination operator is to emphasize the good solutions and eliminate the bad solutions in a population, while keeping the population size constant.
- “Selects The Best, Discards The Rest”.
- “Recombination” is different from “Reproduction”.

Reference: Gold berg ’89 slides

Recombination

- Identify the good solutions in a population.
- Make multiple copies of the good solutions.
- Eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population.

Reference: Gold berg ’89 slides

Crossover Vs Mutation

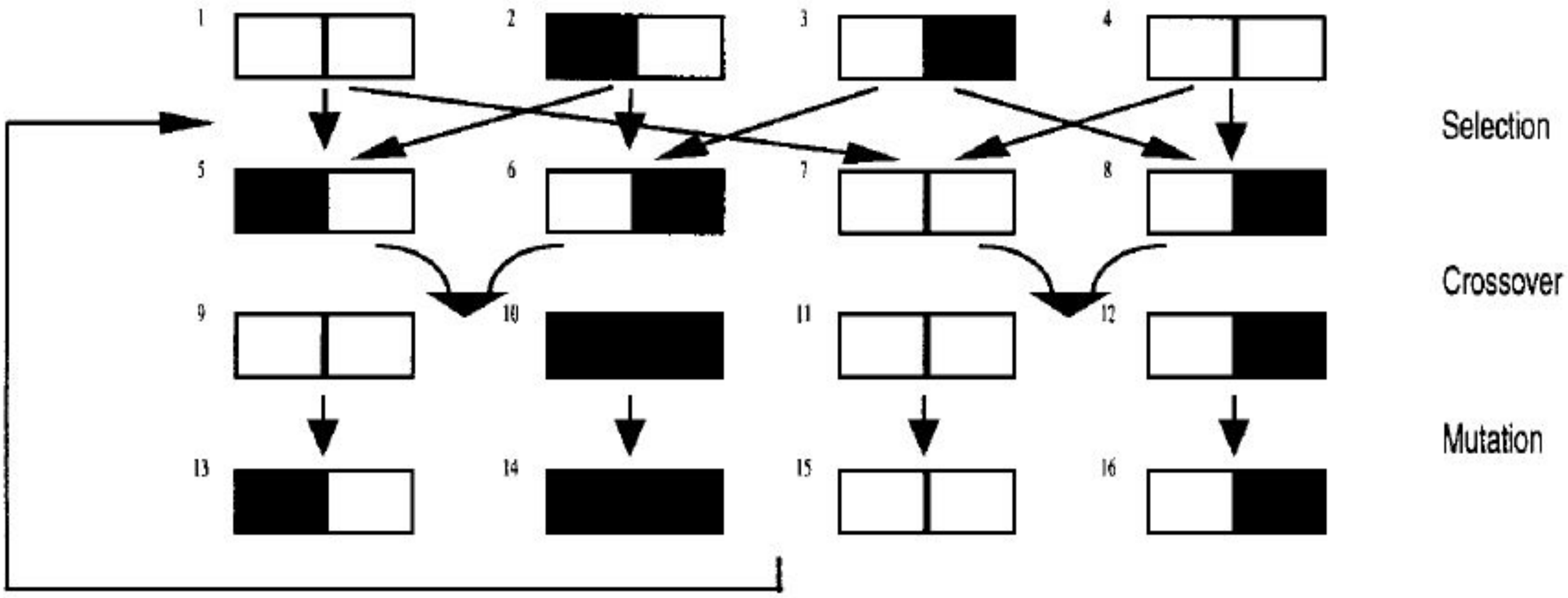
- Exploration: Discovering promising areas in the search space, i.e. gaining information on the problem.
- Exploitation: Optimising within a promising area, i.e. using information.
- There is co-operation AND competition between them.
- Crossover is explorative, it makes a *big* jump to an area somewhere “in between” two (parent) areas.
- Mutation is exploitative, it creates random *small* diversions, thereby staying near (in the area of) the parent.

Simple Genetic Algorithm (Reproduction Cycle)

Select parents for the mating pool

- (size of mating pool = population size)
- Shuffle the mating pool
- For each consecutive pair apply crossover with probability P_c , otherwise copy parents
- For each offspring apply mutation (bit-flip with probability P_m independently for each bit)
- Replace the whole population with the resulting offspring

One generation of a genetic algorithm, consisting of - from top to bottom - selection, crossover, and mutation stages.



“Financial Forecasting using genetic algorithms” - : http://www.ingentaconnect.com

A Genetic-Algorithm-based System

To

Predict Future Performances
of Individual Stocks

Reference:

Technical Document of

LBS Capital Management, Inc., Clearwater, Florida

Link: http://nas.cl.uh.edu/boetticher/ML_DataMining/mahfoud96financial.pdf

The Application

Given a Collection of historical data pertaining to a stock, the task is to predict the future performance of the stock

Specifically, 15 proprietary attributes representing technical as well as fundamental information about each stock is used to predict the relative return of a stock one calendar quarter into the future.
1600 Stocks were considered for running the application.

The Application

Task : Forecast the return of each stock over 12 weeks in future.

Inputs : Historical data about each stock
Historical data here refers to list of 15 attributes.

Attributes : Price to Earning Ratio
Growth rate
Earnings per share

Output : BUY | SELL | NO Prediction

Methodology

A Genetic Algorithm is used for Inductive Machine Learning and then applied to forecast the future performance of the stock

Concepts used by GA in this system:

- Concept Description Structure,
- Michigan approach,
- Niching Method,
- multi-criteria fitness assignments,
- Conflict Resolution etc.

Concept Description Structure

“The Choice of concept description structure is perhaps the strongest bias built into any GA-based inductive learning system”

GAs are capable to optimize any classification structures or set of structures:

- Neural Network weights and topologies
- LISP programs Structures
- Expert System Rules
- Decision Trees etc.

The designed system choose to optimize classification rules:

If GA’s structure consists of two variables representing a particular stock’s price and earning per share, the final rule the GA returns might look like :

IF [Price < 15 and EPS > 1] THEN Buy

Pittsburgh Approach

- Approaches to genetic Classification, named after the Originated University Pittsburgh
- Solutions are represented by individuals that fight each other; those weaker ones die, those stronger ones survive and they can reproduce on the basis of selection, crossover and mutation.

EXAMPLE:

Attribute	**Values**
Head_**S**hape	**R**ound, **S**quare, **O**ctagon
Body_Shape	**R**ound, **S**quare, **O**ctagon
Is_Smiling	**Y**es, **N**o
Holding	**S**word, **B**alloon, **F**lag
Jacket_Color	**R**ed, **Y**ellow, **G**reen, **B**lue
Has_**T**ie	**Y**es, **N**o

Pittsburgh Approach

- To Teach the example
- “the head is round and the jacket is red, or the head is square and it is holding a balloon”
- (<S=R> & <J=R>) V (<S=S> & <H=B>),
- <R***R* V S**B**>
- <100|111|11|111|1000|11 V 010|111|11|010|1111|11>
- 1 – don’t care condition

Michigan Approach

- Another Approach to genetic Classification, named after the Originated University Michigan
- Each individual consists of a *condition* (a conjunction of several blocks) and of a *conclusion*
- Example :

  “it can walk, it can jump but it cannot fly AND it barks AND it is 90 cm long □ it is a dog".

Pittsburgh Vs Michigan Approach

- Michigan approach encodes a single rule
  1. Smaller Memory requirement and faster processing time
  2. Mechanisms must be designed to maintain a cooperating and diverse set of rules within the population, to handle credit assignment and to perform conflict resolution
- Pittsburgh approach encodes each element an entire concept
  1. The best population element at the end of GA's run is the final concept used for classification
  2. Simplified credit assignment and easier conflict resolution
  3. Drawback : redundancy and increased processing time

The designed system choose to adopt Michigan approach

- Encoding each element as in Pittsburgh approach places a large handicap on a GA-based learner

- The problems presented by this system can be handled quite well by the Michigan approach of Genetic Algorithm.

Niching Method

- When GA are used for optimization, the goal is typically to return a single value, the best solution found to date

- The entire population ultimately converges to the neighborhood of a single solution

- GA's that employ niching methods are capable of finding and maintaining multiple rules using a single population by a GA

The designed system maintains multiple rules:

- Having Chosen Michigan approach, the system assures that the population maintains a diverse and cooperating set of rules by incorporating niching method

Credit Assignment as Fitness Function

General Principles of the fitness assignments:

- Award higher fitnesses to more accurate and general classification rules
- When doing Boolean or exact concept learning, penalize heavily for covering incorrect examples

The designed system choose to combine all criteria into a single fitness function

Conflict Resolution

When the rules covering a particular example indicates two or more classifications, a conflict occurs

Ways to resolve Conflict:

- One scheme is not to resolve conflicts
  (This is acceptable in many domains in which an action is not required for every example the system encounters)
- A second possible conflict resolution scheme is to make a random choice between classifications indicated by the overlapping rules
- A third is to choose the most common of the conflicting classifications by sampling the training data

The designed system choose to maintain a default hierarchy method :

- The most specific matching rule wins
- To promote evolution of rules to handle special cases

Forecasting Individual Stock Performance

- Using historical data of a stock, predict relative return for a quarter

Example: If IBM stock is up 5% after one quarter and the S&P 500 index is up 3% over the same period, then IBM's relative return is +2%

- An *example* consists of 15 attributes of a stock at specific points in time and the relative return for the stock over the subsequent 12 week time period.
- 200 to 600 examples were utilised depending on the experiment and the data available for a particular stock
- Combination of rules is required to model relationships among financial variables

Example: Rule-1 : IF [P/E > 30] THEN Sell

Rule-2: IF [P/E < 40 and Growth Rate > 40%] THEN Buy

Preliminary Experiments

- For a Preliminary set of experiments, to predict the return, relative to the market, a Madcap stock randomly selected from the S&P 400.
- 331 examples present in the database of examples of stock X
- 70% of examples were used as a training set for the GA
- 20% of the examples were used as a stopping set, to decide which population is bet
- 10% of the examples were used to measure performance
- A sample rule that the GA generated in one of the experiment:

IF [Earning Surprise Expectation > 10% and Volatility > 7% and …]

THEN Prediction = Up

- Same set of experiments were used using Neural Network with one layer of hidden nodes using backpropagation algorithm with same training, stopping and test sets as that of GA experiment

Observations on the Results

- The GA correctly predicts the direction of stock relative to the market 47.6% of the time and incorrectly predicts the 6.6% of time and produces no prediction 45%
- Over half of the time (47.6% + 6.6%), the GA makes a prediction. When it does make a prediction, GA is correct 87.8% of the time
- The Neural Network correctly predicts the direction relative to the market 79.2% of the time and incorrectly predicts direction 15.8% of the time. When it does make a prediction, the NN is correct 83.4%

Comparison with Neural Networks

- Advantage of GA's over NN's:

  GA's ability to output comprehensible rules

  (1) To provide rough explanation of the concepts learned by black-box approaches such as NN's

  (2) To learn rules that are subsequently used in a formal expert system

- GA makes no prediction when data is uncertain as opposed to Neural Network.

Another most widely used application in Financial Sector

- To Predict exchange rates of foreign currencies.

  Input: 1000 previous values of foreign currencies like

  USD Dollar, Indian Rupee, Franc, Pound is provided.

  Output: Predicts the currency value 2 weeks ahead.

  Accuracy Percentage obtained: 92.99%

A Genetic Algorithm Based Approach to Data Mining

Ian W Flockharta
Quadstone Ltd Chester Street Edinburgh EH RA UK

Nicholas J Radclie
Department of Mathematics and Statistics University of Edinburgh

Presented at "AAAI: Knowledge Discovery and Data Mining 1996", Portland, Oregon

Objective

Design a mechanism to perform directed data mining, undirected data mining and hypothesis refinement based on genetic algorithms

Types of data mining

1. **Undirected data mining**

- System is relatively unconstrained and hence has the maximum freedom to identify pattern
- eg: “Tell me something interesting about my data”

2. **Directed data mining**

- System is constrained and hence becomes a directed approach
- eg: “Characterise my high spending customers”

3. **Hypothesis testing and refinement**

- System first evaluates the hypothesis and if found to be false tries to refine it
- eg: “I think that there is a positive correlation between sales of peaches and sales of cream: am I right”

Pattern Representation

- Represented as subset descriptions.
- Subset descriptions are clauses used to select subsets of databases and form the main inheritable unit
- Subsets consist of disjunction or conjunction of attribute value or attribute range constraints

Subset Description: {*Clause*} {or *Clause*}
Clause: {*Term*} and {*Term*}
Term: *Attribute* in *Value Set*
| *Attribute* in *Range*

Patterns

Rule pattern:

if C then P

C and P represent the condition and prediction respectively of a rule

Distribution shift pattern:

The distribution of A when C and P

The distribution of A when C

A is the hypothesis variable, C and P are subset descriptions

Correlation pattern:

when C the variables A and B are correlated

A and B are hypothesis variables and C is a subset description.

Pattern Templates and Evaluation

- Templates are used to constrain the system
- Constrained based on the number of attributes, number of conjunctions or disjunctions and also based on mandatory attributes
- Components of templates can be initialized or fixed
- Initialized parts occur in all newly created patterns
- Fixed parts cannot be changed by mutation or crossover and other genetic operators
- Undirected mining is done with a minimal template and directed mining is done by restricting the pattern
- Several pattern evaluation techniques based on statistical methods are used to identify the relevance of the pattern

The Genetic Algorithm

- Mutations and crossover are performed at the different levels
- Can be done at the subset description, clause or term level
- Both uniform and single point crossover are done at the clause level
- Single point crossover is done at the term level
- Mutation is done at different levels with specified probabilities or threshold
- Clauses, terms and values can be added or deleted
- Reproductive partners are selected from the same neighborhood to improve diversity and also to identify several patterns in a single run
- The population is updated using a heuristic like replacing the lowest fit

Explicit Rule Pattern

if Proportion of households with 1 child ≥ 0.12
(Approximate percentiles 36% - 100%)
(true: 1618 false: 939 unique false: 272)

and

Number of Ford Dealers > 0
(Approximate percentiles 62% - 100%)
(true: 936 false: 1621 unique false: 809)

and

Proportion of households with 3+ cars **in** 0.01 .. 0.07
(Approximate percentiles 4% - 86%)
(true: 2038 false: 519 unique false: 108)

then

Ford market share segment F (Sierra) **in** 0.06 .. 0.75
(Approximate percentiles 30% - 100%)
(true: 1770 false: 787 unique false: 787)

Left hand side matches 19% of the database
Right hand side matches 69% of the database

	Expected	Actual
Accuracy:	69%	93%
Coverage:	20%	27%

Distribution Shift Pattern

The distribution of "Ford market share of segment D (Escort)"

when Proportion of households — Council ≥ 0.20
(Approximate percentiles 62% - 10%)
(true: 929 false: 1628 unique false: 123)

and

Proportion of households with 2 children ≥ 0.11
(Approximate percentiles 26% - 100%)
(true: 1854 false: 703 unique false: 233)

and

Proportion of unemployed in population ≥ 0.04
(Approximate percentiles 52% - 100%)
(true: 1183 false: 1374 unique false: 48)

and

Proportion of households with 0 cars ≥ 0.31
(Approximate percentiles 60% - 100%)
(true: 1015 false: 1542 unique false: 89)

has median 0.28 and is significantly shifted
from the overall distribution which has median value 0.20.

Other Areas of Application of GA

- Genetic Algorithms were used to locate earthquake hypocenters based on seismological data

- GAs were used to solve the problem of finding optimal routing paths in telecommunications networks. It is solved as a multi-objective problem, balancing conflicting objectives such as maximising data throughput, minimising transmission delay and data loss, finding low-cost paths, and distributing the load evenly among routers or switches in the network

- GAs were used to schedule examinations among university students. The Time table problem is known to be NP-complete, meaning that no method is known to find a guaranteed-optimal solution in a reasonable amount of time.

- Texas Instruments used a genetic algorithm to optimise the layout of components on a computer chip, placing structures so as to minimise the overall area and create the smallest chip possible. GA came up with a design that took 18% less space

Advantages Of GAs

- **Global Search Methods:** GAs search for the function optimum starting from a *population of points* of the function domain, not a single one. This characteristic suggests that GAs are global search methods. They can, in fact, climb many peaks in parallel, reducing the probability of finding local minima, which is one of the drawbacks of traditional optimization methods.

- **Blind Search Methods:** GAs only use the information about the *objective function*. They do not require knowledge of the first derivative or any other auxiliary information, allowing a number of problems to be solved without the need to formulate restrictive assumptions. For this reason, GAs are often called blind search methods.

Advantages of GAs (contd.)

- **GAs use probabilistic transition rules** during iterations, unlike the traditional methods that use fixed transition rules.

  This makes them more robust and applicable to a large range of problems.

- **GAs can be easily used in *parallel machines*-** Since in real-world design optimization problems, most computational time is spent in evaluating a solution, with multiple processors all solutions in a population can be evaluated in a distributed manner. This reduces the overall computational time substantially.

Questions ?