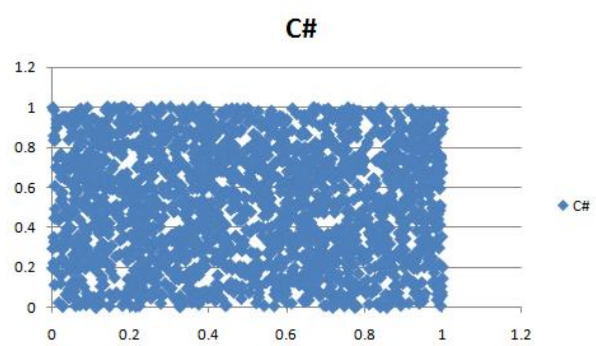
Computational Intelligence

Unit # 7

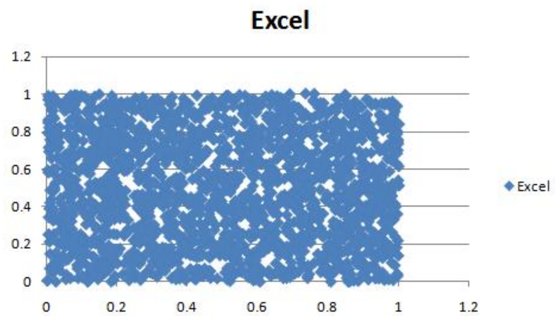
C# Random Numbers

	Random 1 Series	Random 2 Series
Q1	0.253525	0.267425
Median	0.5056	0.5123
Q3	0.749675	0.751325



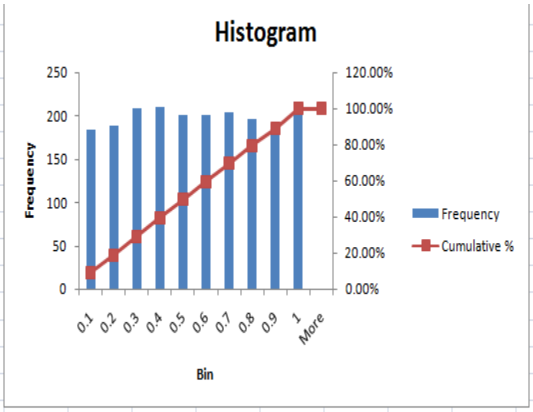
Excel Random Numbers

	Random 1 Series	Random 2 Series
Q1	0.234332	0.24797
Median	0.495124	0.486396
Q3	0.748964	0.743278



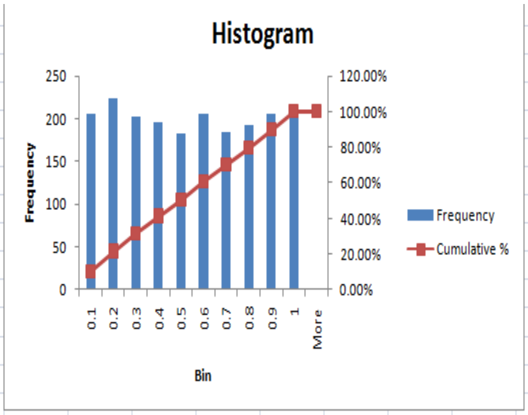
C# Random Numbers Histogram

Bin	Frequency	Cumulative %
0.1	184	9.20%
0.2	188	18.60%
0.3	208	29.00%
0.4	210	39.50%
0.5	201	49.55%
0.6	201	59.60%
0.7	204	69.80%
0.8	196	79.60%
0.9	192	89.20%
1	216	100.00%
More	0	100.00%



Excel Random Numbers Histogram

Bin	Frequency	Cumulative %
0.1	205	10.25%
0.2	224	21.45%
0.3	202	31.55%
0.4	195	41.30%
0.5	182	50.40%
0.6	205	60.65%
0.7	185	69.90%
0.8	192	79.50%
0.9	206	89.80%
1	204	100.00%
More	0	100.00%



C# Program

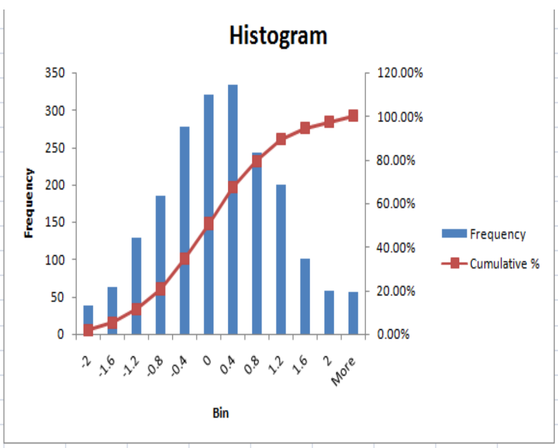
```
•    using System;
•    using System.Collections.Generic;
•    using System.Linq;
•    using System.Text;
•    using System.IO;

•    namespace ConsoleApplication1
•    {
•        class Program
•        {
•            static void Main(string[] args)
•            {
•                StreamWriter strOut = new StreamWriter("output.txt");

•                Random r1 = new Random(1111);
•                 Random r2 = new Random(1729);

•                for (int i = 0; i < 2000; i++)
•                    strOut.WriteLine(r1.Next(10000)/10000.0 + " , " + r2.Next(10000) / 10000.0);

•                strOut.Flush();
•                strOut.Close();
•            }
•        }
•    }
```

Normal (Gaussian) Random Numbers

- The most important transformation functions is known as the **Box-Muller** (1958) transformation.
- It allows us to transform uniformly distributed random variables, to a new set of random variables with a Gaussian (or Normal) distribution.
- The most basic form of the transformation looks like:
  - y1 = sqrt(- 2 ln(x1)) cos(2 pi x2)
  - y2 = sqrt(- 2 ln(x1)) sin(2 pi x2)
- We start with *two* independent random numbers, x1 and x2, which come from a uniform distribution (in the range from 0 to 1).
- Then apply the above transformations to get two new independent random numbers which have a Gaussian distribution with zero mean and a standard deviation of one.

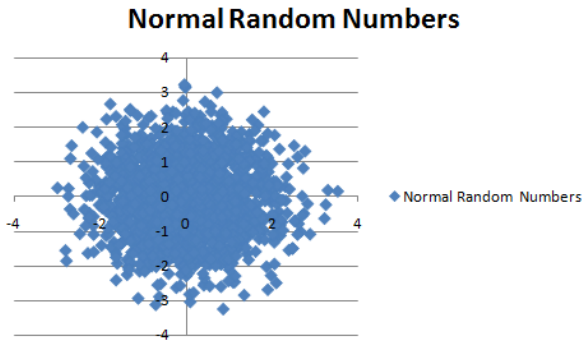
Normal Distribution

Bin	*Frequency*	*Cumulative %*
-2	38	1.90%
-1.6	63	5.05%
-1.2	128	11.45%
-0.8	185	20.70%
-0.4	277	34.55%
0	321	50.60%
0.4	333	67.25%
0.8	242	79.35%
1.2	199	89.30%
1.6	101	94.35%
2	57	97.20%
More	56	100.00%

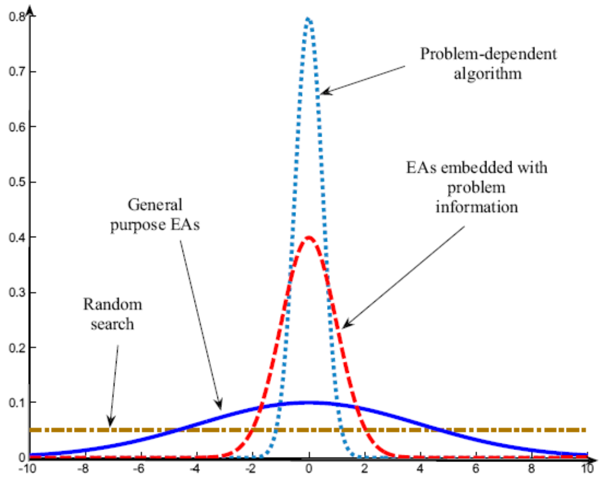


Normal Random Numbers

	Random 1 Series	Random 2 Series
Q1	-0.66701	-0.69655
Median	-0.00948	0.022777
Q3	0.624935	0.69396



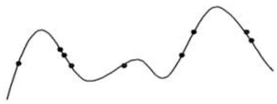
One Possible Interpretation of NFL Theorem



Typical behaviour of an EA

Phases in optimising on a 1-dimensional fitness landscape

Early phase:
quasi-random population distribution



Mid-phase:
population arranged around/on hills

Late phase:
population concentrated on high hills

Probability Distributions for Mutation

- All EAs follows a stochastic search process.
- Stochasticity is introduced by computing step sizes as a function of noise, η_{ij} *, sampled from some* probability distribution. The most popular distributions are
  - Uniform
  - Gaussian

Uniform

- Noise is sampled from a uniform distribution

$$\eta_{ij}(t) \sim U(x_{min}, x_{max})$$

- where x_{min} *and* x_{max} *provide lower and upper bounds for the values of* η_{ij}.

Gaussian

- For the Gaussian mutation operators, noise is sampled from a zero-mean, normal distribution.
- For completeness sake, and comparison with other distributions, the Gaussian density function is given as (assuming a zero mean)

$$f_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)}$$

- where σ is the deviation of the distribution

A Canonical EA (Source: DeJong)

- This canonical EA
  - maintains a fixed-size population of individuals,
  - each of which is a fixed-length vector (chromosome) of real-valued parameters (genes), and
  - whose objective fitness is determined by calling a "landscape" evaluation function.
- After randomly generating and evaluating the fitness of the members of the initial population, EA
  - effects population evolution by repeatedly selecting a member of the current population at random (uniformly) as a parent and
  - produces a clone offspring.

A Canonical EA (Cont'd)

- EA then introduces some variation in the offspring via a simple mutation operator which, on average picks one gene in the genome to modify by randomly adding/subtracting a small value to the current value of the selected gene.
- The new offspring is forced to immediately compete for survival against an existing member of the population selected at random.
- If the objective fitness of the child is greater than the selected member, the child survives and the old member dies off. Otherwise, the child dies without ever residing in the population.

Steady-State Model of EA

- This form of population management in which offspring are produced one at a time and immediately compete for survival is called an incremental or "steady state" model.
- An alternate model is one in which a large number (a batch) of offspring are produced from a fixed pool of parents. Only after the entire batch is produced there is competition for survival into the next generation. Such models are often referred to as "batch" or "generational" algorithms.

Limitations

- It is quite possible that some members of the current population never produce any offspring in spite of the fact that they may have high fitness.
- Similarly, by randomly picking members from the current population to compete with children for survival into the next generation, it is quite possible that weak individuals may survive by the luck of the draw and not out of merit.

History of Evolutionary Algorithms

- Several efforts were started in parallel during 1960s
  - Evolution Strategies (Berlin Technical University)
  - Genetic Algorithms (University of Michigan)
  - Evolutionary Programming (UCLA)
- During 1990s the above communities agreed to the term "Evolutionary Computation".

Evolution Strategies

- At the Technical University Berlin, Rechenberg and Schwefel (1965 paper) began formulating ideas about how evolutionary processes could be used to solve difficult real-valued parameter optimization problems.
- From these early ideas emerged a family of algorithms called "*evolution strategies*" which today represent some of the most powerful evolutionary algorithms for function optimization.

Evolutionary Programming

- At UCLA during the same period Fogel (1966 paper) saw the potential of achieving the goals of artificial intelligence via evolutionary techniques.
- These ideas were initially explored in a context in which intelligent agents were represented as finite state machines, and an evolutionary framework called "*evolutionary programming*" was developed which was quite effective in evolving better finite state machines (agents) over time.

Blondie24

- **Blondie24** is an artificial intelligence checkers-playing computer program.
- The screen name was used on the The Zone, an internet boardgaming site, during 1999. During this time, Blondie24 played against some 165 human opponents and was shown to achieve a rating of 2048, or better than 99.61% of the playing population of that web site.
- The design of Blondie24 is based on a minimax algorithm of the checkers game tree in which the evaluation function is an artificial neural network.
- The neural net receives as input a vector representation of the checkerboard positions and returns a single value which is passed on to the minimax algorithm.

Blondie24 (Cont'd)

- The neural net weightings were obtained by an evolutionary algorithm, in this case by having a population of Blondie24-like programs play against each other and later eliminating those with fewest earned points in which the players earned +1 for a win, 0 for a draw, and -2 for a loss, and repeating the process with a new population derived from the winners. The result was an evolutionary process selecting the programs that played better checkers games.

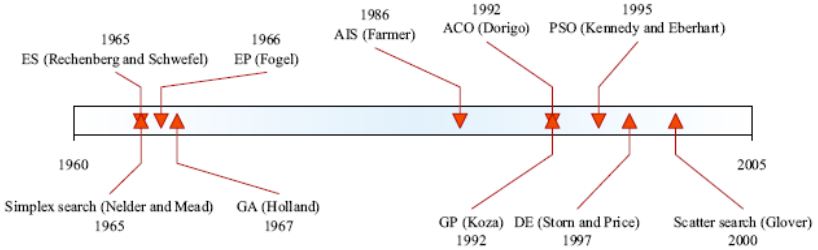
Genetic Algorithms

- At the University of Michigan, Holland (1962 paper) saw evolutionary processes as a key element in the design and implementation of robust adaptive systems, capable of dealing with an uncertain and changing environment.
- His view emphasized the need for systems which self-adapt over time as a function of feedback obtained from interacting with the environment in which they operate.
- This lead to an initial family of "reproductive plans" which formed the basis for what we call "*simple genetic algorithms*" today.

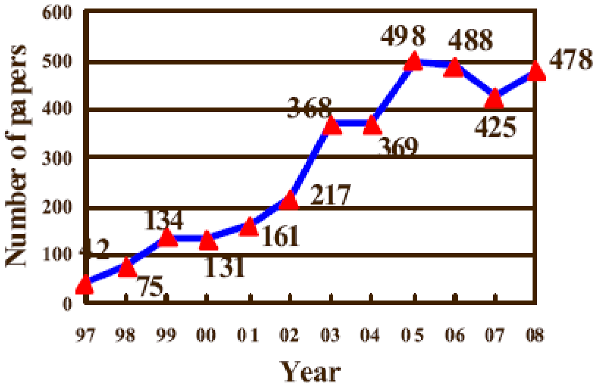
The Unifying 90s

- Till 90s, much of the R&D was done independently and in parallel without much interaction among the various groups.
- The emergence of the various EA conference in the late 1980s and early 1990s, however, changed all that as representative from various EA groups met, presented their particular viewpoints, challenged other approaches, and were challenged in return.
- The immediate result was an agreement on the term "evolutionary computation" as the name of the field and a commitment to start the field's first journal, Evolutionary Computation.

Recap: History of Evolutionary Computation



Number of EC Papers Published in ISI Indexed Journals



Sajjad Haider Spring 2012 27