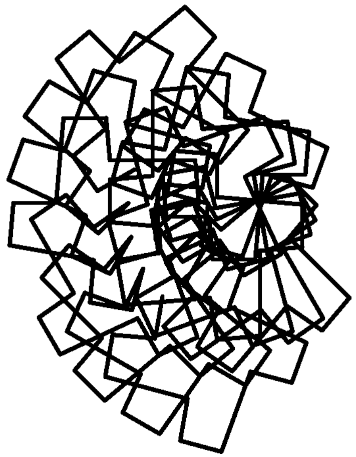
Natural Computing:an introduction and some financial applications

Anthony Brabazon

Wirtschafts Universität, Wien
11 November 2011



Financial Mathematics and Computation Research Cluster

www.fmc-cluster.org

UCD Natural Computing Research & Applications Group (NCRA)
http://ncra.ucd.ie

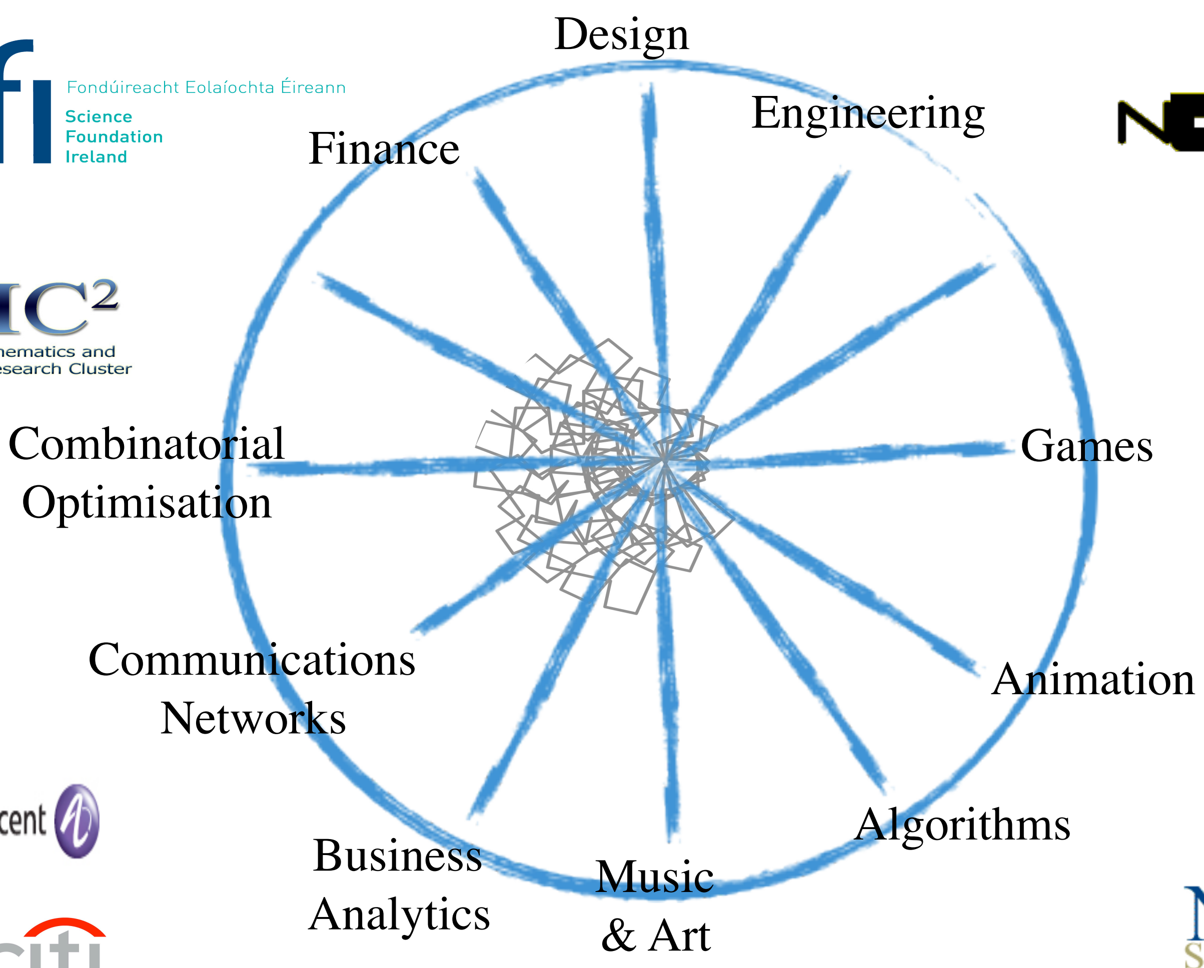
Overview

- Introduction
- Overview of Natural Computing
- A Tour of Some Financial Application Areas …

Research Group

- Natural Computing Research and Applications Group (http://ncra.ucd.ie)
  - Interdisciplinary research group based in UCD's CASL (http://casl.ucd.ie)
- Development of NC algorithms / theory
- Application areas
  - Finance, bioinformatics, architecture, engineering, sound synthesis, game AI etc. etc.
- Staffing
  - Five faculty
  - 15 PhDs / Post docs

Research Group - NCRA



UCD Natural Computing Research & Applications Group (NCRA)

Research Group

FMC²
Financial Mathematics and
Computation Research Cluster

Core Research Team
6 Lead academics (3 universities)
15 academic collaborators (from 5 other universities)
4 Post Doctoral Researchers
15 PhD Researchers
5 Industry Partners

Initial Research Activities

- Robust Asset Allocation
- Fund Performance Evaluation
- Crashes and Portfolio Choice
- Information Theory and Financial Markets - Model Selection and Complexity
- Grammatical Evolution for Asset Allocation – equity and fixed income
- Algorithmic Trading
- Asset Pricing and Risk
- Risk Management of Real Estate
- Pension Risk
- Time-series Dynamics of Multivariate Return Distributions
- Semi-parametric Estimation of Portfolio Risk
- Copulas, Fractals and Chaos

sfi
science foundation ireland
fondúireacht eolaíochta éireann

Biologically-inspired Algorithms

- Biological organisms earn a living in ‘difficult’ environments
  - Typically “high-dimensional” and dynamic

- Mechanisms have arisen which assist the ‘survivability’ / adaptability of populations of biological creatures in these environments

- These are potentially useful in helping inspire us when designing algorithms to attack interesting real-world problems in the finance (and other) domain(s)

Dynamic Environments

- Biological responses to dynamic environments include …
  - Populations rather than individuals
    - Multiple individuals generates diversity
    - Multiple ‘probes’ (learning trials) of the environment
    - Lower risk of extinction if you have diversity
  - Multiple mechanisms for learning
    - Personal lifetime learning capability
    - Social learning (communication between individuals)
    - Genetic learning
  - Mechanisms for maintaining a memory of good past solutions
  - Mechanisms for generating diverse new individuals (solutions)
  - Fitness-based (de)selection
  - Focus is on survival not optimisation (i.e. robustness)

Earning a Living …

- Let’s take a simple example of the problem of ‘earning a living’ in a dynamic environment where the future actions of other agents are unknown …

Earning a Living …

- How can we find good 'rules' for surviving in this (or any other…) environment?

- At each time step in the program one of nine rules (in decreasing priority) is fired

#Rule 1:

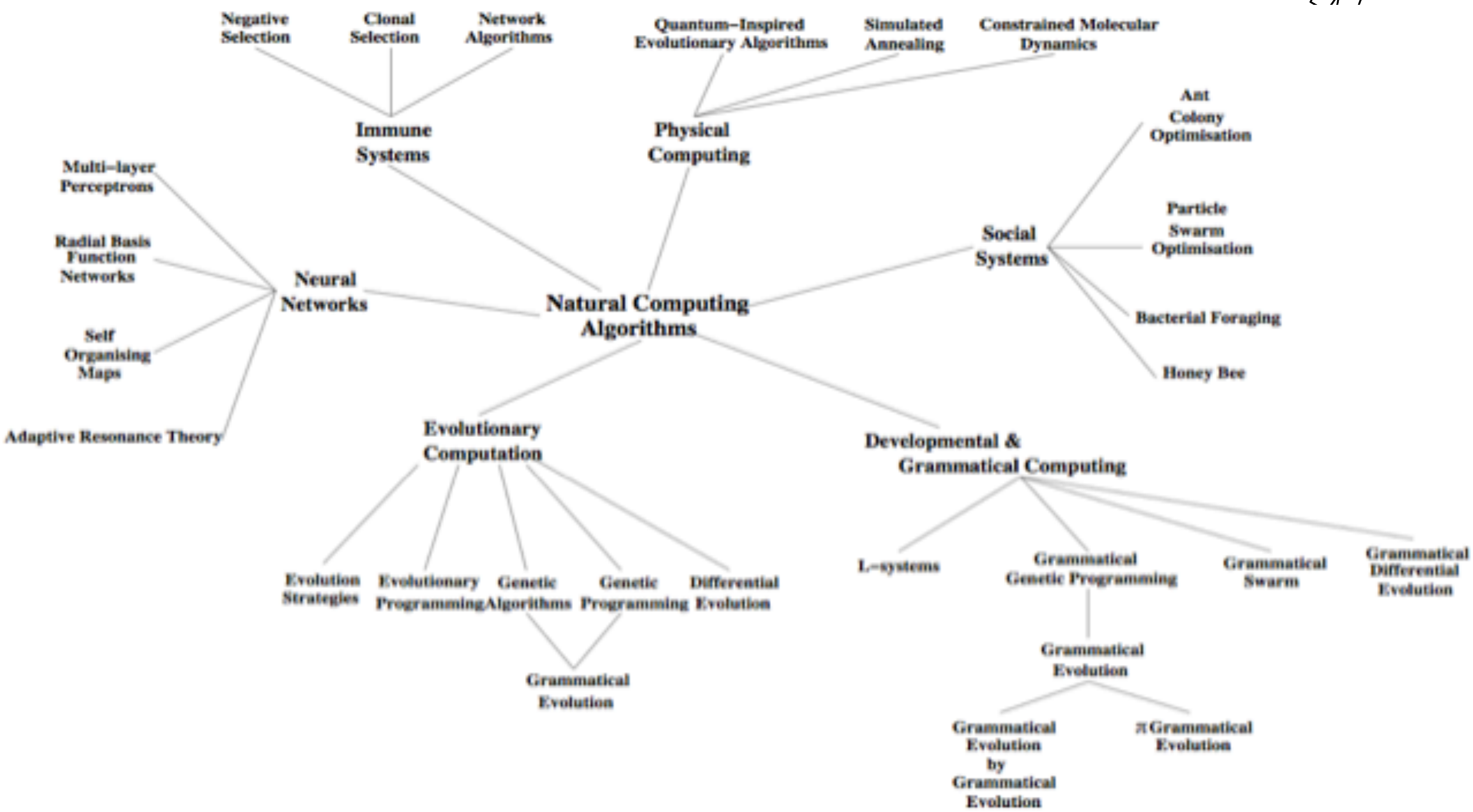
IF (distance(nearest_power_pill) ≦ 5(3*)) AND (4 ≦ distance(nearest_ghost) ≦ 8) AND (distance(ghost_nearest_to_the_nearest_power_pill) ≧ 6(4*)),

THEN stop moving and ambush (enter the ambush state) at the corner or cross point near the nearest power pill waiting for a ghost to come closer, where distance (nearest_power_pill) is the distance from Ms. Pac-Man to the nearest power pill, distance(nearest_ghost) the distance from Ms. Pac-Man to the nearest ghost, and distance(ghost_nearest_to_the_nearest_power_pill) the distance from Ms. Pac-Man to the ghost nearest to the power pill nearest to Ms. Pac-Man, and the numbers with * in the parentheses are those for the second stage of the game.

Natural Computing

- Working definition for this seminar …
  - *"computational algorithms whose design is inspired by systems and phenomena that occur in the natural world"*

Natural Computing



Brabazon & O'Neill (2011) *Natural Computing Algorithms*, Springer.

Financial Application Areas of NC

i. Optimisation

ii. Model induction

iii. Agent-based modelling

Survey Articles

Brabazon, A., O'Neill, M. and Dempsey, I. (2008). An Introduction to Evolutionary Computation in Finance, IEEE Computational Intelligence Magazine, 3(4):42-55.

Brabazon, A., Dang, J., Dempsey, I. and O'Neill, M. (2011). Natural Computing in Finance: A Review, in Handbook of Natural Computing: Theory, Experiments and Applications
G. Rozenberg, T. Baeck and J. Kok (eds),), Berlin: Springer (forthcoming in November 2011).

Evolutionary Cycle

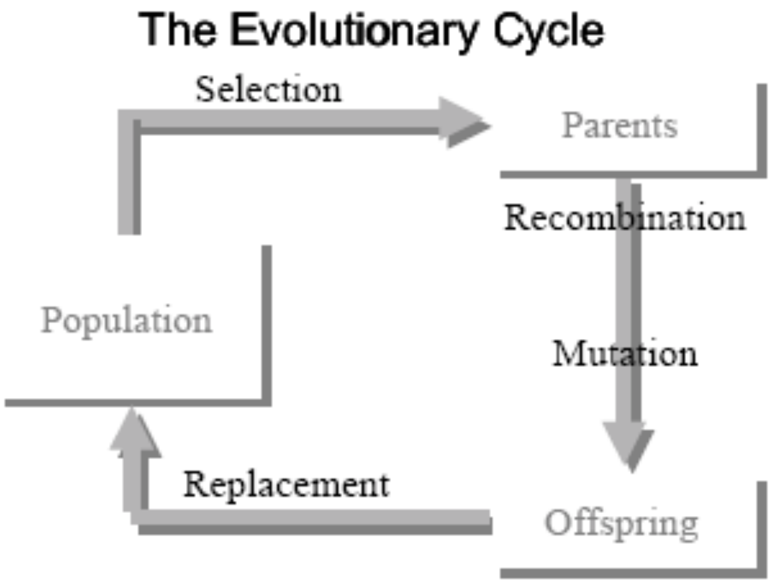


Figure 5.1: *The evolutionary cycle.*

Distinction between *genotype* and *phenotype*

Evolution as a *search process* in genotypic space but the 'worth' of the genotypes is only assessed at phenotypic level

$$x(t+1) = v(s(x(t)))$$

Evolutionary Cycle

- We can apply these general ideas for:
  - Optimisation
    - uncover good coefficients for a (perhaps non-linear) model
  - Model Induction
    - determine a model structure which provides a good 'fit' to a dataset we have?
- The first problem is one of optimisation, the second is one of structural design
- We will concentrate on the first (and simpler!) problem initially …

Evolutionary Computation

Canonical evolutionary algorithm

Initialise population

WHILE (Termination condition False)

    Calculate fitness of each individual

    Select parents

    Create offspring

    Update population

ENDWHILE

Designing an EA to solve a Problem .....

- Need to define:

1. A *genotype to phenotype mapping*

2. Method for *initialising* the population of genotypes
   - may be random if you have no prior … of course, if you have knowledge of 'good regions' of the search space, this can be used to 'seed' the initial population

3. Method for *generating diversity* in your population of genotypes

4. Definition of, and method for *evaluating,* the fitness of an individual in the population

A Simple Optimisation Example

- Sometimes designing the genotype to phenotype mapping is simple
  - For example, suppose we want to design a genotype to encode three coefficients for a linear regression model of the form …

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

  - The genotype could be a real-valued string, encoding the three model coefficients

-3.1245	5.6219	11.3411

A Simple Optimisation Example

	β_0	β_1	β_2		Fitness (MSE)
1	-3.1245	5.6219	11.3411		0.3245
2	-4.5612	-0.2317	6.1311		0.7436
3	2.3412	1.6432	2.7811		0.6718
…	…	…	…		…
n	-3.6245	4.8219	13.3411		0.3015

A Simple Optimisation Example

	β_0	β_1	β_2		Fitness (MSE)
1	-3.1245	5.6219	11.3411		0.3245
2	-4.5612	-0.2317	6.1311		0.7436
3	2.3412	1.6432	2.7811		0.6718
…	…	…	…		…
n	-3.6245	4.8219	13.3411		0.3015

Suppose these two 'good' parents are chosen

A Simple Optimisation Example

	β_0	β_1	β_2		Fitness (MSE)
Child 1	-3.3745	5.2219	12.3411		…

We generate a ‘child’ solution by applying a pseudo-crossover operation to the two parents

Crossover uses information from both parents (recombines their good information – here using a simple averaging process)

A Simple Optimisation Example

	β_0	β_1	β_2		Fitness (MSE)
Child 1	-3.3745	5.2219	12.6500		0.2918 (say)

Next, apply a mutation operator to the child 'solution' and determine its fitness

Mutation allows for the discovery of information not contained in either parent

A Simple Optimisation Example

	β_0	β_1	β_2		Fitness (MSE)
Child 1	-3.3745	5.2219	12.6500		0.2918 (say)
…	…	…	…		…
Child n	…	…	…		…

- Process is repeated until ‘n’ children are created
- These ‘n’ children form the next ‘generation’ of the population, and the algorithm continues
- Iteratively over time, the quality of members of the population improve and converge on the optimal values of β_0, β_1, β_2

Crossover

- Crossover
  - Aims to use information from better parents ….
  - A very simple implementation is intermediate crossover’ (an average)

Parent 1	-3.1245	5.6219	11.3411
Parent 2	-3.6245	4.8219	13.3411
Child	-3.3745	5.2219	12.3411

  - More generally, could use $P_1+\alpha(P_2-P_1)$, where P_1 and P_2 are the real-values in that locus of each parent and α is a scaling factor (perhaps randomly drawn from [-2, +2])
    - Defines a hypercube based on the current location of the parents

Mutation

- Mutation
  - Allows the uncovering of new information that was not present in either parent
  - A simple mutation mechanism for real-valued genotypes could be the addition of a random draw from $N(0, \alpha_i)$ to each element of each child solution
    - Hence, most mutations are small with occasional larger mutation steps
    - Value of α_i is user-defined (scaled as appropriate)

Model Calibration

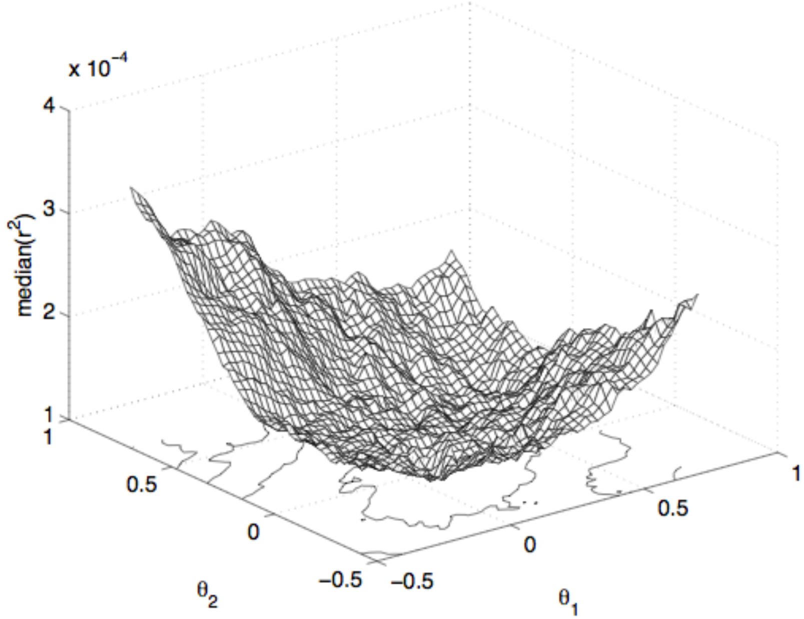
- Often in finance, we may have a theory / model which explains an item of interest such as returns / volatility etc.

- Model may be complex, non-linear

- In order to apply the model, its parameters need to be determined or ‘calibrated’
  - Select model parameter values so that ‘model output’ matches actual market output
  - Calibrated model can then be used to (for example) price financial instruments

Model Calibration

- Heuristic methods such as the GA can be useful particularly,
  - In problems which traditional optimisers find difficult (multiple local optima etc.)
  - In facilitating robust estimation of parameters (e.g. extreme values can make MSE-based parameters unstable) – removing the 'closed form solution' requirement



Exemplar mapping from coefficients into median squared residual illustrating multiple local optima (LMS estimation) (Gilli and Schumann, 2009)

Model Calibration

Black Scholes Option Pricing Model

$$C_{BS}(S_t, K, r, q, \tau; \sigma) = S_t e^{-q\tau} N(d_1) - K e^{-r\tau} N(d_1)$$

$$d_1 = \frac{-\ln m + \left(r - q + \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}} \qquad d_2 = d_1 - \sigma\sqrt{\tau}$$

Calibration (implied volatility)

$$\sigma_{BS}(S_t, K) > 0$$
$$C_{BS}(S_t, K, r, \tau; \sigma_{BS}(S_t, K)) = C_M(S_t, K)$$

Hochreiter, R. and Wozabal, D. (2010). Evolutionary Estimation of a Coupled Markov Chain Credit Risk Model, in Natural Computation in Computational Finance, pp. 31-44, Berlin: Springer.

Dang, J., Brabazon, A., O'Neill, M. and Edelman, D. (2008). Estimation of an EGARCH Volatility Option Pricing Model using a Bacteria Foraging Optimisation Algorithm, in Natural Computation in Computational Finance, Brabazon, A. and O'Neill, M. (eds), pp. 109-131, Berlin: Springer.

Fan, K., O'Sullivan, C., Brabazon, A., O'Neill, M. and McGarraghy, S. (2008). Calibration of the VGSSD Option Pricing Model using a Quantum Inspired Evolutionary Algorithm, in Quantum-Inspired Evolutionary Computation, Nedjah, N., Coelho, L. and Mourelle, L. (eds), pp. 133-153, Berlin: Springer.

Parameterising a Simple Rule

- Other problems of interest include the parameterisation of asset selection / trading rules

- How might you represent a basic technical trading rule of the following form as a string?

IF x day MA of price ≥ y day MA of price
THEN Go Long *ELSE* Go Short

5	10	1

(*5 day MA* ≥ 10 day MA THEN Go Long)

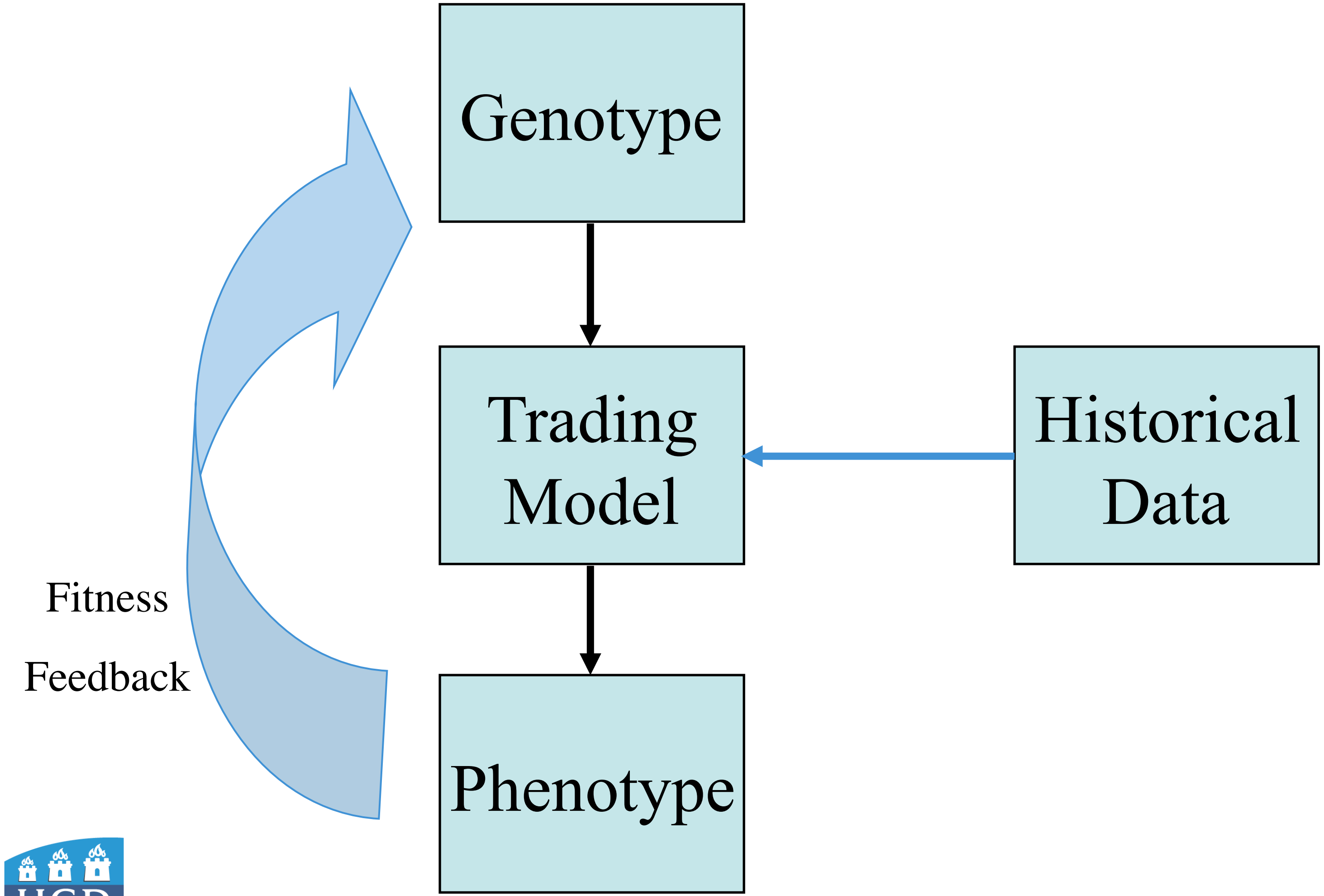
Structure / Model

+

Parameters

Proto-trading rule

Back-testing the Quality of a Genotype ...



A Slightly More Complex Example …

IF [Indicator$_i$(t) (<,>) value$_j$] *THEN* (Buy, Sell, Do nothing)

Indicator$_i$	t	<,>	value$_j$	Buy, Sell, Do nothing

- The above are <u>simple</u> illustrative examples, much more complex, compound, trading rules, which would defy any attempt at discovery via enumerative methods, could also be generated using AND, NOT, OR etc. operators

Uncovering an Stock Selection Rule

High sales growth relative to industry average?	High debt level relative to industry average?	High level of cash flow from operations relative to industry average?	High level of liquidity relative to industry average?	High profit level relative to industry average?

In a simple case, we may be trying to uncover a good subset from an array of plausible filter rules (possible rules depend on your investment style)

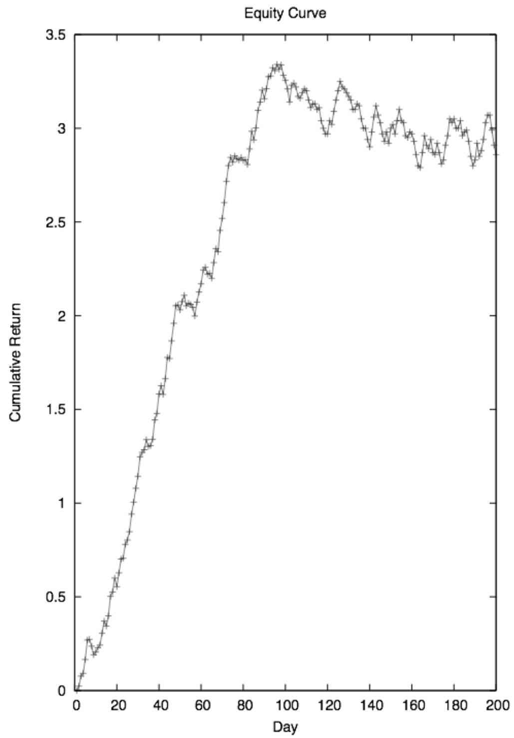
Each indicator could be coded as a 0 (no) or 1 (yes), with an evolutionary process being applied to uncover the best subset of filter rule components

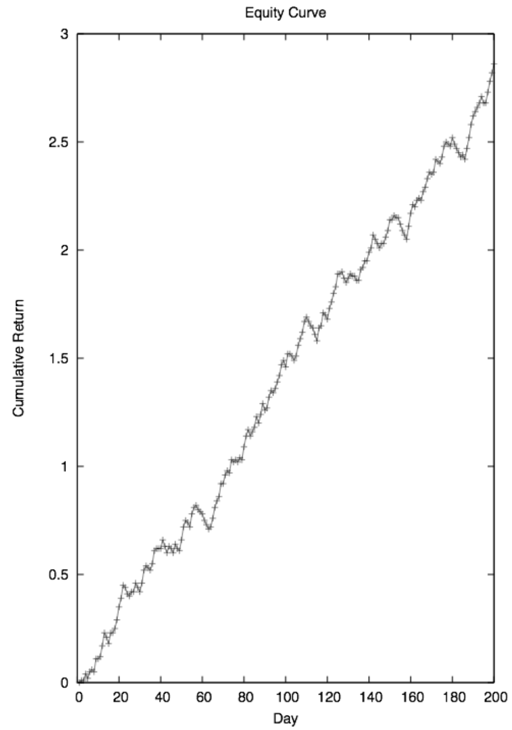
Filter 1	…			Filter n	R/σ (say)
0	…	0	1	1	XX

Of course, you could also apply an evolutionary process to breed the individual elements of the filter rules and their thresholds … but this is better done using an evolutionary model induction methodology

Generalisation / Defining an Appropriate Fitness Function

A statistical measure of 'goodness of fit' is not the same thing as (risk adjusted) 'profitability'!

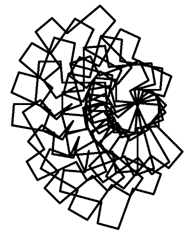




Bradley, R., Brabazon, A. and O'Neill, M. (2010). Objective Function Design in a Grammatical Evolutionary Trading System, Proceedings of the 2010 IEEE Congress On Evolutionary Computation, IEEE Press.

Agapitos, A., O'Neill, M. and Brabazon, A. (2010). Evolutionary Learning of Technical Trading Rules without Data-mining Bias, Proceedings of 11th International Conference on Parallel Problem Solving from Nature (PPSN 2010), Lecture Notes in Computer Science, Springer-Verlag, Berlin.

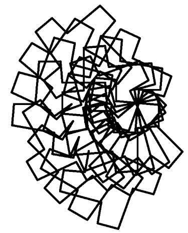
Model Induction



- Can we ‘recover’ a suitable model (+ associated parameters) from a set of data?
  - Likely to be useful when we have data but weak theory (perhaps some idea of the likely relevant variables but less idea how they link together)
  - MLPs (universal approximators … but readability?)
  - Today we will discuss the powerful methodologies of Genetic Programming (and associated grammar-based variants such as Grammatical Evolution)

- Applications include:
  - financial time-series forecasting, credit risk modelling, pricing model discovery, forecasting takeover targets, prediction of earnings, IPO underpricing, trading system design etc. etc.

Genetic Programming

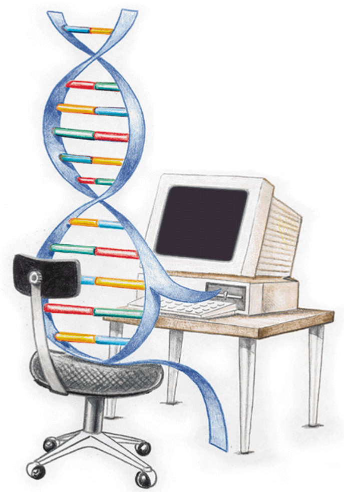


- An evolutionary automated programing methodology

- Popularised by John Koza

  in his 1992 book '*Genetic Programming: on the programming of computers by means of natural selection*'

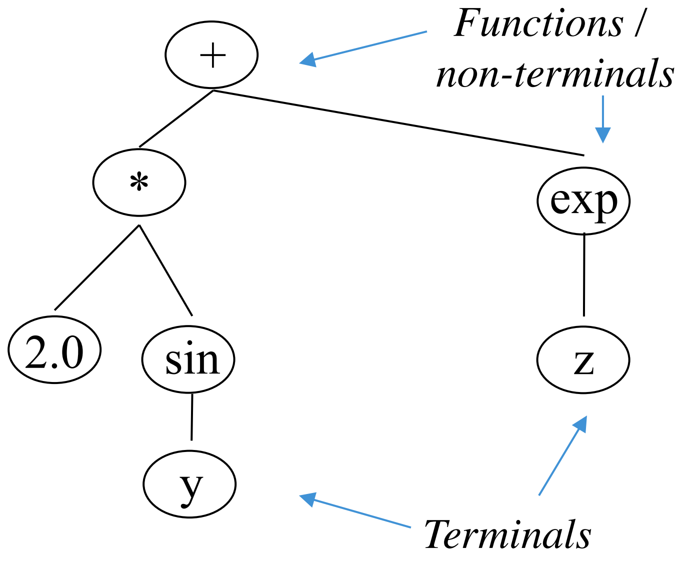
- GP adopts an evolutionary metaphor
  - Generate a population of trial solutions, assess worth of each, select, crossover, mutate, replace



Genetic Programming

- Individual is or represents a program

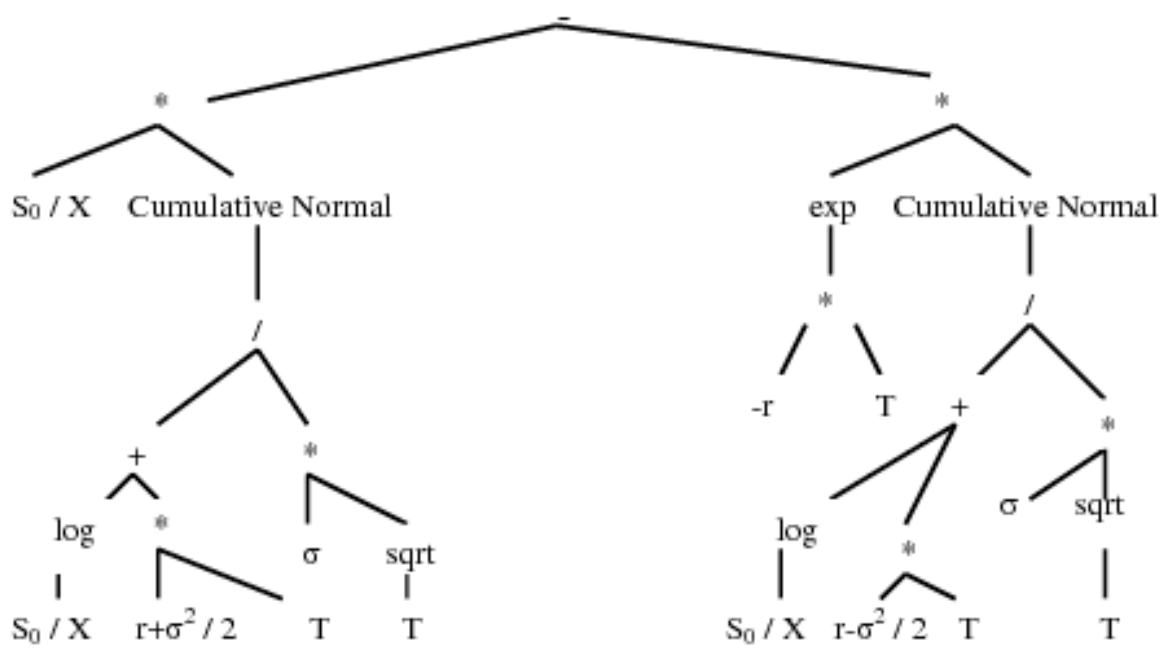
```
#include<stdio.h>
#include<stdlib.h>
…
int main(int argc, char* argv){
      float x=0.0, y=0.0, z=0.0;
      x=atof(argv[1]); y=atof(argv[1]); z=atof(argv[1]);
      z1 = 2.0*sin(y) + exp(z);
      printf("The answer is: %f\n",z1);
      return (0);
}
```



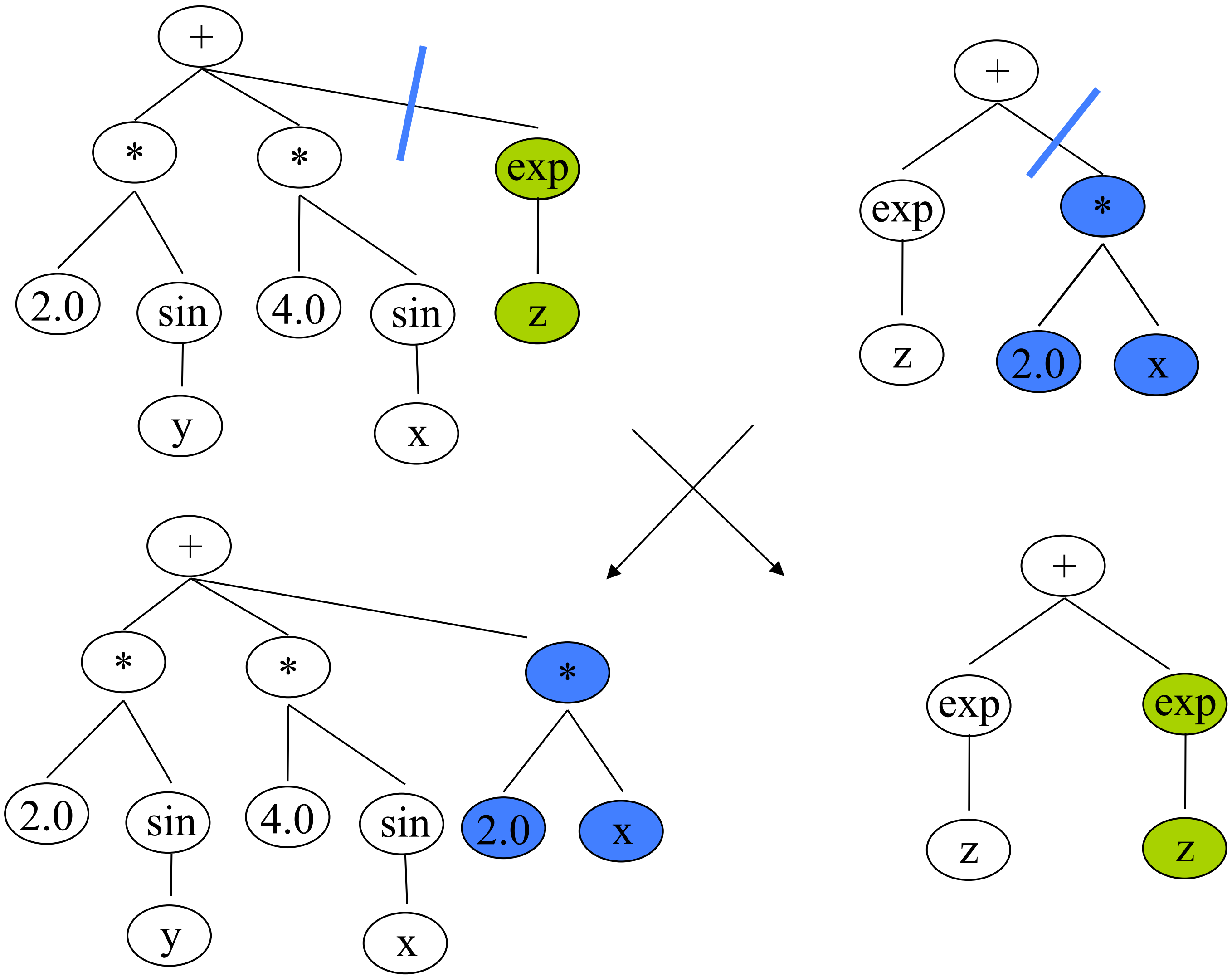
Of course, a 'program' (or the equivalent tree representation) is just a list of rules … and many financial problems can be viewed as a search for a 'good' list of rules …. lending decisions, investment decisions, ….

Genetic Programming

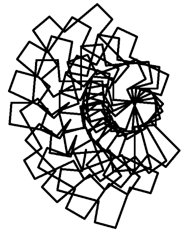
'Stylised' tree representation of the BS option pricing model



Diversity Generation



Genetic Programming



- Typically, in financial applications of GP, the goal is to recover / discover the data-generating model
  - What model can we reverse engineer from the data?

- As each ‘model’ is evolved, its quality / fitness can be assessed by determining how well it explains the observed (training) data

GP Pseudo-code

Define terminals, primitive functions and fitness function

Set parameters for GP run (population size, probabilities for mutation, crossover etc., selection / replacement strategy etc.)

Initialise start population of solutions (grow, full, ramped-half and half)

Calculate fitness of each solution (run each program!)

WHILE (Termination condition False)

  Select parents

  Create offspring

    using mutation, crossover, cloning, architecture-altering...

  Update population

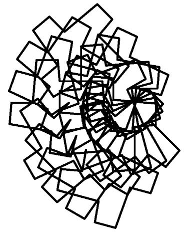
  Calculate fitness of each solution

ENDWHILE

Grammar-based GP

- Canonical GP system can be modified by incorporating a ‘grammar’
- Helps allow the user to ‘guide’ the search process to desired structural form
- This is often important in real-world financial applications
  - Human-readability for mission-critical applications
  - Requirements for risk committee sign-off
  - Allows easy incorporation of domain expertise
- Next we will introduce one form of GBGP – *Grammatical Evolution*

Grammatical Evolution

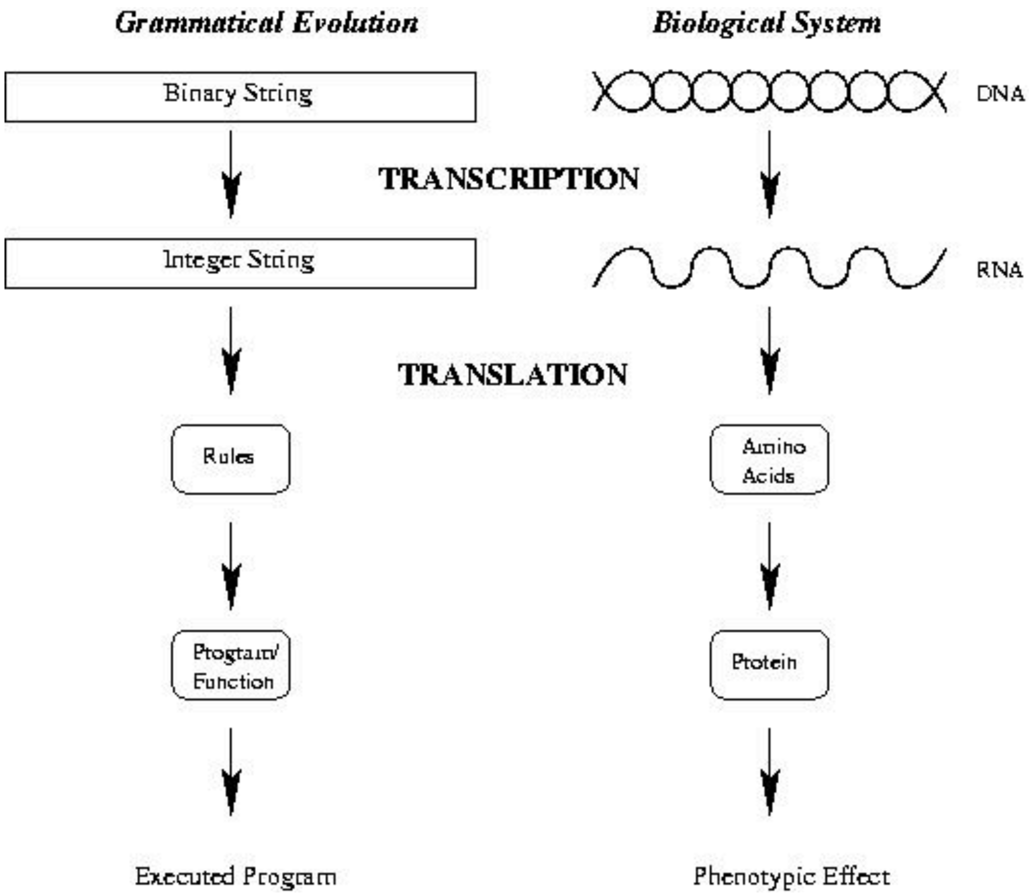


- Like GP, model specification / choice of explanatory variables not assumed ex-ante

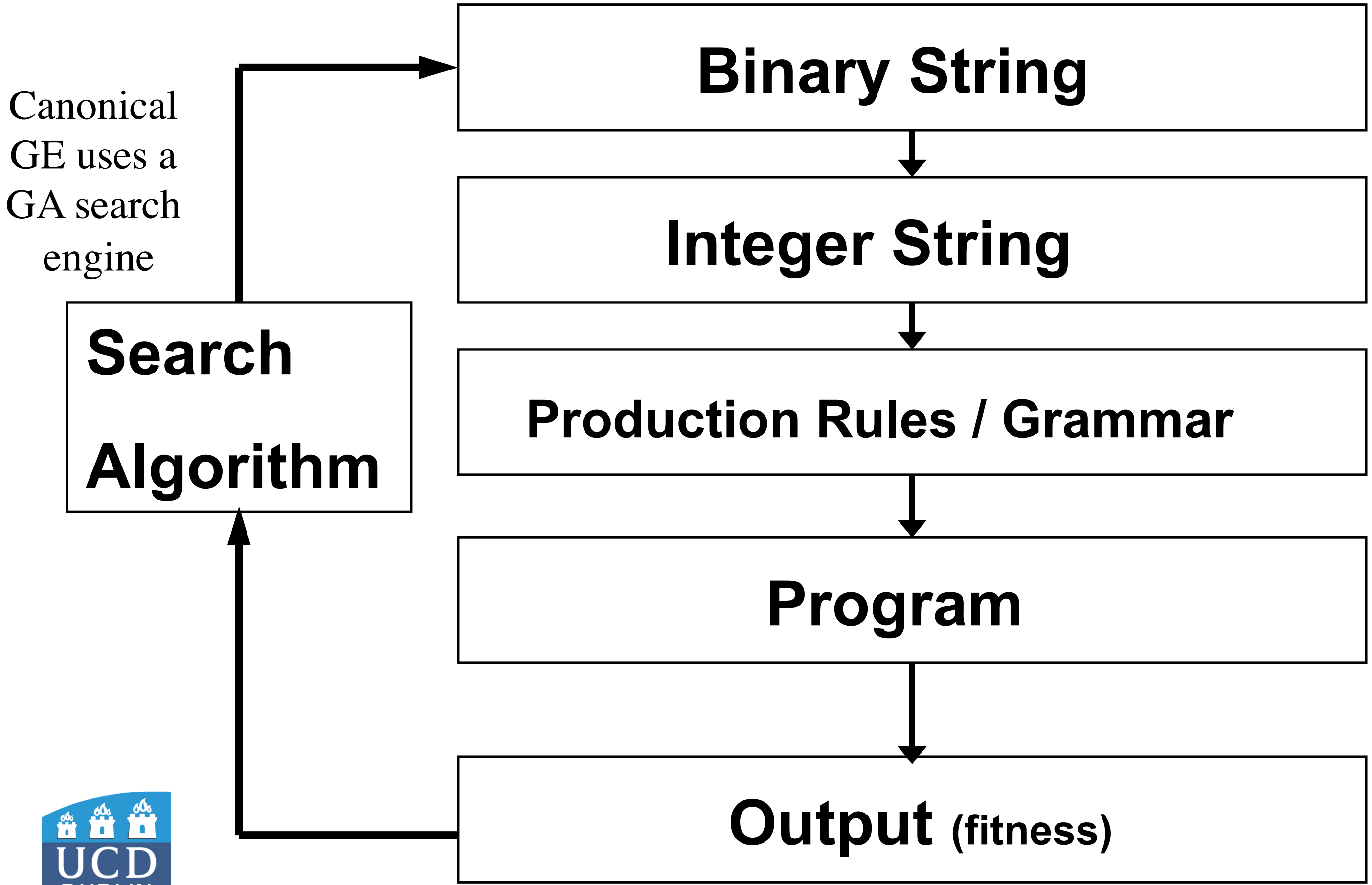
- Inspired by biological process of gene expression
  - Distinction between search and solution space

- Unlike GP, there is a clear genotype → phenotype mapping
  - Each program is generated from a variable length binary (or integer) string
  - Key item is that the evolutionary process is effectively applied to the 'production rules'
    - i.e. the developmental rules governing the production of the phenotype, and not directly to the phenotype itself

Genotype-Phenotype Mapping



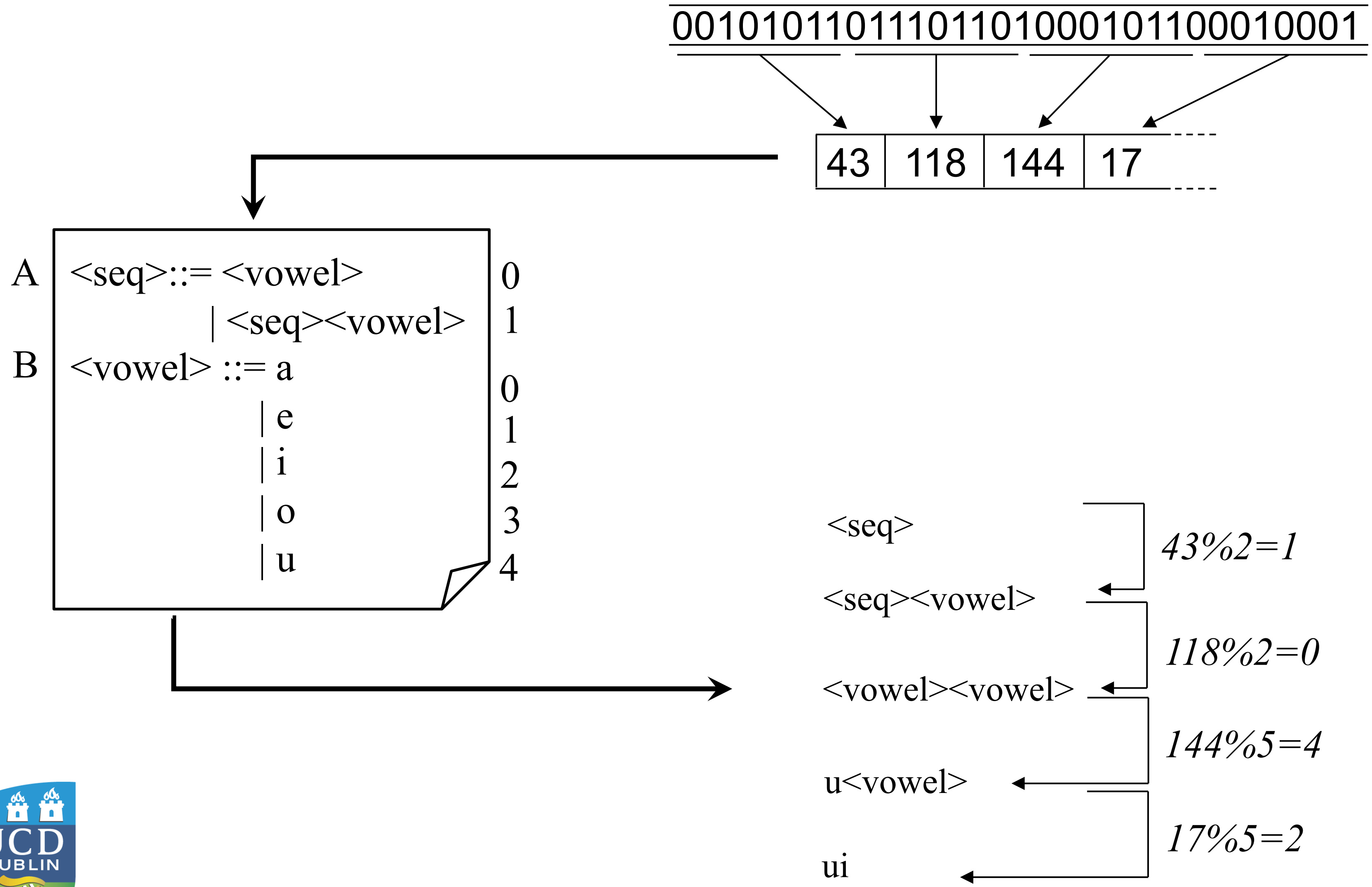
Grammatical Evolution



BNF Grammar

- BNF Grammar a 4-Tuple <*T*, *N*, *P*, *S*>
  - T: Terminal set (*items which can appear in the final program*)
  - N: Non-terminal set (elements of the production ‘grammar’)
  - P: Set of production rules
  - S: Start symbol (an element of N)

Sampe GE Grammar and Mapping



Financial Applications …

- Recent Financial Applications of evolutionary model induction technologies include:
  - Asset selection
  - Options price modelling
  - Credit risk assessment
  - Algorithmic trading
  - Money-laundering detection …..

Credit Risk Modelling

Brabazon, A. and O'Neill, M. (2006). Credit Classification Using Grammatical Evolution. Informatica, 30(3):325-335.

Brabazon, A. and O'Neill, M. (2008). Bond Rating with piGrammatical Evolution, in Studies in Computational Intelligence: Knowledge-Driven Computing, Knowledge Engineering and Intelligent Computation, Cotta, C., Reich, S., Ligęza, A. and Schafer, R. (eds), pp. 17-30, Berlin: Springer Verlag.

Corporate Failure Prediction

Brabazon, A. and Keenan, P. (2004). A hybrid genetic model for the prediction of corporate failure, Computational Management Science, 1 (3-4):293-310.

Brabazon, A. and O'Neill. M. (2004). Diagnosing Corporate Stability using Grammatical Evolution, International Journal of Applied Mathematics and Computer Science, 14(3):363-374.

Trading System Generation

Brabazon, A. and O'Neill, M. (2004). Evolving technical trading rules for foreign-exchange markets, Computational Management Science, 1 (3-4):311-327.

Brabazon, A., Meagher, K., Carty, E. , O'Neill, M. and Keenan, P. (2005). Grammar-mediated time-series prediction, Journal of Intelligent Systems, 14(2-3):123-143.

Illustrative Example – Bond Rating

- Companies can be financed by share capital or debt capital

- Debt capital → Bank debt
  → Bond issue

- If bonds are to be publicly traded, they require (in US) a credit rating from a 'recognised' rating agency (S&P, Moody's, Fitches') …
  S&P cover 99.2% of traded debt in the US

- Credit rating represents an agency's opinion of the creditworthiness of a borrower

Illustrative Example – Bond Rating

Initial Rating	Default (%)
AAA	0.52
AA	1.31
A	2.32
BBB	6.64
BB	19.52
B	35.76
CCC	54.38

Investment Grade

Source:
S&P 1987-2002

Illustrative Example – Bond Rating

Grammar

<lc> ::= if(<expr> <relop> <expr>) class="Junk";
else class="Investment Grade";

<expr> ::= (<expr>) + (<expr>) | <coeff> * <var>

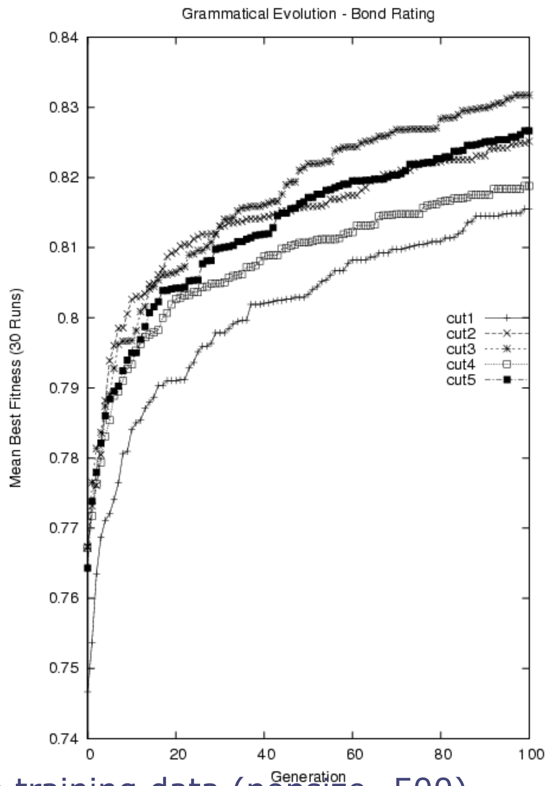
<var> ::= var1[index] |…] | var8[index]

<coeff> ::= (<coeff>) <op> (<coeff>) | <float>

<op> ::= + | - | *

<float> ::= 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | -1 | .1

<relop> ::= <=



Data
600 US firms (420 training:180 out of sample)

S&P bond ratings + data drawn from their financial statements

Best fitness values, over 30 runs and five recuts, on the training data (popsize=500)

Coefficients concord with perceived financial wisdom …

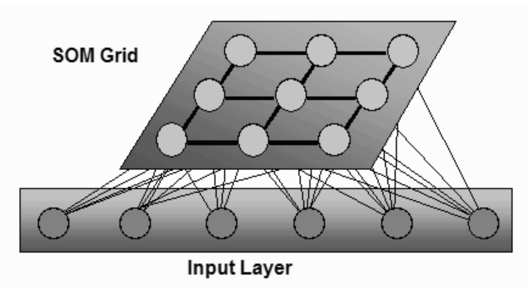
Low/negative retained earnings; low / negative total assets; high levels of debt, are symptomatic of firms with a junk rating

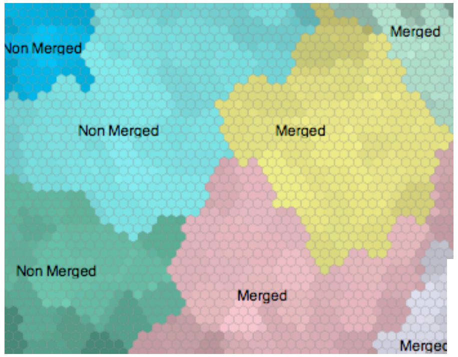
Model Induction

Other Applications / Other Methods

Hickey, R, Little, E. and Brabazon, A. (2006). Identifying Merger and Takeover Targets Using a Self-Organising Map, Proceedings of the 2006 International Conference on Artificial Intelligence (ICAI '06), edited by Arabnia, H. et al., Las Vegas 26-29 June 2006, (ISBN), Vol. 1, pp. 408-413: CSEA Press.

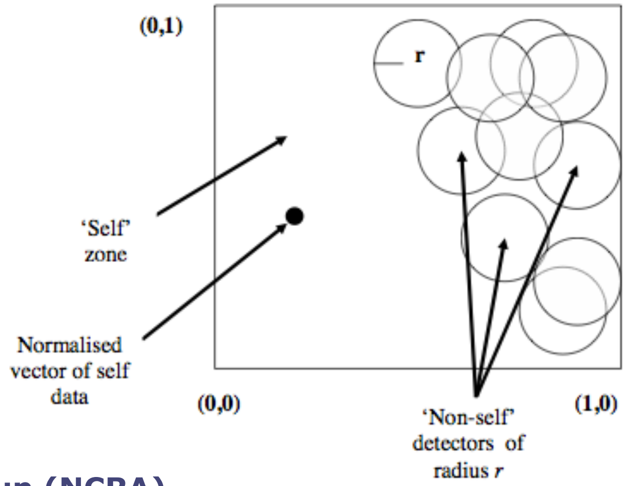
Le-Khac,, N. A., Markos, S., O'Neill, M., Brabazon, A. and Kechadi, M. T. (2009). An investigation into Data Mining approaches for Anti-money Laundering, IEEE International Conference on Knowledge Discovery, (ICKD'09), Manila, Philippine, 6-8 June 2009, IEEE Press.





Cahill, J., Keenan, P., Walsh, D. and Brabazon, A. (2010). Identifying Online Credit Card Fraud using Artificial Immune Systems, Proceedings of the 2010 IEEE Congress On Evolutionary Computation, IEEE Press.

Brabazon, A., Delahunty, A., O'Callaghan, D., Keenan, P. and O'Neill, M. (2007). Financial Classification using an Artificial Immune System, in Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications, Sugumaran, V. (ed), pp. 1525-1540, Hershey, PA, USA: Information Science Reference.



Model Induction

"I'm not superstitious either, but those were the three days Harris wore his lucky socks."

Agent-based Modelling of Markets (ASM)

- Actual market history is a single-sample path
  - Observation does not allow closed world ‘what-if’ analyses

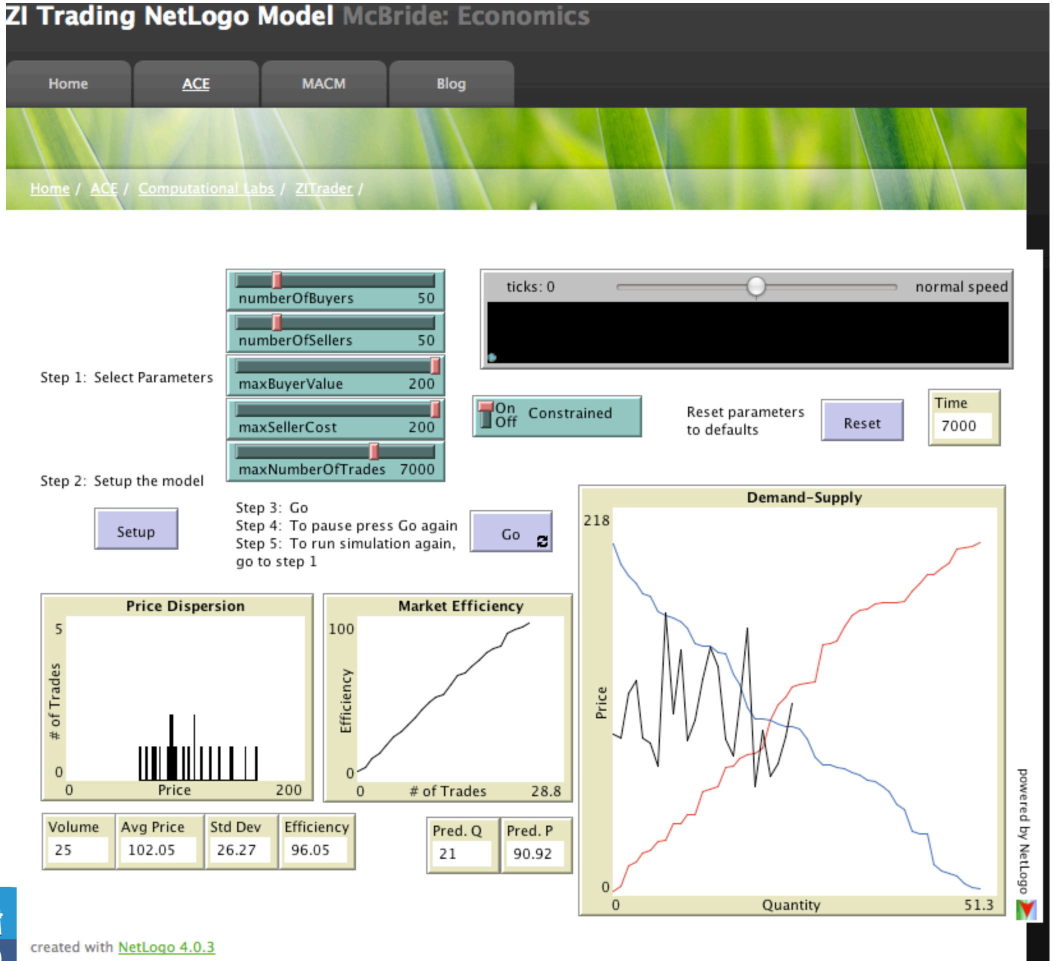
- ABM adopts a ‘bottom up’ approach to ‘simulate’ financial markets (markets ‘in silico’)
  - Autonomous agents whose interactions produce complex, emergent, system dynamics
    - Can embed heterogeneous agents with differing risk attitudes, with differing expectations to future outcomes, and with learning/adaptive capabilities
    - How do agents learn? (application of NC … GP, NN … etc.)
    - The ‘Red Queen’

Agent-based Modelling of Markets (ASM)

- Plausibility testing of ABM / Model validation (?)
  - Can we replicate observed real world characteristics of markets?
  - What are the critical assumptions in our ABM which are necessary to reproduce stylised real-world phenomena ?

- Applications of ABM
  - Attempt to explain market behaviour (theory building)
  - Provide insights for policy makers and regulators
  - Provide test-bed for ‘simulation’ of trading strategies / execution strategies

Agent-based Modelling of Markets (ASM)



Source: Mark E. McBride
Economics , Miami University

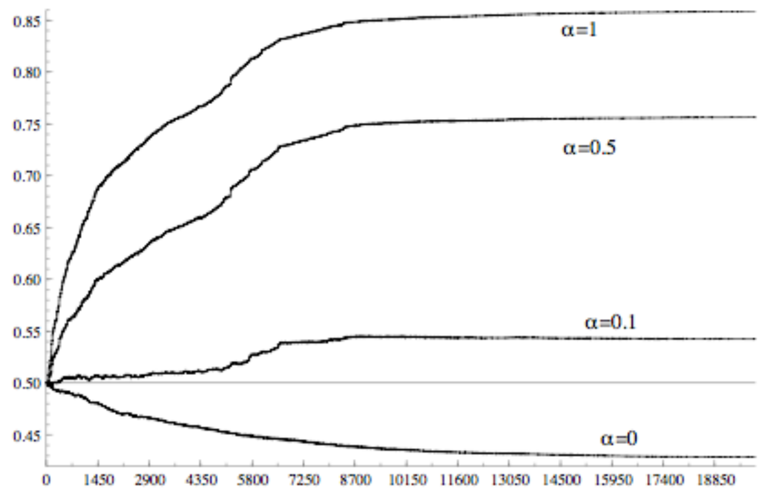
Model based on Gode and Sunder (1993)

http://mcbridme.sba.muohio.edu/ace/labs/zitrade/zitradenetlogo.html

ABM for Understanding

- Market fraction asset pricing model in order to investigate the market dominance, the profitability, and the survival rates of both fundamental and trend-following investors across varying time scales

- Simulation results results indicate that in contrast to the prediction of traditional financial theory, trend-followers can survive in the market in the long run and in the short run they can outperform fundamentalists



Time series of absolute wealth accumulation of the fundamentalists for varying levels of alpha (confidence of fundamentalists in their estimates of fundamental value). Note, trend followers – technical traders - persist (survive) at all levels of alpha.

He, X-Z., Hamill, P. and Li, Y. (2008) Can Trend Followers Survive in the Long-Run?: Insights from Agent-Based Modeling, , in Natural Computation in Computational Finance (Volume I), Brabazon, A., O'Neill, M. (eds), pp. 253-269, Berlin: Springer.

ASM as a Test Bed

- How can investors buy/sell large orders efficiently?
- The *traders dilemma* ....

trade quickly → Market impact

trade slowly → Opportunity cost

- We need to balance *Market Impact* (embeds cost of 'liquidity demand' and 'information leakage') and *Opportunity cost*

Trade Execution Strategy

- Trade execution requires the ‘design’ of an appropriate ‘trading strategy’
  - Order scheduling (number of sub-orders, their size, when submitted to market)
  - Order aggressiveness (what ‘style’ to adopt?)
  - Split of order across markets

- A practical problem then is how do we assess the utility of a strategy without actually implementing it?

- Scope to ‘test’ in an ASM

Cui, W., Brabazon, A. and O’Neill, M. (2011). Adaptive Trade Execution Using a Grammatical Evolution Approach, International Journal of Financial Markets and Derivatives.

CALL FOR PAPERS

EVO* 2012

EvoFIN 2012
6th European event on Evolutionary and Natural Computation in Finance and Economics
11-13 April 2012
Malaga, Spain

EvoFIN 2012 focuses primarily on the use of EC and related Natural Computing techniques in Computational Finance and Economics.

Applications of interest include (but are not limited to) forecasting financial time series, portfolio selection and management, estimating econometric parameters, pricing options, and developing risk management systems. Other applications of interest include artificial stock markets.

Methodologies of interest include EC methods such as genetic programming, genetic algorithms and evolutionary strategies, as well as other related natural computing methodologies such as particle swarm, foraging algorithms, artificial immune systems, hybrid systems and agent-based systems.

Conference proceedings are published as part of Springer’s LNCS series

Submission deadline: 30th November 2011

http://www.evostar.org