Biologically-inspired Algorithms for Financial Modelling

Prof. Anthony Brabazon

CFA Seminar
26 February 2010

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

• Introduction to FMC$^2$

• Earning a living ...

• Biologically-inspired algorithms

• A few financial applications ...
**Financial Mathematics and Computation Cluster (FMC\textsuperscript{2})**

**SFI Strategic Research Cluster**

**Lead PI:** Prof. Anthony Brabazon  
**Lead Institution:** UCD

**CO PIs**
- Prof. Paulo Guasoni, DCU / Boston  
- Prof. Gregory Connor, NUIM  
- Dr. Michael O’Neill, UCD  
- Dr. David Edelman, UCD  
- Prof. John Cotter, UCD

**Pioneer Investments**
- Ryan Capital  
- Institute of Bankers  
+ … + …

**Core Researchers**
- 6 Lead academics,  
- 15 academic collaborators (from 5 other universities),  
- 3 Post Docs,  
- 15 PhD Researchers

**Award Size**
- €5.3m (inc. overheads)

**Term**
- 5 Years  
- Commencing Sep. 2009

- Cluster’s research programme focuses on the development of theory and methods for the task of asset management

- Cluster has a strong focus on industry engagement via direct research collaboration with industry partners - thereby supporting partner product / process innovation
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

Recent Highlights


- Dr. O'Neill and Prof. Brabazon co-Chairs EvoFIN 2010, the 4th European Event on Evolutionary and Natural Computing in Finance and Economics, Istanbul, Turkey (7-9 April 2010) Published in Springer's LNCS.


- Prof. John Cotter appointed to the Steering Committee of the Irish Chapter for PRMIA (Professional Risk Manager’s International Association).
Industry Engagement

- **The Ideal Partner ...**
  - We are happy to engage with companies of all sizes
  - Project within broad scope of cluster research themes
  - Industry partner willing to commit resources to collaboration (e.g. data, personnel time, money etc.)
  - Timescale: generally 12 months +

- **Engagement Process**
  - Each project is unique ...
  - ‘Signpost’ - ‘Vocabulary Sharing’ - ‘Pilot’ - Full Spec
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 ...

• 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.

• How can we find good ‘rules’ for surviving in this (or any other...) environment?
Biologically-inspired Algorithms

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

• Mechanisms have arisen which assist the ‘survivability’ / adaptability (robustness) 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)
Evolutionary Cycle

Distinction between *genotype* and *phenotype*

Evolution can be considered as being a *search process* in genotypic space but the ‘worth’ of the genotypes is only assessed at phenotypic level.
Evolutionary Cycle
Evolutionary Computation

Initialise population

\textbf{WHILE} (Termination condition False)

\begin{itemize}
  \item Calculate fitness of each individual
  \item Select parents
  \item Create offspring
  \item Update population
\end{itemize}

\textbf{ENDWHILE}

The canonical evolutionary algorithm is the “\textit{genetic algorithm}” - primarily used for optimisation purposes

UCD Natural Computing Research & Applications Group (NCRA)
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

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>Fitness (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.1245</td>
<td>5.6219</td>
<td>11.3411</td>
<td>0.3245</td>
</tr>
<tr>
<td>2</td>
<td>-4.5612</td>
<td>-0.2317</td>
<td>6.1311</td>
<td>0.7436</td>
</tr>
<tr>
<td>3</td>
<td>2.3412</td>
<td>1.6432</td>
<td>2.7811</td>
<td>0.6718</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>-3.6245</td>
<td>4.8219</td>
<td>13.3411</td>
<td>0.3015</td>
</tr>
</tbody>
</table>
## A Simple Optimisation Example

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>Fitness (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.1245</td>
<td>5.6219</td>
<td>11.3411</td>
<td>0.3245</td>
</tr>
<tr>
<td>2</td>
<td>-4.5612</td>
<td>-0.2317</td>
<td>6.1311</td>
<td>0.7436</td>
</tr>
<tr>
<td>3</td>
<td>2.3412</td>
<td>1.6432</td>
<td>2.7811</td>
<td>0.6718</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>-3.6245</td>
<td>4.8219</td>
<td>13.3411</td>
<td>0.3015</td>
</tr>
</tbody>
</table>

Suppose these two ‘good’ parents are chosen.
A Simple Optimisation Example

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>Fitness (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child 1</td>
<td>-3.3745</td>
<td>5.2219</td>
<td>12.3411</td>
<td>...</td>
</tr>
</tbody>
</table>

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)
### A Simple Optimisation Example

Next, apply a mutation operator to the child and determine the fitness of the child.

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

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>Fitness (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child 1</td>
<td>-3.3745</td>
<td>5.2219</td>
<td>12.6500</td>
<td>0.2918 (say)</td>
</tr>
</tbody>
</table>
**A Simple Optimisation Example**

<table>
<thead>
<tr>
<th></th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>Fitness (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child 1</td>
<td>-3.3745</td>
<td>5.2219</td>
<td>12.6500</td>
<td>0.2918 (say)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Child n</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- 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 $\beta_0, \beta_1, \beta_2$
Crossover

- Crossover
  - Aims to use information from better parents ….
  - Could implement ‘intermediate crossover’ (a simple average)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent 1</td>
<td>-3.1245</td>
<td>5.6219</td>
<td>11.3411</td>
</tr>
<tr>
<td>Parent 2</td>
<td>-3.6245</td>
<td>4.8219</td>
<td>13.3411</td>
</tr>
<tr>
<td>Child</td>
<td>-3.3745</td>
<td>5.2219</td>
<td>12.3411</td>
</tr>
</tbody>
</table>

- 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 \( \alpha \) is a scaling factor (perhaps randomly drawn from \([-2, +2]\))

- Defines a hypercube based on the location of the parents
Mutation

- Mutation
  - Allows the uncovering of new information that was not present in either parent
  - Could add 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 $\alpha_i$ is user-defined
A slightly more complex genotype-phenotype mapping ... 

• How might you represent a simple technical trading rule of the following form as a string?

\[
\text{IF } x \text{ day MA of price } \geq y \text{ day MA of price} \\
\text{THEN Go Long ELSE Go Short}
\]

\[
\begin{array}{c|c|c}
5 & 10 & 1 \\
\end{array}
\]

( 5 day MA \geq 10 day MA THEN Go Long)
Back-testing the quality of a genotype...
A slightly more complex genotype-phenotype mapping ...

\[
IF \ [\text{Indicator}_i(t) \ (<,>) \ \text{value}_j] \ THEN \ \text{(Buy, Sell, Do nothing)}
\]

<table>
<thead>
<tr>
<th>Indicator_i</th>
<th>t</th>
<th>&lt;,&gt;</th>
<th>value_j</th>
<th>Buy, Sell, Do nothing</th>
</tr>
</thead>
</table>

- The above are simple 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
Evolving an Asset Selection Rule

<table>
<thead>
<tr>
<th>High sales growth relative to industry average?</th>
<th>High debt level relative to industry average?</th>
<th>High level of cash flow from operations relative to industry average?</th>
<th>High level of liquidity relative to industry average?</th>
<th>High profit level relative to industry average?</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Filter 1</th>
<th>...</th>
<th>Filter n</th>
<th>R/σ (say)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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

UCD Natural Computing Research & Applications Group (NCRA)
Genetic Programming

• An evolutionary model-induction methodology

• Idea dates from the 1960s, 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

- These solutions could be many things ....
  - A computer program (evolutionary automatic programming)
    - a trading system
    - an asset pricing model
    - A credit risk assessment model etc. etc.

- Unlike GA, GP adopts a variable-length representation
  - In GA the number of elements that comprise a genotype is fixed at the start of the run
  - In GP the length of a solution is not assumed to be known *a priori* and is instead evolved

- Hence the *structure* as well as parameters of the solution is evolved
Genetic Programming

• Individual is or represents a program

```c
#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 decision, investment decision, ….
Genetic Programming
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
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?
  
  - Utility in building forecasting models ... but also in theory development ...

• As each ‘model’ is evolved, it’s quality / fitness can be assessed by determining how well it explains the observed (training) data
Financial Application Areas of NC

• Optimisation
  - Combinatorial & real-valued optimisation
  - Can be used for parameter estimation / model calibration, variable selection
  - GA, DE, ACO, PSO …

• Model induction
  - Can be used to uncover both model structure and parameters
  - ANNs (universal approximators ... but have practical problems concerning the readability of models), GP / GE (produce readable rules and can easily embed human knowledge)
  - Weak theory ... plentiful data?

UCD Natural Computing Research & Applications Group (NCRA)
Financial Applications ...

- Recent Financial Applications of GE
  - Asset selection
  - Options pricing
  - Credit risk assessment
  - Algorithmic trading
  - Money-laundering detection
  - ......
Contact Details:

Prof. Anthony Brabazon
Head of Research
Quinn School of Business
University College Dublin
Belfield
Dublin 4
Ireland

**Telephone:** + 353 - 1 - 716 4705 / 5301

**Email:** anthony.brabazon@ucd.ie

**Web:** http://www.fmc-cluster.org