# Picking Stocks

In this project, you will create a genetic algorithms implementation that uses historical data to create a good rule for stock market trading. This is actually an area of professional application for genetic algorithms, although our solution will be very simplistic in comparison with the more intense and robust solutions that stock traders use (and they still lose money sometimes). *Disclaimer: assigning this project is in no way an endorsement or encouragement that you should use your generated solution in real-world stock trading using actual money.*

You will implement your solution in PHP or Python; if you would like to utilize a different language, please get instructor approval before beginning.

Trading Rules

Three basic trading rules are used in this experiment:

(1) Simple Moving Average (SMA): in this rule, if the actual closing price of a share is higher than the SMA over N days, then you buy; otherwise, you sell.
SMA gathers data from N days and weights all data points equally.

$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

(2) Exponential Moving Average (EMA): in this rule, if the actual closing price of a share is higher than the EMA over N days, then you buy; otherwise, you sell.
EMA gathers data from N days but weights recent data more than older data.

$$\text{EMA}_{\text{today}} = \frac{p_1 + (1-\alpha)p_2 + (1-\alpha)^2 p_3 + (1-\alpha)^3 p_4 + \cdots}{1 + (1-\alpha) + (1-\alpha)^2 + (1-\alpha)^3 + \cdots}$$ for $\alpha = 2 / (N+1)$

(3) Maximum (MAX): in this rule, if the actual closing price of a share is greater than the maximum closing price from N preceding days (not including this one), you buy; otherwise, you sell. MAX compares only to the previous known high point.

Individuals

Each individual in your population represents a rule, either be one of the basic trading rules or a combination of basic rules using the logical connectors AND or OR.
Each individual will be encoded as a string of 14 characters, with three sections indicating the rule applied over a number of days and connected logically, such as:

- `s050&e000&m000` – this individual acts on SMA in 50-day period ("s050").
  For at least 50 days (until there is enough history), this rule will not buy anything.
  Note that a 0-day period indicates the rule is not applied ("e000", "m000"). In these cases, it should have *no effect* on the buy/sell decision, which means logically that in a condition AND, it will be treated as true; in OR, it will be treated as false.
- `e025|m093&s000` – this individual acts on the combination of EMA in 25-days ("e025") or MAX in 93-days ("m093"); the "s000" component has no effect.
- `s050&e030&m010` – this individual acts on the combination of SMA in 50-days ("s050") and EMA in 30-days ("e030") and MAX in 10 days ("m010").
  Rules are evaluated from left to right, so this is (SMA 50 AND EMA 30) AND MAX 10.
- `s075&m002&s088` –this individual is included as an example to emphasize that two or more components can apply the same basic rule (here, SMA).

Initial Population
You may define your initial population in any reasonable manner. As a starting point, perhaps use genomes variously each implementing the basic rule over different time periods, such as: `s010&e000&m000`, `s025&e000&m000`, `s000&e010&m000`, `s000&e025&m000`, `s000&e000&m010`, `s000&e000&m025`, and some random combinations of rules, such as: `s043&e057&m109`, `s083&e100&m036`, `s011|e140|m040`, `s052|e130&m024`, *etc*.

Crossover
The two individuals that are taking part in a crossover are selected randomly from the intermediate population, with a single-crossover point chosen randomly in (0, 13). You may adjust the crossover rate as you test but a starting point of 0.8 is recommended, so that the two selected individuals will be copied at a rate of 0.2.

Mutation
Mutation is performed by changing a character in the string that represents an individual with another appropriate character. A basic rule {s,e,m} must be replaced by one of the other rules; a digit can be replaced by any digit 0..9; a logical operator will be "flipped" { '&' / '|' }.
Each character in the string is changeable under a defined mutation rate, which you may adjust as you test but a starting rate of 0.01 is recommended.

Population Size and Halting Condition

Your population size in each generation will be 50 individuals and you will run 200 generations from the initial population. When complete, output the individual discovered with best fitness.

Fitness

Individuals are evaluated based on the performance of the rule that they represent. While many aspects of this project are adjustable, all teams will utilize the same fitness evaluation to establish common comparison between results.

Fitness will be evaluated by applying the rule using historical price data over five periods {01/2000-12/2002, 01/2003-12/2003, 01/2008-12/2009, 01/2013-12/2014, 01/2017-12/2019} for five different companies {AAPL, F, GE, RTX, DDS}.

In each of the 25 cases (time/company), $20,000 is provided as starting capital. The profits obtained after selling shares are taken out of the account and kept in a "gain" account. If the capital account balance falls below $20,000, money will be moved from the gain account to restore the capital account to $20,000 (or emptying the gain account, if not enough is available). Each buy or sell transaction has a $10 fee applied, taking that money from the capital account.

When the end date is reached, any shares that are held must be sold using the last closing price. Rules that did not buy or sell shares at all during the time period are penalized, reducing the money by half; this is designed to avoid a convergence to non-participatory rules. The total final money amounts from all companies over all time periods are summed together – this total value is the fitness of the individual.

Experiment

You will develop a best-fit individual (let's call it Hero) by Tuesday, March 28, when you submit the project code. At that time, you will be given new data for testing.

First, you will be given data for a different set of companies and time periods. You will use this historical data as a basis to compare Hero against each of the basic rules (SMA, EMA, MAX) applied individually; these results will be included in your summary paper.
Here you will not use your genetic algorithms solver to build a solution, just compare Hero's performance against naïve applications of the basic rules.
For this experiment, two changes will be applied: (1) a starting account of $100,000 is provided instead of $20,000, and (2) there is no penalization for a rule that chooses not to buy at all.

Second, you will be given a new single company and time period. You will use the historical data as a basis to compare the gains of your best-fit individual in a 3-month, 6-month, 12-month, 24-month and 48-month period for that company within the given time period.

Summary Paper

Along with submitting your project code, you will write a paper (fitting the ACM template provided) that summarizes your implementation decisions and addresses Hero's performance using results gathered from the experiments. This paper will be due Monday, April 3.