# Algorithmic Currency Trading using NEAT - based Evolutionary Computation

Dan Hu
Omar Chowdhury

May 9, 2014

**Abstract**

This paper introduces NEAT-based Evolutionary Computation as the basis for a fully automated trading system application. The system is designed to trade FX markets by detecting profitable currency cycles in the most widely traded currencies and forecasting future exchange rates. To do this, it relies on a fitness function that measures profitability, as well as a fitness function that measures accuracy of future predictions. The trading system is able to generate consistent gains based on this back-tested and predictive approach.

## 1 Background

### 1.1 Algorithmic Trading

Algorithmic Trading (sometimes referred to as High Frequency Trading – more on that later) is a broad blanket term for all electronic trades that are computerized or placed by an algorithm that has been designed to execute automated transactions. These systems are typically implemented both on public exchanges and on dark pools (specialized exchanges utilized by financial institutions and sophisticated traders; not open to the public), the latter of which is serviced nearly exclusively through algorithmic trade tools. The industry is relatively new, as the earliest algorithmic trading systems originated in the early 2000s. Since then, a plethora of market participants, including economists, mathematicians, computer scientists, data analysts, actuaries, and financiers have contributed to the field and the development of algorithmic trade systems.

### 1.2 Speed and Data Processing

Algorithmic trading systems are conceptually uncomplicated. Unlike the aggressive traders hollering across an expansive floor to communicate, algorithmic trading is largely data-driven; systems must be able to evaluate information and predict how economic and political events will impact relationships between asset prices. Because prices are interconnected and affected by global influences, algorithmic systems' ability to quickly process information is far superior to that of human traders. One brokerage firm estimates that a 1-millisecond advantage in trading applications is worth approximately $100 million.[4] Thus, the role of the algorithmic trading system is to receive information as input, evaluate the data, and output a specific trading position. The speed and purpose of these algorithms varies - some algorithms place thousands of trades within a fraction of a second (high-frequency or low-latency), others may place a single trade every few months.

### 1.3 Widespread Prevalence

Currently, there are dozens of products that can be traded via algorithms. Depending on the type of product, anywhere from 50-70% of all trades are placed by algorithms today.[1] There are a number of well-researched strategies that have been proven to generate positive returns. Arbitrage is one such example - it entails the simultaneous purchase and sale of identical products in different markets that exploits a price difference. This strategy can be applied to many products, although it is most commonly implemented on equities and currencies. Currency triangles demonstrate this perfectly (where exchange rate differences are exploited by say, purchasing the US Dollar, exchanging it for the Mexican Peso, then the Japanese Yen, and then back to the US Dollar).

## 2 Currency Trading

Currency Markets (or forex, or simply FX markets) are particularly interesting to study because they are large, liquid, and easily accessible. Approximately $5.3 trillion dollars worth of currencies trade every single day, 24 hours a day, except for the weekends.[5] This is equivalent to about $220 billion dollars per hour – the global equities markets in aggregate only trades about $181 billion dollars per day. The enormous volume that passes through FX markets creates thin spreads and thus the need for significant leverage (debt) to produce substantive profits.

Researchers have built thousands of models in an effort to explain and forecast the movement of exchange rates. These models are usually driven by a number of correlated factors - these include economic, political, and psychological reasons. Evolutionary computation is an emerging concept that offers promise in a variety of financial applications. It attempts to mimic the biological and evolutionary competition between individuals within a species where those with the best fitness are preserved and serve as the basis for the subsequent generation. The most important motivation for developing evolutionary computation is to increase the profit gain while also decreasing risk.

## 3 Currency Trade Example

Before moving on, it is important to first understand how a simple currency triangle trade is profitable. We will walk through an example here.

Assume the following exchange rates. The first currency listed is the denominated currency, so 1 EUR is equivalent to 1.3697 USD.

1. EUR/USD = 1.3697
2. JPY/EUR = 0.0071
3. USD/JPY = 101.625

If we were to begin trading with 100,000 USD, our first transaction would leave us with 73,008.688 EUR. With 73,008.688 EUR, our second trade would convert into 10,282,913.8 JPY. And finally, with 10,282,913.8 JPY, we would place our final trade and end up with 101,184.884 USD – effectively turning a profit of $1,184.884 dollars through this triangle.

In this example, the profit is just over 1%, which is highly unlikely in the actual market. Currency triangles are usually difficult to exploit because forex markets are extremely efficient. Leverage utilization ultimately remains the method that drives profit gains on an absolute basis.

## 4 Literature Review

Algorithmic trading strategies have been a heavily researched topic over the last decade. Surprisingly enough, there have been a fair number of attempts to develop an adaptive learning algorithm capable of trading in the active market. One of the earliest such papers was published at the National University of Singapore in 1996; the researchers hypothesized that currency prices traded within a mean bounded by the maximum and minimum cumulative deviations of the observations of a time series from its mean.[2] They used neural networks and divided historical data into three portions: training, validation, and testing sets. The training set contained two thirds of the data, while the validation and testing sets contained two fifteenths and three fifteenths respectively. Although the data set they used spanned 11 years, the actual data points consisted of weekly closing prices, which would be meaningless for active market trading.

Another alternative called for an agent to be trained using multi-layered adaptive reinforcement learning. Implemented by Dempster and Leemans, this method involves a layer for risk management (to prevent losses if the agent begins to make suboptimal trades), a dynamic optimization layer, and a recurrent layer.[3] It can make one currency trade in one timestep (that is it's RRL set of actions). Trades are constrained by a programmer-defined cost factor so the agent must make decisions about whether to go for low profit trades. The trades are then fed into the risk layer (which implements specific trading strategies to prevent sloppy trading) and then into an optimization layer that implements complex mathematical operations. Dempster and Leemans' strategy proved to be a profitable one.

A third technique calls for feeding alternative data into a trained or NEAT neural network. Although counterintuitive, Eric Simon specifically posits that because interest forecasting is complex, we should avoid inputting past exchange rate data and then subsequently expect to trade on the predicted values.[6] In his implementation, rates were forecasted using a multilayer neural network and back propagation. Data can be inputted along with medium or long run averages, and Simon also tried to introduce political factors as boolean variables. In his experiment, he chose to include the political party of the President and whether the United States was at war. Simon's concluding point is that his method is very flexible - various variables can be added seamlessly.

## 5 Hypothesis

We would like to use historical data to design an algorithmic trading model that not only identifies currency cycles, but also has predictive value: the ability to forecast future prices and profitably exploit price inefficiencies. We will be using Evolutionary Computation with an Elman (recurrent) network for the adaptive learning process. The competition between individual agents to find the most profitable trades within a generation will eliminate the underperformers, and with sufficient turnover, the best performers will always be retained. Over many generations, this process should engender a population adept at quickly recognizing the most profitable currency cycles.

## 6 Project Design

This project will be divided into two major components. The first step is to train a neural network using the Bellman-Ford algorithm on historical pricing data. Currency trading is a commonly used application for the Bellman-Ford algorithm; it is a simple method that can detect currency triangles without using adaptive learning. The second part entails creating a NEAT – based Evolutionary Computation method that will challenge individual agents within each generation to identify the most profitable currency cycles.

### 6.1 Bellman-Ford Shortest Path Algorithm

The Bellman-Ford is a graph algorithm that computes single source shortest paths in a weighted, directed graph. It determines single source shortest paths from an input vertex, to every other vertex in the graph. Unlike Dijkstra's algorithm, which requires the graph to have no negative edge weights, the Bellman-Ford algorithm permits the graph to have edges with negative weights, as well as negative weight cycles. It detects whether there is a negative weight cycle that is reachable from the source. The Bellman-Ford algorithm has a runtime of O(VE), where V is the number of vertices and E is the number of edges. Compare this to Dijkstra's algorithm, where the runtime is O(E + Vlog(V)). Thus, while Bellman-Ford is more costly than Dijkstra's in terms of speed, the benefit is that it is more versatile.

### 6.2 Bellman-Ford applied to Currency Trading

As mentioned before, Bellman – Ford affords the flexibility of using edges with negative weights. The exchange rates between currencies are assigned to be the edge weights of the graph. If the product of the exchange rates in a cycle is greater than 1, then a profitable triangle exists and can be traded on. Conversely, if the product of the exchange rates in a cycle is less than 1, then there exists a losing strategy. Thus, profitable strategy detection comes down to finding a triad of exchange rates whose product exceeds 1. When currencies are mapped using the Bellman-Ford, the graph is fully connected - this is because every currency can be exchanged for any other currency. Thus, adding new currencies to the graph increases the graph's complexity exponentially.

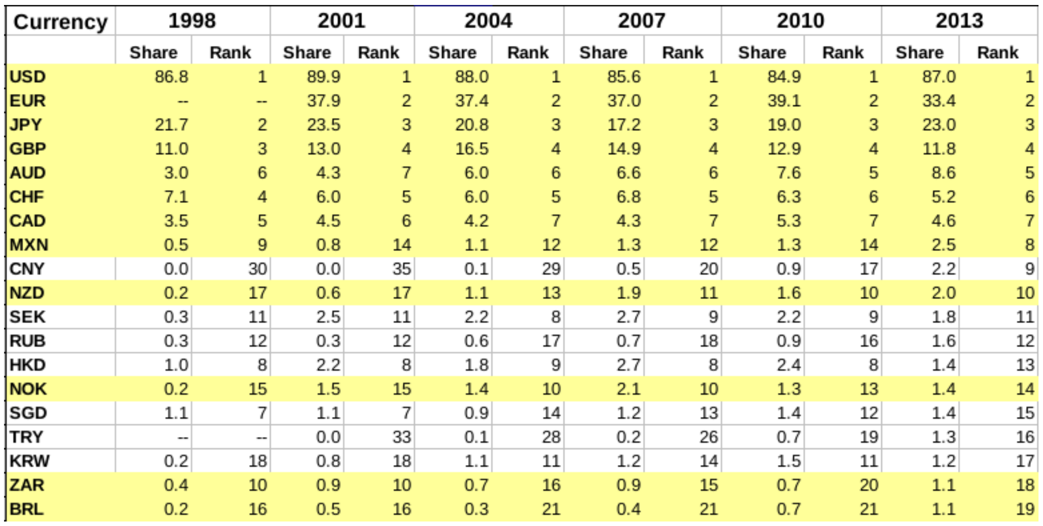
Ultimately, our simulation using the inputs from the Bellman-Ford network on the trained neural network yielded inconclusive results. We only used 3 currencies (USD, EUR, and GBP), and the profits that were generated were within floating point error.

### 6.3 Overview of NEAT and Evolutionary Computation

Using the Bellman-Ford did not produce substantive results, so we posited that a NEAT based evolutionary computation method would be better at determining the most profitable currency trades. Neuroevolution of Augmenting Topologies (NEAT) was developed by Ken Stanley of the University of Texas; it is a genetic algorithm that evolves artificial neural networks and encourages individual fitness, diversity, and complexity within a generation over time. Specifically, we can apply this to a population of evolutionary agents by feeding them historical exchange rates. Each individual within a generation identifies profitable trades and outputs the appropriate trade order. The fitness score strictly measures the profitability of each individual's trades, and determines which individuals will be included or excluded from the subsequent generation.

## 7 Currency Mix

For our evolutionary algorithm, we expanded our currency basket to include 12 different currencies, an increase from the three that we used for our Bellman-Ford simulation. All 12 are among the 20 most liquid currencies in the world; in order of market share, they are: USD, EUR, JPY, GBP, AUD, CHF, CAD, MXN, NZD, NOK, ZAR, BRL. Incidentally, the first eight in that list are the eight most widely traded in the world, and the NZD is the 10th most traded. We omitted the Chinese Yuan, the ninth most traded, because the Yuan's value has been artificially deflated for the past several years in order to sustain China's export-led economy. Instead, we substituted the Yuan for the Norwegian Krone, which is often viewed as one of the safest currencies in the world (along with the Swiss Franc). To round out our dozen, we included the South African Rand and the Brazilian Real, because both South Africa and Brazil are developing countries whose currency values fluctuate often. This is good because it will really challenge the evolutionary agents to discover profitable trades between a host of both stable and volatile currencies.



| Currency | 1998 | | 2001 | | 2004 | | 2007 | | 2010 | | 2013 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Share | Rank | Share | Rank | Share | Rank | Share | Rank | Share | Rank | Share | Rank |
| USD | 86.8 | 1 | 89.9 | 1 | 88.0 | 1 | 85.6 | 1 | 84.9 | 1 | 87.0 | 1 |
| EUR | -- | -- | 37.9 | 2 | 37.4 | 2 | 37.0 | 2 | 39.1 | 2 | 33.4 | 2 |
| JPY | 21.7 | 2 | 23.5 | 3 | 20.8 | 3 | 17.2 | 3 | 19.0 | 3 | 23.0 | 3 |
| GBP | 11.0 | 3 | 13.0 | 4 | 16.5 | 4 | 14.9 | 4 | 12.9 | 4 | 11.8 | 4 |
| AUD | 3.0 | 6 | 4.3 | 7 | 6.0 | 6 | 6.6 | 6 | 7.6 | 5 | 8.6 | 5 |
| CHF | 7.1 | 4 | 6.0 | 5 | 6.0 | 5 | 6.8 | 5 | 6.3 | 6 | 5.2 | 6 |
| CAD | 3.5 | 5 | 4.5 | 6 | 4.2 | 7 | 4.3 | 7 | 5.3 | 7 | 4.6 | 7 |
| MXN | 0.5 | 9 | 0.8 | 14 | 1.1 | 12 | 1.3 | 12 | 1.3 | 14 | 2.5 | 8 |
| CNY | 0.0 | 30 | 0.0 | 35 | 0.1 | 29 | 0.5 | 20 | 0.9 | 17 | 2.2 | 9 |
| NZD | 0.2 | 17 | 0.6 | 17 | 1.1 | 13 | 1.9 | 11 | 1.6 | 10 | 2.0 | 10 |
| SEK | 0.3 | 11 | 2.5 | 11 | 2.2 | 8 | 2.7 | 9 | 2.2 | 9 | 1.8 | 11 |
| RUB | 0.3 | 12 | 0.3 | 12 | 0.6 | 17 | 0.7 | 18 | 0.9 | 16 | 1.6 | 12 |
| HKD | 1.0 | 8 | 2.2 | 8 | 1.8 | 9 | 2.7 | 8 | 2.4 | 8 | 1.4 | 13 |
| NOK | 0.2 | 15 | 1.5 | 15 | 1.4 | 10 | 2.1 | 10 | 1.3 | 13 | 1.4 | 14 |
| SGD | 1.1 | 7 | 1.1 | 7 | 0.9 | 14 | 1.2 | 13 | 1.4 | 12 | 1.4 | 15 |
| TRY | -- | -- | 0.0 | 33 | 0.1 | 28 | 0.2 | 26 | 0.7 | 19 | 1.3 | 16 |
| KRW | 0.2 | 18 | 0.8 | 18 | 1.1 | 11 | 1.2 | 14 | 1.5 | 11 | 1.2 | 17 |
| ZAR | 0.4 | 10 | 0.9 | 10 | 0.7 | 16 | 0.9 | 15 | 0.7 | 20 | 1.1 | 18 |
| BRL | 0.2 | 16 | 0.5 | 16 | 0.3 | 21 | 0.4 | 21 | 0.7 | 21 | 1.1 | 19 |

Figure 1: This is a list of the 19 most widely traded currencies. The currencies we used are highlighted in yellow

## 8 Data Collection and Processing

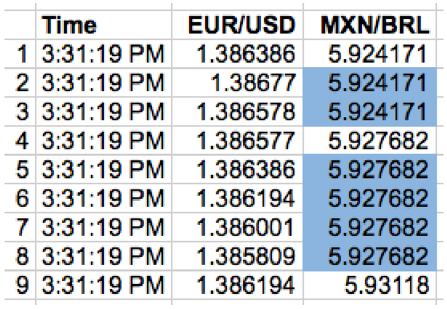
### 8.1 Collection

All of the data that we used was collected from the Bloomberg Terminal, a computer system containing a large database of financial information including real time and historical price quotes. We chose to fetch data spanning 9:30AM to 4:00PM on April 10th, 2014 - the busiest time in a 24-hour period as it coincides with the opening and closing of US Equity Markets. April 10th was also a day in the recent past where currency markets were calm. The reason we selected a tempered trading day instead of a volatile session is because profits are less easily generated when the market behaves rationally and moves slowly. During a choppy or turbulent period, mispricings and inefficiencies abound, and profits are easier to come by. Thus, we wanted to test the integrity of our system within a harsher scenario. In total, this time span yielded 2,816,108 data points for us, each one corresponding to an executed trade.

Within the seven and a half hour period, we reduced our scope even further to concentrate on a forty minute period. Between 3:20 and 4:00 PM, equity traders are fully occupied in wrapping up their day and positioning themselves for the next day, and the currency markets benefit from this activity. Our data confirmed this, no other forty minute period saw as many executed trades as the one coinciding with the close of the equity markets.

### 8.2 Extrapolation

Even within our highly active 40 minute trading period, there was a complication within our data that required adjusting. Despite the fact that we selected the most frequently traded currencies, the most popular ones were still out-trading the less popular ones. That is, the most liquid pairs in our data set would fire off more than a hundred trades within a second at varying prices, while some of the slower pairs would trade once every two or three seconds. To counteract the effect of the faster trades, we needed to "speed up" the slower trades in order to normalize the data set. We did this by extrapolating the last known price for the less frequently traded currencies wherever a gap was created by the more frequently traded ones. An example of the adjustments we made is provided here:



| | Time | EUR/USD | MXN/BRL |
|---|---|---|---|
| 1 | 3:31:19 PM | 1.386386 | 5.924171 |
| 2 | 3:31:19 PM | 1.38677 | 5.924171 |
| 3 | 3:31:19 PM | 1.386578 | 5.924171 |
| 4 | 3:31:19 PM | 1.386577 | 5.927682 |
| 5 | 3:31:19 PM | 1.386386 | 5.927682 |
| 6 | 3:31:19 PM | 1.386194 | 5.927682 |
| 7 | 3:31:19 PM | 1.386001 | 5.927682 |
| 8 | 3:31:19 PM | 1.385809 | 5.927682 |
| 9 | 3:31:19 PM | 1.386194 | 5.93118 |

Figure 2: An example of how some currencies trade relative to others. Extrapolated values are highlighted in blue

As you can see, the MXN/BRL is less frequently traded than the EUR/USD. The values highlighted in blue indicate our extrapolation (not actual trades). In the time it takes three trades between the MXN/BRL to execute, nine exchanges between the EUR/USD have occurred. Consequently, we assume that the exchange rate at time steps 2 and 3 are equivalent to time step 1, since the price has not yet changed. At time step 4, the price changes, and thus we assume that up until time step 9, the price in time steps 5-8 remains what it was at time step 4.

## 9 NEAT Implementation

Upon successfully collecting and adjusting our data, we were ready to run our NEAT experiment. Utilizing a basic NEAT implementation with a population size of 50 individuals, each individual's input nodes was fed the 132 exchange rates (12 currencies each exchangeable for the other 11). At each generation, we randomly selected a block of 1000 consecutive trades to be processed by the evolutionary agents. The purpose of this was to challenge the evolutionary agents to make sense of a seemingly nonsensical set of data. Each agent then produced an output containing various activations, which are numbers greater than 0 and less than 1. These activations signal a trading order; the highest activation corresponds to the denominated currency, the second highest activation is the first trade, and the third highest activation is the second trade. At a certain point, the output activation falls below a certain threshold, which is where the current currency is exchanged for the originally denominated currency. The fitness function applied to this experiment is simple: degree of profitability is the only measurement.

In a follow-on experiment, we investigated the predictive ability of the algorithm. The fitness function was manipulated accordingly; at each time step t the evolutionary agents were asked to predict the exchange rates two time steps ahead at time t+2. This experiment was important to our ultimate objective. Pattern and cycle recognition is only the first component of algorithmic trading. Upon detecting currency cycles, the trading system must immediately act on profitable trades by accurately predicting the direction of the currencies within the near future.

## 10 Results

In our first experiment where we used average profit as the fitness function, we discovered that the fitness scores began to plateau by the 15th generation. The same was true for our second experiment, where we tested for predictive ability. The marginal improvement in each of the early generations was significantly large, which suggests that the evolutionary agents were able to learn relatively quickly. Furthermore, the fact that a performance ceiling was reached after such a small number of generations indicates that there is a finite limit to how much profit can be generated, and how well the agents can predict future exchange rates. This is unsurprising because currency cycles are limited in size and profitability, and they rarely deviate from a mean range relative to one another.

Below is the fitness graph produced from the first experiment - profitability only.. The population's average fitness is denoted in blue, and the best fitness is denoted in red.

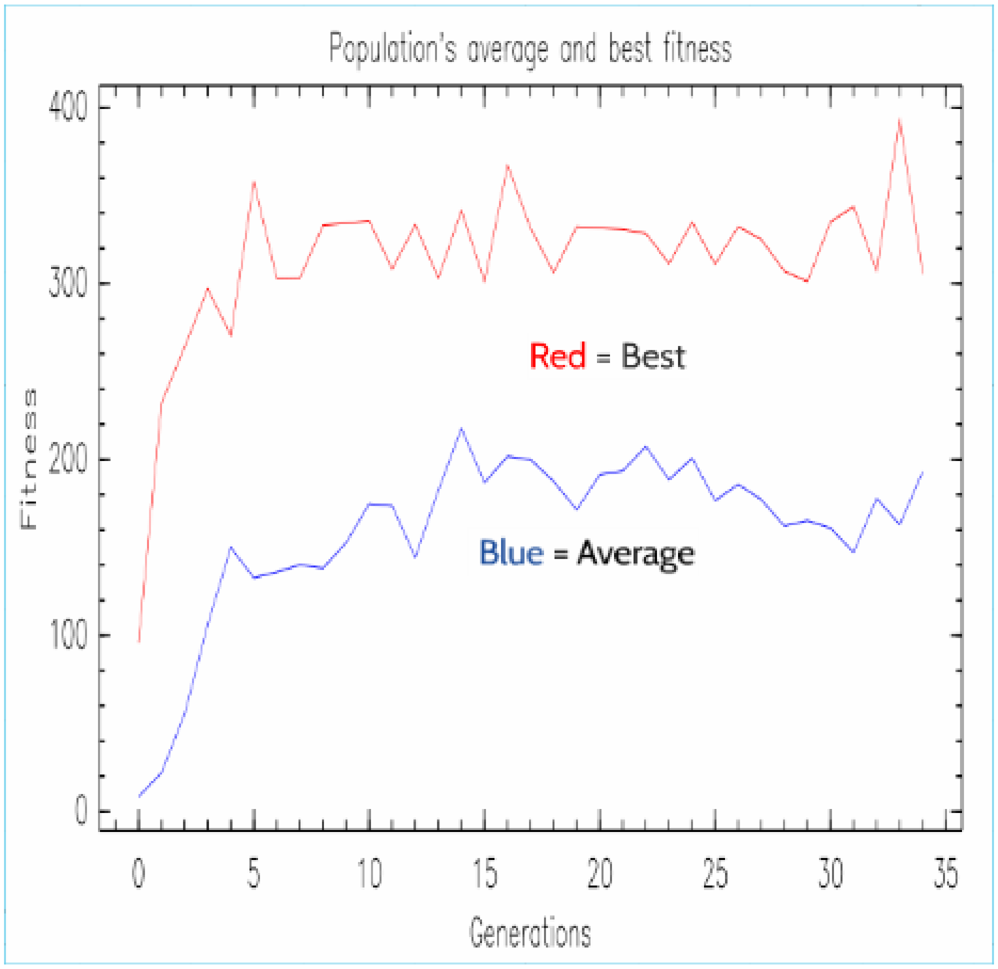


Figure 3: The average and best fitness of each population, measured by profitability

The next diagram is produced from our second experiment - predictive ability only at time t+2 time steps. Again, the population's average fitness is denoted in blue, and the population's best fitness is denoted in red.

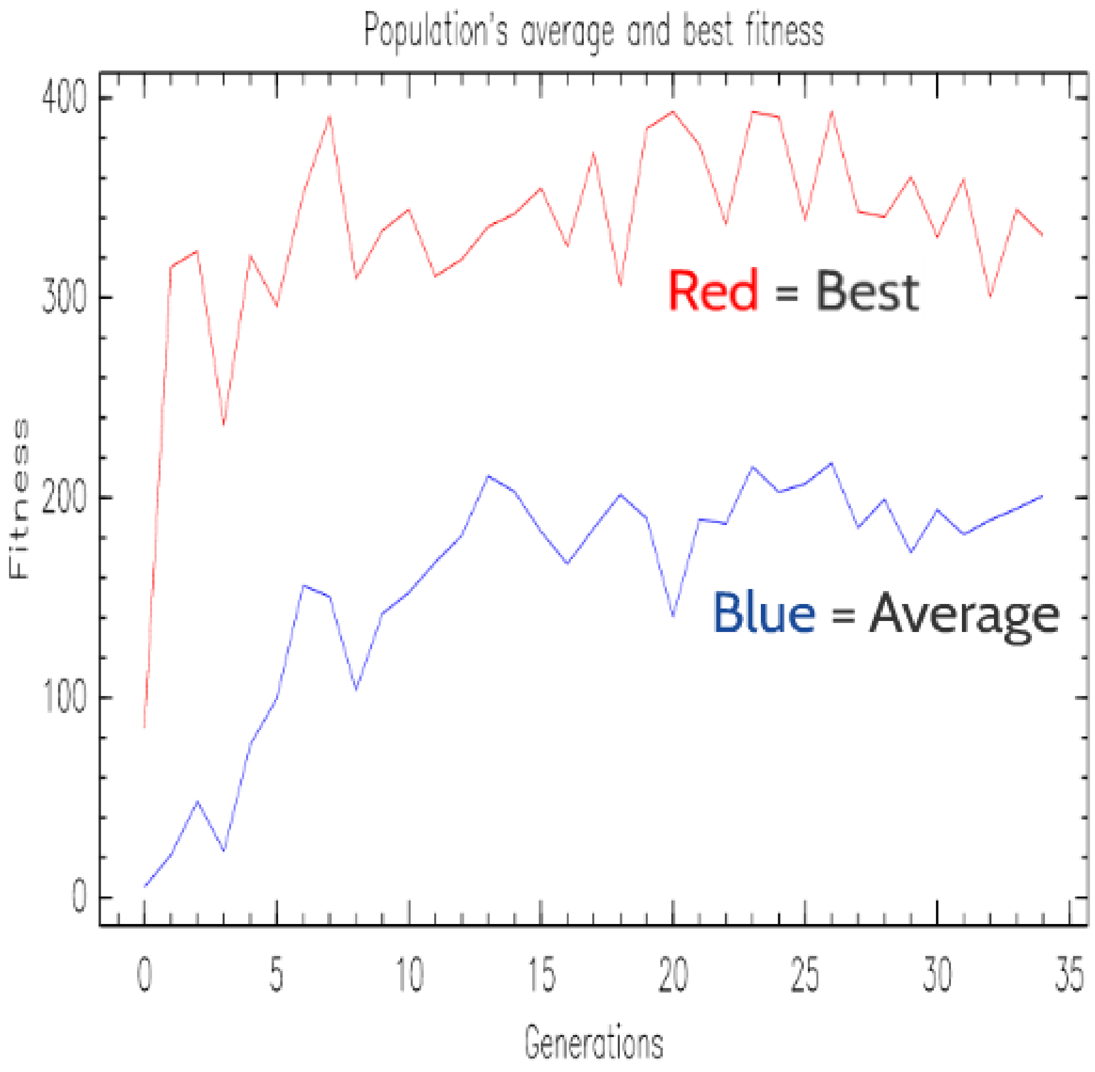


Figure 4: The average and best fitness of each population, measured by predictive ability at time t+2

One problem for these two experiments is that the currency prices do not react to our trades, and the method of extrapolation we used keeps prices steady for timesteps where we don't have a data point. In both cases, the neural network converged on simple structures with few hidden nodes. This was because they could exploit the most consistently profitable trading circle without adding much complexity.

## 11 Discussion

Our implementation is an excellent proof of concept. However, our model is somewhat limited and remains inhibited from trading in the actual market. Although we were able to accumulate a massive number of data points, they were still insufficient for developing an actual trading system, and there were significant difficulties involved in obtaining more data at a useful level of granularity. Additionally, currency trades in real life are executed with varying degrees of "bargaining" on different exchanges. Buyers will provide a maximum bid for some amount of currency and sellers will provide a default asking price for some amount of currency. In a dynamic market, this is what produces profitable arbitrage disparities.

Furthermore, our data was neither impactful nor dynamic in the sense that our trades did not affect the prices of the currencies upon execution. This means that our NEAT traders often evolved to exploit one particular currency cycle that never became unprofitable. We made attempts to fix this by penalizing the trader for making the same trade multiple times in a row but this was not especially effective. The traders just evolved to change the trade once in a while and continue to exploit the same disparity. It would have been better to mimic reality: each time a currency cycle is traded on, it becomes unprofitable.

## 12 Further Study

Currency arbitrage is by no means the only algorithmic trading strategy that has been successfully implemented; there are thousands of other strategies that exist. Mean reversion and gap trading are two such popular strategies. Mean reversion hypothesizes that asset prices revert about a mean value. Profits are generated when the price diverges away from and converges towards the mean. This strategy essentially trades the volatility or fluctuations in price. One example of this is mean reverting index arbitrage, where two similar indices that should theoretically mirror each other's performance are selected. When the two indices diverge, a long trade is placed on the lower valued index (anticipating an increase in value), and a short trade is placed on the higher valued index (anticipating a decrease in value), as convergence is expected.

Gap trading seeks to take advantage of "gaps" or unfilled intervals. Gaps are typically created between the end of trading on one day, and the beginning of trading on the next day. If the highest price of an asset is lower than the lowest price on the following day, or if the lowest price on one day is higher than the highest price on the following day, then a gap has been created. As a simple example, if the price of an asset closes at 100 dollars on one day, but opens at 90 dollars the following day, a long trade can be placed on the prediction that the asset price will rise in value and "close the upgap." Similarly, if the price of the asset were to open at 110 dollars the following day, a short trade could be placed on the expectation that the asset will decrease in value to "close the downgap."

## 13 Conclusion

We used NEAT to produce a currency trader that would detect the most profitable currency cycle in a set of data. Within ten to fifteen generations, our NEAT currency traders were consistently able to detect and exploit the most profitable cycle in the data set. However, the most profitable cycle in the data set persisted over the course of the experiment and hence did not complexify very well. Introducing trading costs or reactive currency prices might have addressed this issue and made them more complex. In general, our experiment's shortcomings had more to do with the simplicity of our model than with NEAT. NEAT reacted well to the more complex predictive model, and in preliminary experiments where we added reactive prices and trading penalties (transaction costs).

## References

[1] Beverly Goodman. "Tapping the Speed on High - Speed Trading". In: *Barron's* (February 25, 2012).

[2] Hean-Lee Poh Jingtao Yao and Teo Jasic. "Foreign Exchange Rates Forecasting with Neural Networks". In: *Smartquant* (1996).

[3] Vasco Leemans and Michael Dempster. "An Automated FX Trading System Using Adaptive Reinforcement Learning". In: *Centre for Financial Research, Judge Institute of Management, University of Cambridge* (December 24, 2004).

[4] Richard Martin. "Wall Street's Quest to Process Data at the Speed of Light". In: *InformationWeek* (April 20, 2007).

[5] Gregory McLeod. "Forex Market Size: A Trader's Advantage". In: *DailyFX* (April 14, 2014).

[6] Eric Simon. "Forecasting Foreign Exchange Rates With Neural Networks". In: *Computer Science Institute, University of Neuchatel* (February 15, 2002).