



University of Pennsylvania  
ScholarlyCommons

---

Wharton Research Scholars

Wharton School

---

5-12-2008

# Genetic Algorithms and Investment Strategy Development

Michael Dworkis  
*University of Pennsylvania*

Darien Huang  
*University of Pennsylvania*

Follow this and additional works at: [http://repository.upenn.edu/wharton\\_research\\_scholars](http://repository.upenn.edu/wharton_research_scholars)

 Part of the [Business Commons](#)

---

Dworkis, Michael and Huang, Darien, "Genetic Algorithms and Investment Strategy Development" (2008). *Wharton Research Scholars*. 87.

[http://repository.upenn.edu/wharton\\_research\\_scholars/87](http://repository.upenn.edu/wharton_research_scholars/87)

This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/wharton\\_research\\_scholars/87](http://repository.upenn.edu/wharton_research_scholars/87)

For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

# Genetic Algorithms and Investment Strategy Development

## **Abstract**

The aim of this paper is to investigate the use of genetic algorithms in investment strategy development. This work follows and supports Franklin Allen and Risto Karjalainen's previous work<sup>1</sup> in the field, as well adding new insight into further applications of the methodology. The paper first examines the capabilities of the algorithm designed in Allen and Karjalainen's work by using human-developed (rather than market-historical) datasets to determine whether the algorithm can detect simple signals; the results show that the algorithm is quite capable of such basic tasks. Next, the S&P 500 test performed in Allen and Karjalainen's original work was confirmed. Then, experiments were conducted in emerging equity markets, as well as commodities markets with a range of fundamental as well as technical indicators. The results generally show no significant positive excess returns above a buy-and-hold strategy; speculations for possible reasons are discussed. In addition, suggestions for future research endeavors are presented

## **Keywords**

genetic, algorithms, investment strategy, development

## **Disciplines**

Business

# Genetic Algorithms and Investment Strategy Development

Michael Dworkis, Darien Huang

Faculty Mentor: Dr. Franklin Allen

May 12, 2008

Wharton Research Scholars  
The Wharton School, University of Pennsylvania

**Abstract:** The aim of this paper is to investigate the use of genetic algorithms in investment strategy development. This work follows and supports Franklin Allen and Risto Karjalainen's previous work<sup>1</sup> in the field, as well adding new insight into further applications of the methodology. The paper first examines the capabilities of the algorithm designed in Allen and Karjalainen's work by using human-developed (rather than market-historical) datasets to determine whether the algorithm can detect simple signals; the results show that the algorithm is quite capable of such basic tasks. Next, the S&P 500 test performed in Allen and Karjalainen's original work was confirmed. Then, experiments were conducted in emerging equity markets, as well as commodities markets with a range of fundamental as well as technical indicators. The results generally show no significant positive excess returns above a buy-and-hold strategy; speculations for possible reasons are discussed. In addition, suggestions for future research endeavors are presented.

---

We would like to thank Dr. Franklin Allen and Dr. Risto Karjalainen for their immense help and guidance throughout our research process. We thank Dr. Martin Asher and the Wharton Research Scholars program for helping to make this research possible. All the views expressed in this paper are solely attributed to the authors and do not in any way represent any authors cited or any employers of the authors.

## Outline

- I. Motivation & Introduction to GA (p.3)**
  - Experiment Motivation
  - Investment analysis process
  - Limitations in traditional investment strategy development process
  - Introduction to Genetic Algorithms and implementation in investment strategy development
  - Potential strengths and weaknesses of Genetic Algorithms for investment strategy development
  - Investing: Art or Science?
  - Programming language choice
  - Previous work on genetic algorithm use in finance
  
- II. Perfect foresight experiment (9)**
  - Experiment Motivation
  - Input selection
  - Implementation
  - Results & Discussion
  
- III. S&P 500 experiment (13)**
  - Experiment Motivation
  - Input selection
  - Implementation
  - Results & Discussion
  
- IV. Emerging Markets Experiment: China A Shares Market (15)**
  - Experiment Motivation
  - Input selection
  - Implementation
  - Results & Discussion
  
- V. Gasoline Prediction Experiment (19)**
  - Experiment Motivation
  - Input selection
  - Implementation
  - Results & Discussion
  
- VI. Conclusions (24)**
  - Experiment result discussion
  - Suggested Future Experiments and Speculations

## I. Motivation & Introduction to Genetic Algorithms

**Experiment Motivation:** Market participants are constantly searching for new investment strategies to earn excess returns (defined as returns above a benchmark measure) in financial markets. Investment strategies can be based on models as simple as buying stocks with low price/earnings ratios, or as complex as trading a levered derivatives portfolio based on the historical correlations between a portfolio of fixed income securities, while dynamically hedging. Strategies proven to yield excess returns can be exploited in the market to earn money. The development of new successful investment strategies, or the improvement of methodologies to produce new successful investment strategies can be a profitable business venture.

**How are investment strategies developed?** The answer can vary across asset classes. In the case of stocks and corporate bonds, traditional fundamental analysis entails analyzing the corporation, the quality of the assets, and the specifics of the securities issued<sup>2</sup>. Such analysis is usually carried out through the study of traditional quantitative indicators emphasizing value (various price/earnings metrics), financial stability (liquidity ratios), and qualitative opinions such as management depth and expertise and market dominance. The goal of such analysis is to determine what is the real, intrinsic value of a security, and to then compare that value to the price being offered in the market. Traditional analysis tools include discounted cash flow modeling, multiples analysis, and comparable transactions analysis. When a discrepancy between the intrinsic value and market price exists, there is a chance to profit by buying securities believed to be undervalued and selling securities believed to be overvalued. Other types of analysis tools include technical analysis, in which an analyst studies variables such as current price, historical price, volume, and more to predict future prices, and invests accordingly.

Investment strategies can be based on qualitative factors such as investing in “green” companies with a superior focus on corporate social relations and alternative energy, or on quantitative factors such as trading futures based on a belief that the relationship between the S&P500 and the Japanese Nikkei index is mean-reverting over a 6-month horizon. Often investment strategies are based on both qualitative and quantitative factors, specializing in a specific market niche (e.g. Asian small-cap

---

<sup>1</sup> Allen, Franklin, Risto Karjalainen. "Using genetic algorithms to find technical trading rules." Journal of Financial Economics 51(1999): 245-271.

<sup>2</sup> Whitman, Martin J., and Martin Shubik. The Aggressive Conservative Investor. Wiley, 1979.

industrial) and then analyzing specific securities based on both quantitative financial and qualitative managerial variables.

New investment strategies are generally developed by a combination of innovative hypothesizing and empirical research. Generally a human uses various financial analysis tools to discover a repeated discrepancy between intrinsic value and market price, and then formulates an investment strategy to take advantage of this perceived discrepancy. The search for new investment strategies is carried out by thousands of finance professionals around the world, and has the potential to yield huge profits if found. For this reason, it is worth thinking not just about individual potential strategies, but about refining the process through which new strategies are developed.

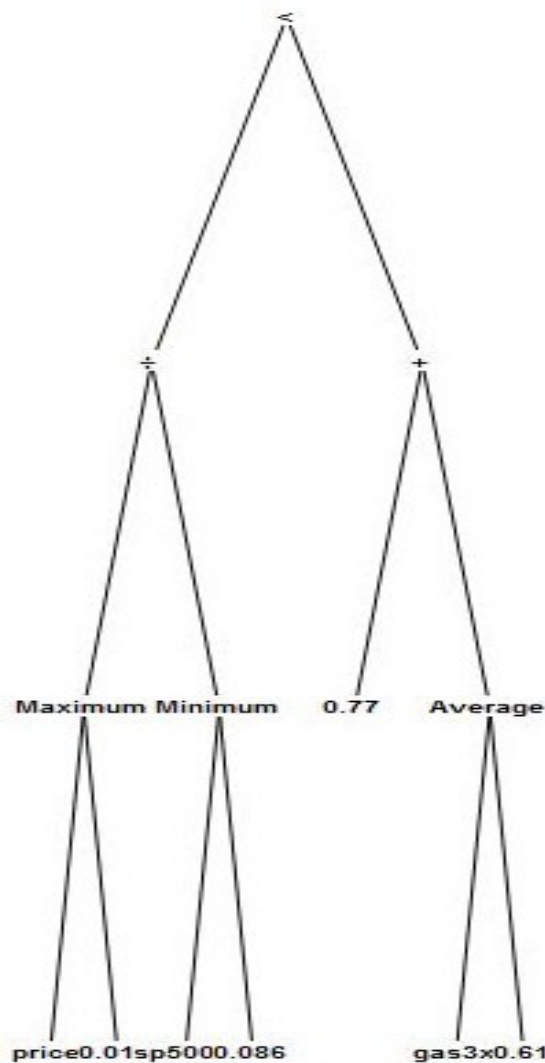
**Limitations in traditional investment strategy development process:** Two primary bottlenecks exist in the process of humans developing new investment strategies. Firstly, human thought processes must choose what variables are significant and worth spending time to analyze. This process can be biased both by traditional investing philosophy (e.g. that low P/E ratios often present a better value than high P/E ratios) and by the lack of human conceptualization of potential relationships among variables. Secondly, there exists a bottleneck in human ability to process and analyze large data sets. An analyst might be interested in potentially investing in thousands of publicly listed companies, but would never have time to thoroughly analyze all of their public statements.

In an attempt to alleviate both of these potential bottlenecks, this paper explores the use of a genetic algorithm to optimize the analysis process in the development of investment strategies. Genetic algorithms were first recognized as a promising tool for financial research because of their previous success in solving various NP-hard and complex problems in engineering.

**Introduction to Genetic Algorithms and implementation in investment strategy development:** Genetic algorithms are a type of evolutionary algorithm, which refers to a group of search heuristics inspired by evolutionary processes found in biology. These evolutionary search heuristics attempt to find optimal solutions to problems by creating solution populations which are then evolved over time according to fitness criteria pertaining to the specific problem.

The genetic algorithm used in this paper is implemented in the Mathematica environment, using the model developed in the original Allen and Karjalainen paper. Specifically, solution candidates are represented in Mathematica as nested tree functions, which return a signal to buy or sell. In the context

of the following experiments, all of the solution candidates return Boolean functions, because the signal is either to buy or not to buy. However, other templates could be structured to allow for buying, short-selling, or neutral positions, or trading multiple assets within a strategy. Each node in the solution tree can be a function, variable, or value. Solution candidates are initially randomly generated according to a pre-defined template, which contains the available potential functions and variables. Values are generated through a randomizing function. Basic functions included in all experiments with the algorithm include arithmetic operators, absolute difference, and a moving average function. More functions can and should be developed to suit individual models.



**Sample trading rule. (Inputs: Price, S&P500 close, 3 month gasoline futures contract)**

First, a population of individual solution candidates is generated according to the solution template, and is then evaluated by the fitness function. In the case of investment strategies, the fitness function measures excess return above a buy and hold strategy. The individuals are then ranked according to their fitness, and then randomly mutated and recombined with each other (similar to the biological process with recombination and mutation of DNA during the reproduction process), with a bias towards the most fit individuals passing on their traits. Next, the best rule from each generation is tested against a range of data called the selection period. If the rule applied to the selection period outperforms the previous rules applied to the selection period, it is then saved as the best rule developed so far. The best rule is then applied to the out-sample test period, and its fitness is again evaluated.

This process of evaluating the fitness of the current generation and then creating a new generation of solution candidates based on the traits of the parent generation is then continued until a termination criterion is reached. Generally in experiments with investment strategy development, the termination criteria include reaching a maximum number of generations, or reaching a plateau of fitness where no progress is being made across subsequent generations.

**Potential strengths and weaknesses of Genetic Algorithms for investment strategy development:**

Genetic algorithms help address the two aforementioned bottlenecks in the investment strategy development process, but also come with their own limitations. Firstly, there is still an element of human choice with respect to even hypothesizing which variables should be passed on to the genetic algorithm-based process. If a human cannot conceive of or find a quantifiable variable to analyze, there is no way for it to be fed into the genetic algorithm-based model. With many modeling techniques, the traditional adage of “junk in, junk out” applies, implying that the results of a model can only be as quality as the data fed into it. With genetic algorithms, because of their unique ability to evolve a solution to a problem, “junk in, junk out” is not entirely the case. Even if just some of the input data is relevant to the problem at hand, the genetic algorithm should be able to filter out the useless data from that with some degree of predictive merit, and develop an investment strategy based solely on the data with predictive merit, essentially ignoring the junk data, finding only the diamond in the rough.

The human bottleneck of choosing where to spend time searching for potentially profitable relationships can be greatly aided with a genetic algorithm. Because of its sheer computational power advantage over a human, it can look at vastly more potential relationships than a human would have time to analyze, and the cost of doing so is minimal in terms of computational power and memory.



Further, genetic algorithms approach problems without previously existing biases. A genetic algorithm would be just as likely to initially consider strategies investing in equities with high price/book values as those with low price/book values, as the algorithm is uninhibited by “conventional wisdom”. This can add significant value when searching for new relationships to develop investment strategies yielding above-average returns.

**Investing: Art or Science?** By solely relying on any quantitative, “black box” investment methodology, one is making to some degree an assumption that investing is a quantifiable science. Some investors, including legends like Warren Buffet<sup>3</sup> would likely argue that investing is more of an art, in which financial statement analysis, innovative thinking and “gut feeling” are the primary components in success, rather than a science, in which companies, securities, and markets can be quantified, analyzed, and successfully predicted. With respect to a genetic algorithm-generated investment strategy, one must analyze whether the strategy makes any economic or intuitive sense, or if it is just a semi-random combination of variables that perfectly predicted the past, but offer limited insight into predicting the future of the markets. However, there does not have to be complete disconnect between the art and science approaches to investing. Algorithm-based models can be used as screens, to weed out potential investment opportunities which might yield an above average expected profit if analyzed using sound fundamental analysis techniques.

**Programming language choice:** The genetic algorithm was implemented in Mathematica, using the model developed in 1999 by Franklin Allen and Risto Karjalainen. Mathematica is a popular mathematical symbolic manipulation software which is commonly used in science, engineering, as well as in finance; however, its use in finance is mainly limited to academic research. Practitioners in the financial industry (with a strong computing background) generally prefer other packages (such as MATLAB) and programming languages (such as C++), because of their computational efficiency. Mathematica is comparatively slow to run, and also has a steeper learning curve than other comparable packages. However, it is worth mentioning that – especially for finance professionals who may not necessarily have the background in computer science – Mathematica is much easier to grasp than programming in C++. For someone with prior exposure to modern programming languages such as Java and Python, MATLAB programming is relatively easy to pick up; Mathematica, however, generally takes

---

<sup>3</sup> Warren Buffett, multiple letters to Berkshire Hathaway shareholders and various interviews.

longer to master. For computational speed, directly programming in C++ as opposed to using a software package is generally the method of choice.

For this paper, Mathematica was used because it was used to build the original genetic algorithm model in Allen, Karjalainen. Due to the long calculation times, this method is not feasible for trading on an intra-day level (tick-by-tick, within each trading day), since some of the calculations take several days of computing time. This problem is exacerbated when many indicator variables are considered). Genetic algorithms definitely may have use in intra-day trading, and future applications could consider writing such programs in C++ for computational speed. While it has its flaws, Mathematica is still a widely used software package with very detailed user documentation and a large community of users.

**Previous work on genetic algorithms in finance:** Allen & Karjalainen's 1999 work and ensuing not yet published work is the basis for this paper and several of its experiments. The 1999 paper found that: "After transaction costs, the rules [found by the genetic algorithm] do not earn consistent excess returns over a simple buy-and-hold strategy in the out-of-sample test periods." The paper went on to suggest several topics for future research, including applying the genetic algorithm to futures markets, and expanding inputs to the model to include fundamental variables.

Fernández-Rodríguez, González-Martel, and Sosvilla-Rivero (2005)<sup>4</sup> found that using genetic algorithms to optimize moving average trading rules, excess profits above a buy-and-hold strategy were achieved for the General Index of the Madrid Stock Market.

Several papers have been published examining the potential for genetic algorithms to add value in project finance applications. In particular, constrained optimization models solved using genetic algorithms have yielded beneficial results in building portfolio management<sup>5</sup> maintenance and

---

<sup>4</sup> Fernández-Rodríguez, Fernando, Christian González-Martel, Simón Sosvilla-Rivero. "Optimization of technical rules by genetic algorithms: evidence from the Madrid Stock Market." Applied Financial Economics 15(2005): 773–775.

<sup>5</sup> Tong, Tam and Chan. "Genetic Algorithm Optimization in Building Portfolio Management." Construction Management and Economics 19, (2001): 601-609.

replacement planning, semiconductor capital expenditure budgeting<sup>6</sup>, and public infrastructure investment<sup>7</sup>.

In addition to published research, it is believed that numerous hedge funds and private investment vehicles are using genetic algorithms, neural networks, and other methods of evolutionary computation as parts of their quantitative trading strategies. Algorithmic trading by hedge funds and investment banks now accounts for a significant amount of market volume.

### I. Perfect Foresight Experiment; Abilities of Genetic Algorithm

**Experiment Motivation:** To start off demonstrating what the genetic algorithm is capable of, “practice” data was used. This means that the data has been “manipulated” with certain profitable trends to see if the algorithm can identify the trends and formulate profitable trading strategies under these environments. The experiment is called “Perfect Foresight” – the algorithm is given the ability to see one day ahead by including a second variable called *t<sub>mr</sub>* (for “tomorrow”) which is the closing price of the next day. If this experiment fails, the model must be adjusted before going forward with other, more sophisticated experiments. Results indicate that the algorithm performs quite well.

**Input Selection & Data:** 2 variables were chosen in addition to a generic (non-varying) interest rate set arbitrarily low and constant for the entire duration of the data. While it may seem extraneous to explain the variables (they are indeed, quite simple in this basic experiment), the paper will adopt this convention of explaining the set up of each experiment for all the more sophisticated experiments to follow.

1. *Price:* 1 month gasoline futures prices are used as a realistic proxy, although any liquid asset could have been used.
2. *Tomorrow’s Price:* This is the price at the close of the next day, given to the algorithm today. The objective is for the algorithm to “learn” that tomorrow’s price is a profitable indicator.

---

<sup>6</sup> Wang, K.J., S.H Lin. "Capacity Expansion and Allocation for a Semiconductor Testing Facility under Constrained Budget." Production Planning & Control 13(2002): 429-437.

<sup>7</sup> Hsieh, Ting-ya, Hsin-Lung Yu. "Genetic Algorithm for Optimization of Infrastructure Investment Under Time-Resource Constraints." Computer-Aided Civil and Infrastructure Engineering 19(2004): 203-212.

3. *Rate*: This is a proxy interest rate, to calculate returns to “staying out of stock market” (and being in cash market). While this is not necessary here, this is used because the algorithm expects to take in cash rates, so for simplicity a 1% constant interest rate value is used. Note that this does not mean rate is in the set of consideration variables, so there needs to be no worry about cases where the stock tomorrow closes less than 1% higher than today but still higher.

**Implementation:** For this experiment, a population size of 100 was used for 10 generations. This was chosen because of computational efficiency, and it would be more interesting if the genetic algorithm could find the signal more quickly. Fitness is defined as cumulative excess return over buy and hold strategy. Transaction costs were set to 0% (since actual performance is not what is of concern here). The training period was 5 years and the out-sample was run for approximately 10 years.

**Results & Discussion:** Recall from the earlier discussion that one drawback of the algorithm is that it is not possible to “short-sell” securities directly in the algorithm (although this is addressed somewhat, through a manual trick using spreadsheet programming; this technique is reserved for more sophisticated experiments). Therefore, given this feature, the most profitable trading strategy should be the simple signal:

```
tmr > Price           //Long stock if tomorrow's closing price greater than today's, hold cash otherwise.
```

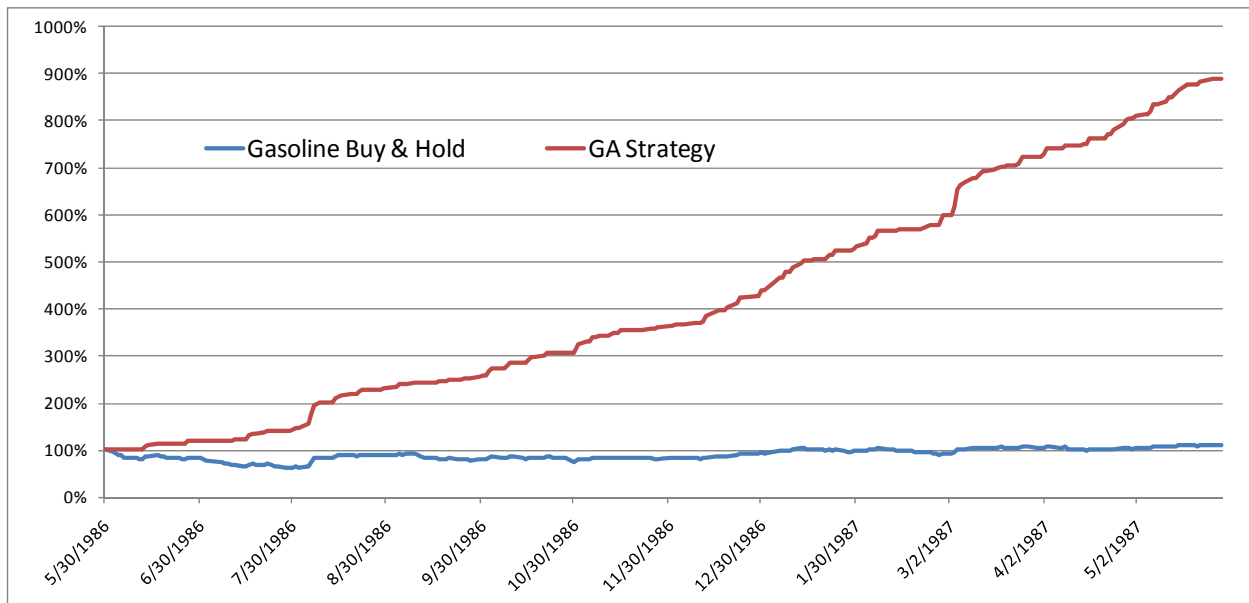
Any other signal captures noise and will not be as profitable as this. Using population size 100, for 10 generations, the algorithm finds:

**Best solution candidate:**

```
Price > tmr //Long stock if tomorrow's price > today's price. Most trials found this to be the best solution candidate; those that did not found very similar strategies. Given larger trial sizes (500/50, for example), it is likely that all best solutions will have this form.
```

Trial	Fitness	Best trading rule	Generation
1	3.8953	price < tnr	1 / 10
2	3.8953	Maximum[price,1] < tnr && 0.903147 < tnr	3 / 10
3	3.8953	Maximum[price,1] < tnr	1 / 10
4	3.8953	price < Minimum[tnr, Distance[0.341211, 0.993895]] && tnr ≥ 0.0524023	11 / 10
5	3.8953	price < tnr	1 / 10
6	3.8953	price < tnr	1 / 10
7	3.8953	price < Minimum[tnr, 0.969038]	2 / 10
8	3.8952	tnr ≥ Maximum[tnr, 1.03044]	1 / 10
9	3.8953	Distance[-tnr + Average[tnr, 0.543804], Maximum[price,1]] < tnr	6 / 10
10	3.8953	Maximum[price,1] < tnr	3 / 10

This signal will recommend a long position if tomorrow's stock price is greater than today's. This is exactly what the algorithm should have found! This was found in the relatively small experiment size of 100/10, with great speed (computing time was 23 seconds). While this seems quite obvious (anyone would know to buy stocks that are sure to increase in price the next day), the ability of the program to find it is a minimum necessary condition to show that algorithm might potentially be capable of finding excess returns in real market environments.



This strategy is quite successful in generating excess returns - as would be expected. The excess returns here represent the maximum *possible* excess returns over the time period in consideration, in the absence of short-selling. Once again, this result only demonstrates that the genetic algorithm is able to quickly pick up simple patterns and there is potential for it to find excess returns in real market conditions.

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1986-1992	10	+2.2341	10	1947	+0.017969	0.015757	1805	-0.018537	0.017537	+0.036506	(+45.300)	10
1991-1997	10	+2.4557	10	1298	+0.020069	0.017112	1199	-0.020454	0.018987	+0.040523	(+37.300)	10
1996-2002	10	+2.5140	10	651	+0.020860	0.017420	599	-0.020961	0.019097	+0.041821	(+26.600)	10

Note: (The dates are just used as placeholders, since the template from another experiment is used) This is the out-sample excess returns for a representative candidate among the many run-throughs.

The table above will be presented throughout the paper, so it is worthwhile to discuss what various columns mean. In-sample shows the years that were used as in-sample to generate strategy candidates. K is the number of profitable strategies found in-sample, Excess is the average annual log return, and K+ is the number of profitable strategies in the out-sample period (K+ ≤ K always). 'b' represents in-sample performance, and 's' represents out-sample performance. So, r<sub>b</sub> is in-sample daily log returns, and r<sub>s</sub> is out-sample daily log returns. N<sub>b</sub> is the number of days in-sample, N<sub>s</sub> is number of days out-sample. r<sub>b</sub>-r<sub>s</sub> is an indication of whether the strategy actually generates returns out-sample in excess of in-sample, and the t-statistics given are at the 95% confidence level.

As can be seen here, "perfect foresight" significantly outperforms out-sample – these strategies are profitable. However, this was only a demonstration of the algorithm's capabilities. Realistic stock market environments will be considered next.

## II. S&P 500 Experiment

**Experiment Motivation:** The goal for this experiment was to try to obtain the results in Allen, Karjalainen (1999) based on their experiments with the S&P500. The experiment was undertaken to make sure that the algorithm was working properly, and to learn how to set up experiments for later use. Essentially, the experiment hoped to achieve the same no-excess returns above buy and hold result. All the variables used are publicly available.

**Input Selection & Data:** 2 variables were chosen: S&P 500 closing price and interest rates. The experiment was set up exactly as in Allen, Karjalainen (1999).

1. *S&P500*: This is the daily closing price for the S&P500 index, normalized by dividing by a 250 day moving average.
2. *Interest Rates*: 1-month Treasury bill yields at first until 1992, then rolls over to Eurodollar deposit rates thereafter. Note that rates here are not included as an indicator, but only as the returns for holding cash. This means that the model, with the only indicator being price, would never develop a strategy which stays out of stock market if returns to cash in the treasuries market are high. Rather, the strategy developed will be based solely on functions derived from past prices, and rates will only measure returns while holding cash.

**Experiment set-up:** For this experiment a population of size 500 and 50 generations was used. Fitness was defined as cumulative excess return over buy and hold strategy. Transaction costs of 0.1%, 0.25%, and 0.5%. The training period was calculation deserves discussion. The data started in 1954, and training periods began every 5 years, until 1979. What this means is the first training set consisted of data from 1954 to end of 1958, and 1959 to 1963 is another training set, and so on. The year following training is the selection period, where profitable strategies from the training period are kept if and only if they perform even better in the selection period, otherwise they are discarded. The out-sample evaluation period was all the years from one year after the selection period (or two years after the end of the training period), until 2002. Each training set essentially leads to a new experiment; this is valuable to avoid over-fitting the algorithm to events that are specific to a certain period of time (for example, a sudden market crash, etc).

**Results & Discussion:** It can be seen from the following table that under low transaction cost environment, the excess returns are essentially 0 (mean approximately 0, with slight variation), while excess returns become increasingly negative on average as transaction costs increase (which is expected). This is in line with the results of Allen, Karjalainen as well as generally accepted theory: the S&P500 is one of the most efficient markets, and any strategy that considers only technical price data (without even consideration for other tools of technical analysis, such as volume) cannot make returns in excess of a buy-and-hold strategy, according to even the weakest forms of efficient market hypothesis.

**Panel A. A summary of the out-sample test results for transaction cost = 0.1%**

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1959-1965	1	+0.0226	1	7845	+0.000547	0.008862	1573	-0.001293	0.013132	+0.001839	(+6.840)	1
1964-1970	8	-0.0176	2	5085	+0.000636	0.009030	3100	-0.000102	0.012937	+0.000737	(+3.160)	5
1969-1975	10	-0.0506	0	2531	+0.001194	0.010863	4392	+0.000007	0.010895	+0.001223	(+3.170)	8
1974-1980	1	-0.0612	0	3365	+0.000713	0.009527	2295	-0.000230	0.012059	+0.000943	(+3.280)	1
1979-1985	4	+0.0094	4	3302	+0.000465	0.008852	1094	-0.000098	0.016172	+0.000563	(+1.450)	0

**Panel B. A summary of the out-sample test results for transaction cost = 0.25%**

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1964-1970	4	-0.0063	1	7312	+0.000311	0.009332	873	+0.000044	0.014758	+0.000268	(+0.711)	1
1969-1975	10	-0.0405	0	105	+0.003865	0.023872	6818	+0.000327	0.010113	+0.003540	(+1.800)	2
1979-1985	2	+0.0012	2	3269	+0.000452	0.008864	1127	-0.000046	0.016023	+0.000498	(+1.290)	0

**Panel C. A summary of the out-sample test results for transaction cost = 0.5%**

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1964-1970	2	-0.0272	0	7444	+0.000251	0.009168	741	+0.000523	0.016648	-0.000273	(-0.702)	0
1969-1975	10	-0.0410	0	212	+0.014626	0.024691	6711	+0.000319	0.010165	+0.014313	(+3.480)	4

Primarily, this experiment was used to gain a better understanding of the algorithm and how it works. Having confirmed the results of Allen, Karjalainen (1999), the next test involves using the algorithm to search for trading strategies on a new set of data.



### III. Emerging Markets Experiment: China A Shares Market

**Experiment Motivation:** Perhaps the S&P500 is simply too efficient to allow technical strategies to earn excess returns. If that is the case, would there be gains from applying the algorithm to potentially less-efficient emerging markets? The China A Shares (open to purchase by Chinese investors and selected qualified foreign institutions) market is interesting and relevant for several reasons. First and foremost, up until December 2007, China was experiencing an enormous bull market. At the same time, institutional rules for Chinese financial markets were favorable for the algorithm: short-selling is not permitted (as currently set up, the algorithm cannot identify short-sell opportunities), and shares cannot be sold on the same date purchased (the algorithm performs calculations on an inter-day basis, and does not generate intraday trading strategies). Finally, the presence of a large mass of retail investors potentially increases the likelihood of potential opportunities, compared against the more sophisticated hedge funds, institutional investors, etc that are dominant players in more developed markets.

**Input Selection & Data:** This experiment was run analogous to the S&P500 experiment above. The price data is China Shanghai A-Composite Index closing prices normalized by dividing by the 250 day moving-average, and the interest rate data was government mandated Chinese Central Bank Overnight rate.

**Experiment set-up:** The experiment was set up using a population size of 500, and 50 generations. Fitness is defined as cumulative excess return over buy and hold strategy. Transaction costs were 0.1%, 0.25%, and 0.5%. The training period was 1998 to 2002, selection 2003 to 2004, out-sample 2005 to 2007. Note that recently the Chinese market has fallen quite dramatically, which would lower the returns to buy-and-hold strategy (and make the algorithm strategy look more favorable) – these data points were not incorporated in the experiment.

## Results & Discussion:

**Panel A. A summary of the out-sample test results for transaction cost = 0.1%**

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1998-2004	6	-0.2286	2	133	+0.001471	0.013919	388	+0.001091	0.012940	+0.000549	(+0.580)	2

**Panel B. A summary of the out-sample test results for transaction cost = 0.25%**

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1998-2004	7	-0.3070	0	46	+0.001662	0.014726	475	+0.001341	0.012951	+0.000542	(+0.531)	1

**Panel C. A summary of the out-sample test results for transaction cost = 0.5%**

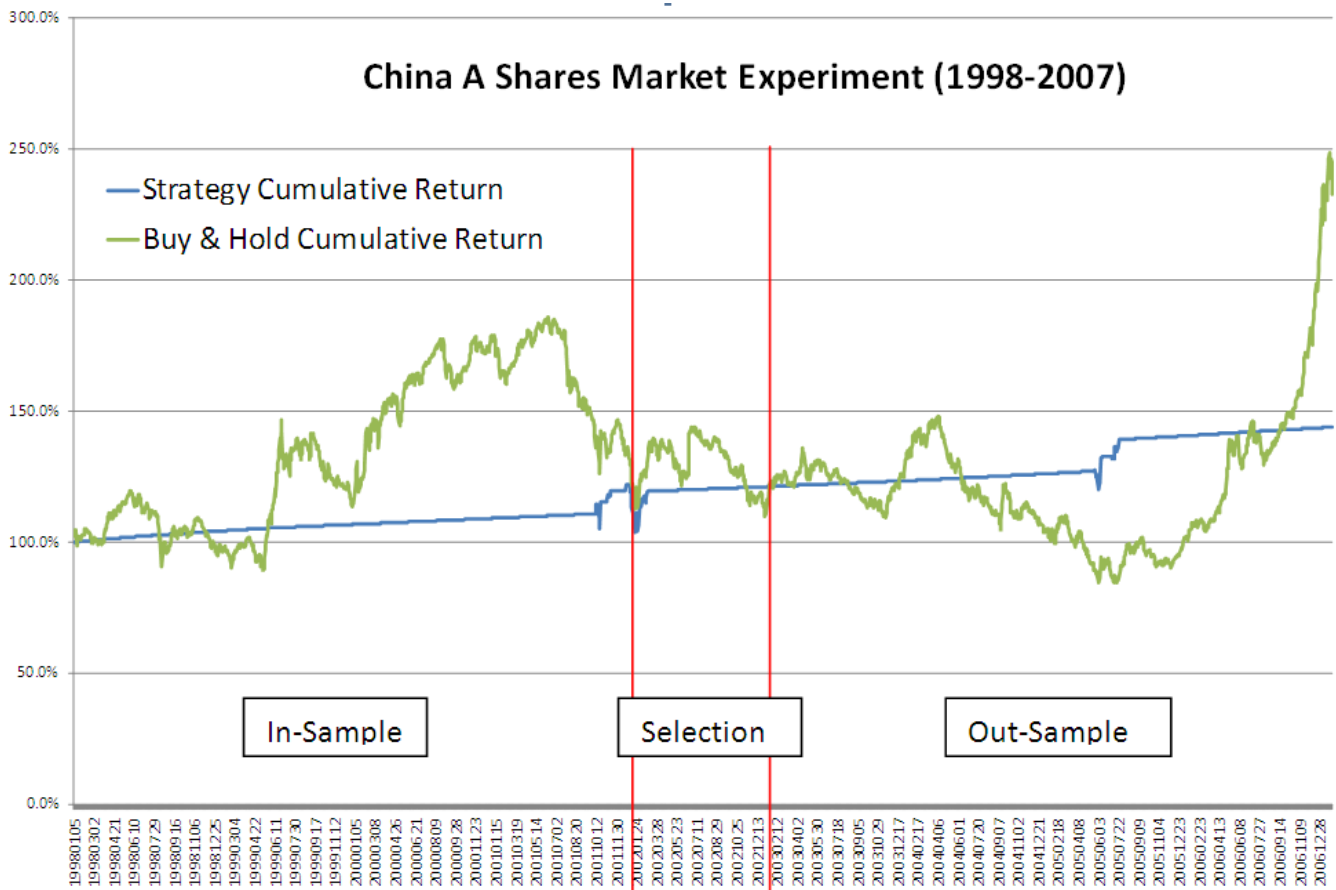
In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1998-2004	10	-0.2876	1	70	+0.007417	0.021542	451	+0.001465	0.013510	+0.005917	(+1.230)	3

- C.1 In-sample years = 1998-2004, out-sample years = 2005-2006

Trial	Fitness	Best trading rule	Generation
1	0.1136	price < 0.744566	1 / 5
2	0.1136	0.62411 ≥ Average[price, -0.00954731 + Minimum[price,1]]    price < 0.676145	1 / 5
3	0.1136	price Distance[Minimum[price, Average[price,1]], 0.341676] < $\frac{0.307822 \text{ price}}{\text{Minimum[price,1]}}$	3 / 5
4	0.1136	price < 0.814553	1 / 5
5	0.1136	0.80217 ≥ Average[price, Distance[0.375934, 0.67079]]	1 / 5
6	0.1136	price $\left( 0.0444041 + \text{Distance} \left[ \frac{0.735382}{\text{price}}, 0.530641 \right] - \right.$ $\left. \text{Minimum[price,1]} \right) \geq -0.916198 + \text{Average[price,1]}$	1 / 5
7	0.0163	Minimum[price,1] ≥ 0.110265 && Maximum[price,1] ≥ 0.912731 && 0.566016 < Distance[0.77047, price] (price + Minimum[price,1])	1 / 5
8	0.0014	price ≥ Distance[0.262359, price] Minimum[price, price]	1 / 5
9	0.1136	Distance[Distance[Minimum[price,1], Maximum[price,1]], 0.8333] ≥ 0.0362738 + price    0.280321 price < 0.0487373	1 / 5
10	0.1136	If[Minimum[price, Distance[0.654178, 0.0855631]] ≥ 0.118579 && 0.759413 ≥ price, True, -0.536284 ≥ price]	1 / 5

The most eye-catching feature of the table on the previous page must be the significant negative excess return, varying from -22% to -30%. Note, however that the T-test of out-sample returns vs. in-sample returns are essentially zero, so the strategy is not necessarily just over-fitting in-sample and failing out-sample. In a volatile market like China, a more appropriate measure of return should indicate some measure of volatility—such as a Sharpe Ratio or Sortino Ratio—which the buy-and-hold benchmark does not account for. If the time series was ended earlier for this experiment, these negative excess returns would be much smaller.

In the development of trading strategies, negative excess returns are not inherently bad. If returns are significantly negative with a high statistical confidence level, then one can profit from taking the inverse of the suggested trading rule. If however, returns are just slightly negative, there is no clear strategy to achieve profit. However, because of China’s institutional constraints preventing short-selling, even an strong excess negative return does not allow for a successful trading strategy.



There are several possible reasons why excess returns were significantly negative.

In the out-sample period, China experienced of the largest bull markets in history– simply buying the stock at the start of out-sample and holding it until 2007 produces annual gains of 40%+. As the A-Shares index climbed to its 2007 peak, it occasionally experienced small dips along the way. Any long-only strategy competing with a buy and hold would need to time those dips exactly right in order to exhibit significant out-sample performance in such a bull market. However, the trading strategies which performed best in the in-sample period often stayed out of the A Shares market and held cash for varying lengths of time during the in-sample. While in “normal” markets this would be considered an essential part of a strategy in terms of risk management, these strategies could be characterized as overly cautious in this massive bull Chinese out-sample bull market . By holding cash for even small periods of time before coming back into the equity market, large portions of the bull run were missed. The above graphed strategy highlights an extreme example of an overly cautious strategy missing a bull market.

While returns against the buy-and-hold are negative as measured on the specific dates used, they are still quite high in and of themselves. In the context of the huge bull market of the out-sample period, the best performing strategy from the run still returned over 30% annualized, which is very strong compared to other international equity market returns over the same period.

Second, once again, only technical price data was considered. Essentially this experiment was the Allen, Karjalainen S&P500 experiment, but done using Chinese market data.

Finally, the time series being considered is very short, seeing as the Chinese market is quite nascent. This means that there is not much time for the algorithm to “learn” during the training period, since it is inevitable to make the training period short enough to have meaningful selection and out-sample evaluation periods. The decision whether to use large samples of data and risk over fitting versus using small samples of data and finding limited predictive value is always a tradeoff, and highlights elements of strategy development that are largely art rather than science.

The Chinese A-Shares experiment suggests that strategies which trade only on past price data are generally not effective in the long-run, even when used in a “less efficient” emerging market.

#### IV. Gasoline Prediction Experiment

**Experiment Motivation:** The goal for this experiment was to develop a model through use of the genetic algorithm to predict gasoline futures prices. Predicting the gasoline futures contract price presents a particularly relevant and interesting challenge: the commodities markets are in the midst of a huge bull run, and energy issues are taking a prominent place on the macroeconomic scene, both in their effect on inflation and national security. With oil prices over \$100/bbl, there is both increasing attention paid to and speculation in the energy futures markets. Although the front month Cushing crude contract is the world's most heavily traded futures contract, the choice of a downstream distilled product (gasoline) rather than crude oil enables the algorithm to develop a model a model incorporating more fundamental data gathered from throughout the energy value chain. This was a new test for the algorithm, on non-financial, industry specific data being used to predict the gasoline futures price. Additionally, all data used in this experiment were gathered from free publicly available sources, notably the US Energy Information Administration<sup>8</sup>, and the International Energy Agency<sup>9</sup>.

**Input Selection & Data:** 14 variables were chosen, from all along the gasoline value chain, as well as a select few macroeconomic, market-wide variables. The data spans from May 1985 through December 2007, which was chosen as the endpoint because of the lagged release of several of the variable indicators even though price data for the gasoline contract was available through today.

1. *Normalized (by 250-day moving average) Front month gasoline future contract price:* This NYMEX traded futures contract is the variable that the algorithm is attempting to predict, but it is also included as an input variable, so that relationships between past prices and future prices (technical analysis) can be discovered. Data from the New York harbor grade contract was used until 2004, when the market switched over to the RBOB (reformulated gasoline blendstock for oxygen blending) specification, which was used for the duration of the data.
2. *3 Month gasoline future contract:* the 3 month contract is included in order to provide the study of the shape of the futures curve. Commodity futures prices are driven by the spot price, cost of storage, risk-free rate, and convenience yield, which takes into account expectations of supply

---

<sup>8</sup> Official website: <http://www.eia.gov>.

<sup>9</sup> Official website: <http://iea.org>.

and demand, including seasonal variance. By including the 3 month contract, the algorithm has the potential to recognize futures curve shapes, such as “contango” and “backwardation”, which are commonly associated with bearish and bullish signals for the market. With the exception of delivery date, the same contract specifications as the front month contract are used.

3. *Front month crude oil contract*: this NYMEX traded contract for light sweet crude oil for Cushing, Oklahoma delivery is the most heavily traded futures contract in the world. Crude oil is the input commodity to be distilled into gasoline, and its price is a key factor in the price of distilled products, including gasoline.
  4. *3 Month crude oil contract*: with the exception of delivery date, this contract is the same specification as the front month crude oil contract, and is included along the same line of reasoning as the 3 month gasoline future, in order to facilitate the development of a futures curve.
  5. *Front month heating oil future contract*: the No. 2 grade NYMEX-traded heating oil future contract was chosen because of its relationship with gasoline as another crude oil end product. Crude oil is distilled into a variety of finished products. The differential between the simultaneous purchase of crude oil futures and the sale of various finished product futures is known as the “crack spread”, and is perceived as an approximation of refinery profits. Chief among finished products are gasoline, heating oil and jet fuel. Heating oil is particularly related to gasoline because of the relative seasonal demand for each product: heating oil is in higher demand in the winter during cold weather, and gasoline is in higher demand in summer, during peak driving season.
  6. *Third month heating oil future contract*: with the exception of delivery date, this contract is the same specification as the front month heating oil contract, and is included along the same line of reasoning as the 3 month gasoline future, in order to facilitate the development of a futures curve.
  7. *Percentage change in crude oil ending stocks*: this data, gathered monthly from the EIA with a one month lag in release time, represents the percentage change in US held crude oil stocks. Since crude is the input for gasoline production, change in available supply would logically have an effect on output gasoline prices. Although this data only refers to US sources and the energy
- Dworkis & Huang, Genetic Algorithms and Investment Strategy Development, 20

futures contracts represent a truly global market, it is still worth including because the US holds approximately 20% of global refining capacity<sup>10</sup>, and the predicted contract variable also refers to a US delivery point.

8. *U.S. percent utilization of refinery operable capacity*: this data, gathered monthly from the EIA with a 3 month lag in release time, might be helpful in determining the relative demand for distilled crude products. Averaging 89.5%, periods of extreme high or low US capacity utilization might be correlated with periods of high or low gasoline demand, or with outages in other global refining capacity.
9. *U.S. Operable Crude Oil Distillation Capacity (Thousands of Barrels per day)*: this data, gathered monthly from the EIA with a 3 month lag in release, offers insight into the total US distillation capacity.
10. *U.S. Percentage Change Finished Motor Gasoline Stock*: gathered monthly from the EIA and released with a 3 month lag. This is a measure of end supply of US gasoline, and would be expected to correlate to some degree with changes in price.
11. *Worldwide Rig Count*: This measure of global on and off-shore rigs, gathered monthly and released quarterly from Baker Hughes Inc., is an indicator of upstream crude supply and widely used throughout the industry.
12. *CPI Inflation*: released monthly by the Bureau of Labor Statistics, this is a non-core (ex food and energy) measure of inflation. Unlike the other variables included, there no clear rationale why CPI inflation would be a leading indicator for gasoline prices. However, one of the advantages of using a genetic algorithm to build a model rather than human intuition is that sometimes new relationships are discovered.
13. *S&P 500 Close*: although there is nearly no correlation between the S&P 500 daily returns and crude oil daily returns over the long run, there might be an inverse relationship in times of extremely high energy prices, representing a market consensus concern about inflated energy prices slowing economic growth.

---

<sup>10</sup> Source: Energy Information Administration, World Crude Oil Distillation Capacity, January 1, 1970 - January 1, 2008

14. *10-year yield*: collected daily, this was used as the return in the cash market when a long position was not taken in the futures market. It was also available for the algorithm to use as an indicator, although logically it should have less predictive ability than other industry-related data.

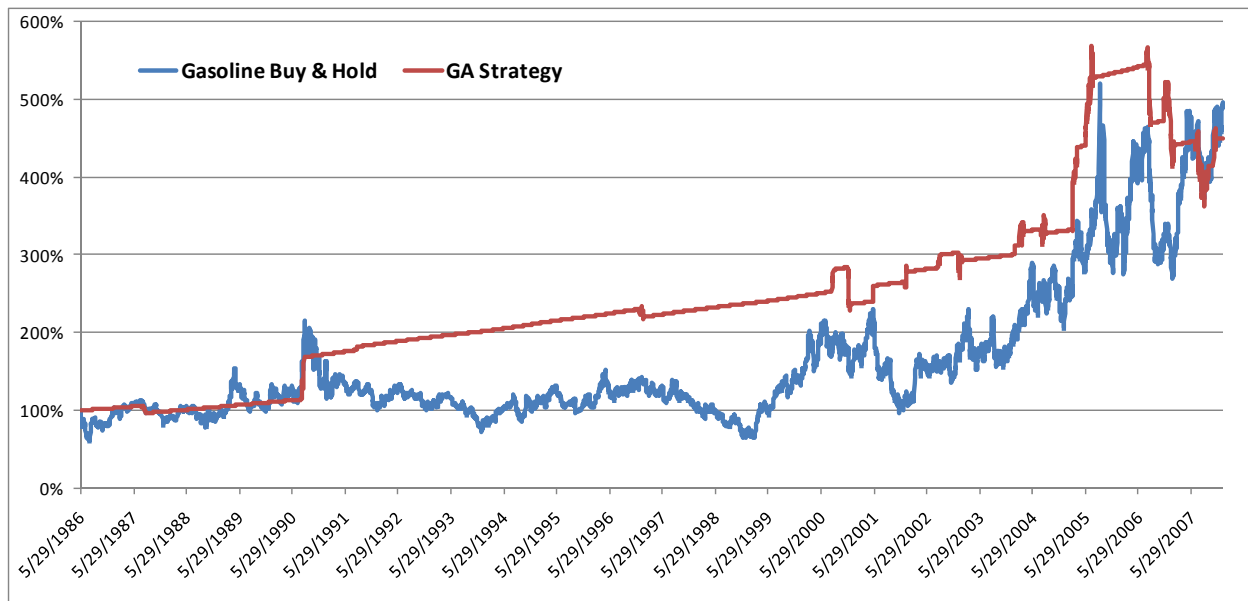
**Experiment set-up:** For this experiment, a population size of 500 was used for 50 generations. Fitness was defined as cumulative excess return over a buy and hold strategy. Transaction costs were set to 0%. The training periods were 1986-2002, 1991-1997, 1996-2002, each of which was followed by one year selection period, and out-sample evaluations were from one year after selection period to 2008.

**Results & Discussion:** As the following summary table shows, on average, slight negative excess returns were found.

In-sample	K	Excess	K <sup>+</sup>	N <sub>b</sub>	r <sub>b</sub>	σ	N <sub>s</sub>	r <sub>s</sub>	σ	r <sub>b</sub> -r <sub>s</sub>	(t)	T*
1986 – 1992	9	-0.0530	1	1677	+0.000339	0.024009	2076	+0.000612	0.024974	-0.000299	(-0.375)	0
1991 – 1997	10	-0.0744	0	705	+0.000863	0.027057	1793	+0.000537	0.027132	+0.000326	(+0.225)	0
1996 – 2002	10	-0.0820	1	538	+0.001794	0.026184	713	+0.000863	0.029237	+0.000931	(+0.342)	1

Among the various trading rules developed throughout all the different in-sample periods, only two of the rules yielded excess returns, in the amounts of 2.48% and 6.87%. Of the two rules one seems non-sensical, and the other is based on relative change in crude stocks, gasoline stocks and heating oil prices. The following graph shows the the best performing trading strategy developed, versus a buy and hold strategy.





The best rule from the gasoline experiment appears to be fairly conservative—it stays out of the market for long periods of time (up to 5 years) during periods of lower volatility. However, it is more likely to enter the market during periods of high volatility, as can be seen from the trades in the past few years.

Examining the graph reveals that although the best strategy underperformed the buy and hold during the total out-sample as defined, during many parts of the out-sample the strategy did indeed outperform. This highlights the notion that multiple measures of return (e.g. Sharpe Ratio, Sortino Ratio, etc.) should be examined before actually executing any strategy developed by this or any other algorithm.

There are several potential reasons why excess returns might not have been found. First, the length of training data was relatively limited, compared to the S&P 500 experiment for example. Although there were 14 different predictive variables included in the model, several additional variables were unable to be included because either they were not available for a long enough time span, or they were not freely available in the public domain, for example DOE and EIA stated predictions of future gasoline prices, or any measure of future or options volatility. Second, the use of a population size of 500 and 50 generations might have possibly over-fit the model to the past data, at the expense of future predictive ability. Third, the model has no way to take into account non-quantifiable events, such as the spike in gas prices caused by hurricane Katrina in 2005, or any other change in price caused by the geopolitical economy.

## V. Conclusions:

Across all experiments, no significant excess returns were found by using the genetic algorithm to develop investment strategies.

1. Foresight Experiment: Passing this experiment (successfully identifying the profitable strategy that was purposely put into the data) was the minimum necessary condition for the genetic algorithm to be able to potentially generate profitable trading strategies. The next day's price was given to the algorithm, and it was able to successfully detect this fact to give the intuitive trading strategy that is most profitable.
2. S&P 500 Experiment: This experiment recreates Allen and Karjalainen's 1999 work, mostly undertaken as a learning exercise, but also to show consistency of results. An interesting note here is in Allen and Karjalainen's original experiment, the computational time required using pre-year 2000 technology took over a day's worth of computing time; by comparison, approximately the same experiment took under two hours to process on 2007 hardware.
3. China A Shares Experiment: On an elementary technical level, China A-Shares market appears to be somewhat efficient, given institutional regulations. Results question validity of buy-and-hold as a benchmark against which genetic algorithm-generated strategies are compared, because buy-and-hold returns are sensitive to the last few data points included in the experiment. A profitable strategy can look less promising (and vice versa) just due to high or low returns on the last few days. It is important to compare excess returns from buy-and-hold with returns of out-sample minus in-sample, and its associated statistical significance; profitable strategies will show significant positive (at least, non-negative) difference between out-sample and in-sample returns. There appears to remain potential for profitable trading strategies that incorporate additional variables, such as fundamentals.
4. Gasoline Prediction Experiment: Although no excess returns were found, the results seem to suggest that incorporating additional fundamental data into the genetic algorithm can lead to improved results.

Although no excess returns were found in these experiments, there are many ways to potentially improve both the structure and use of the algorithm, both in terms of building blocks the model has to

work with (greater sophistication of solution templates), and choice of inputs into the model (wider variety of input variables - potentially preprocessed in line with fundamental analysis). There is room to improve the model both on the “art” and “science” front.

The use of genetic algorithms for investment strategy development is a nascent concept which has great potential.

#### **Suggested Future Experiments and Speculations:**

**Penalties against nonsensical and over complex output:** When developing investment strategies with a genetic algorithm, there is always the risk of over-fitting the strategy to the past data set, at the expense of losing predictive ability for the future. Often over-fit investment strategies use an above average level of complexity, both in variables used and random numbers generated. One way to attempt to cut down on this might be to change the fitness function from only accounting for positive additions to excess return to also incorporating a negative value, penalizing for excess levels of complexity. The new function would be in the form of  $[Fitness = +F*excess\_return -P*complexity\_factor]$ . This could potentially reduce overfitting, and could also make execution easier, particularly in situations where liquidity and transaction costs play large roles.

**Limited set of linear building blocks:** One potential limitation of the algorithm is that all of the solution templates worked with in this paper were comprised of linear building blocks. Given that the maximum possible branch depth of the solution candidates was generally 6, it is possible to build polynomial strategies from the linear building blocks, but unlikely to build a polynomial strategy that can incorporate a significant amount of variables.

**Starting with a human developed strategy, then letting the genetic algorithm tweak specific variable levels:** In addition to generating trading strategies from scratch, one potential use of the genetic algorithm would be to optimize parts of a trading strategy already created by a human. For example, an options trading strategy based on a trader’s personal belief about the volatility levels in the market might be dynamically hedged in the underlying asset’s market with assistance from a genetic algorithm. Another example would be if a fund manager believes that an equity seems underpriced based on fundamental analysis, and wants to purchase a large quantity of the stock without significantly moving the price. In this case, a genetic algorithm could be used in a constrained optimization model to

formulate the purchase of the position over a fixed amount of time, while minimizing the effects of trading costs and market movements.

**Purposely over-fitting to the past and then adopting the inverse strategy:** given the risk of over-fitting to the past, how can one use over-fitting purposefully to achieve superior returns? One of the traditional beliefs of all varieties of efficient market theories is that strategies which yield above average returns in the past based on analysis of historical prices will not be able to consistently outperform in the future, because all past information becomes incorporated into the market. If this is the case, one idea would be to purposefully over-fit the data to the past to create a strategy that yields extraordinary excess returns on historical data, and to then take the inverse of that strategy as a new strategy going forward. The idea behind this, broadly labeled “contrarianism”, would be that anything that has worked so well in the past would underperform in the future thanks to the “invisible hand” of the market, so trading on the inverse of that strategy could potentially therefore outperform in the future.

**Intra-day considerations:** As mentioned earlier, the algorithm is programmed in Mathematica, which is a bit too slow to be practical in intra-day trading – each run takes upwards of several days of computing time (when dozens of indicators are used, for example). However, it may be possible to use a small assortment of technical indicators (price, volume, measures of momentum, etc) and generate potentially profitable intra-day trading strategies. For these applications, it may be preferable to program a similar algorithm in a faster language such as C++.

**Asset allocation experiment across multiple markets, asset classes:** Genetic algorithms have been proven to be successful heuristics in many examples of constrained optimization problems, both in engineering and in project finance. Building on this body of knowledge, one area for experimentation is for a constrained optimization model to be set up to distribute a large amount of total assets across multiple different assets. Constraints can come in the form of minimum investment in each asset group, or in the fitness function of maximizing returns while minimizing risk. This sort of model can be useful in multiple scenarios, including global asset allocation among major asset management firms, risk management for national financial institutions, or a fund of funds allocating assets to different hedge funds. This experiment could yield results beneficial for investors of all sizes, from sovereign wealth funds diversifying their country’s equity market exposure all the way down to an individual choosing an optimal 401(k) allocation.

**Testing of optimal distribution of data among training, selection and out-sample periods:** Taking a quantitative approach to the issue of how to divide available data into training, selection and out-sample testing periods, it would be beneficial for one to examine the effects of different data division strategies on the out-sample returns of a common experiment. Although certain elements of data division are unique to each particular experiment, perhaps one could find various heuristics to help with the process. Again, the need for this sort of experimentation highlights how the use of genetic algorithms in investment strategy development is both an art and a science.

## Works Cited

Allen, Franklin, Risto Karjalainen. "Using genetic algorithms to find technical trading rules." Journal of Financial Economics 51(1999): 245-271.

Buffett, Warren, multiple letters to Berkshire Hathaway shareholders and various interviews.

Fernández-Rodríguez, Fernando, Christian González-Martel, Simón Sosvilla-Rivero. "Optimization of technical rules by genetic algorithms: evidence from the Madrid Stock Market." Applied Financial Economics 15(2005): 773–775.

Hsieh, Ting-ya, Hsin-Lung Yu. "Genetic Algorithm for Optimization of Infrastructure Investment Under Time-Resource Constraints." Computer-Aided Civil and Infrastructure Engineering 19(2004): 203-212.

Tong, Tam and Chan. "Genetic Algorithm Optimization in Building Portfolio Management." Construction Management and Economics 19, (2001): 601-609.

Wang, K.J., S.H Lin. "Capacity Expansion and Allocation for a Semiconductor Testing Facility under Constrained Budget." Production Planning & Control 13(2002): 429-437.

Whitman, Martin J., and Martin Shubik. The Aggressive Conservative Investor. Wiley, 1979.