Dynamic-Radius Species-Conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index Stocks

Conference Paper · July 2013

DOI: 10.1007/978-3-642-39712-7_3

3 authors, including:

Michael Scott Brown
University of Maryland, Baltimore County
40 PUBLICATIONS **96** CITATIONS

Dynamic-Radius Species-Conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index Stocks

Michael Scott Brown, Michael J. Pelosi, and Henry Dirska

University of Maryland University College, Adelphi Maryland, USA
michaels.brown@faculty.umuc.edu,
mpelosi@maui.net,
dirska@nova.edu

Abstract. This research uses a Niche Genetic Algorithm (NGA) called Dynamic-radius Species-conserving Genetic Algorithm (DSGA) to select stocks to purchase from the Dow Jones Index. DSGA uses a set of training data to produce a set of rules. These rules are then used to predict stock prices. DSGA is an NGA that uses a clustering algorithm enhanced by a tabu list and radial variations. DSGA also uses a shared fitness algorithm to investigate different areas of the domain. This research applies the DSGA algorithm to training data which produces a set of rules. The rules are applied to a set of testing data to obtain results. The DSGA algorithm did very well in predicting stock movement.

Keywords: Niche Genetic Algorithm, Genetic Algorithm, stock forecasting, financial forecasting, classification, black-box investing.

1 Introduction

Forecasting the price movements of stocks is a difficult task. The possible financial reward of picking the correct direction that a stock will move has created much interest in developing systems to predict such behavior. Early work on formal financial forecasting began in the early 1900's [1] and continues to this day [2]. In the last 20 years a variety of techniques have been used to predict stock movement. These include Genetic Algorithms (GAs), Neural Networks and other artificial intelligence techniques.

A variety of methods use GAs and Genetic Programming (GP) to predict stock and security movements [3-5]. These methods take different approaches. Some research uses GAs and GPs to develop classification rules, while others use GAs in hybrid approaches [2]. Much research has been done using these evolutionary approaches to perform black-box investing.

This paper presents a new system for financial forecasting using a Niche Genetic Algorithm (NGA). The presented research used an NGA and a set of financial data to derive a set of classification rules that the research later applied to another set of data. The training and test data each comes from a full quarter of stock prices from the

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 27–41, 2013.

Dow Jones Index. These stocks represent 30 of some of the largest companies in the United States.

2 Literature Review

Over the last few decades, hundreds of papers have been written on GAs. But only a small number of these studies use GAs to classify and forecast stock market data. While sparse, research that uses genetic algorithms for this purpose validates its effectiveness and value for future research. Medical research also supports using classifiers based on GAs.

The following literature review discusses these points and lists significant papers supporting them. The literature review also provides a brief explanation of genetic algorithms, lists a few significant early studies, and suggests that NGA's can be effective in a variety of domains.

2.1 Genetic Algorithms

GAs are very useful algorithms that can locate optima within very complex domains. Early research in the subject began the 1950's [6]. The 1970's was another period of important research in the area [7-9]. Research in the subject has been going on continuously since the 1950's and with more powerful computing power we are beginning to see more applied uses for GAs.

A GA is a specific type of search technique that models biological systems. *Individuals* within a population are modeled after values in the domain. Each individual has *genes* that represent different traits. A *Fitness Function* is used to determine an individual's *Fitness*, which is how well an individual copes with the environment or domain. A GA begins with an initial population, normally randomly generated. Each generation goes through three biological operations: *Selection*, *Cross-over* and *Mutation*. The fitness of each individual in the population is used to determine which individuals will reproduce. Individuals with higher fitness have a greater chance of reproducing. The process of selecting two individuals to reproduce is called selection. Cross-over is the process of taking some genes from each parent and producing new offspring. To encourage exploration within the domain, some genes are changed or mutated based upon a *mutation rate*. Selection, cross-over and mutation produce a new generation. Over numerous generations the population converges to the optimum within the domain.

2.2 Genetic Algorithms as Classifiers

Because GAs are search techniques they are not normally thought of first as being tools that classify. However, GAs can be used for classification. GAs have been used as data classifiers in the diagnosis of cancer [10]. They have been used in financial security forecasting [3-5]. While not the most common use of a GA, they can be used to classify data.

The challenge of using a GA for classification is how to represent the search space. Individuals in GAs represent domain values. But classification rules are more

complex than simple domain values. Rules have conditions and conclusions. Conditions can be very complex. Mapping these complex rules into strings allows GAs to seek out optimal rules.

De Jong, Spears and Gordon [11] used a GA to develop an algorithm called GABIL. GABIL represents classification rules as individuals in disjunctive normal form. Each individual within the population is a classification rule. The left-most genes describe a condition and the right-most genes indicate the class that data matching the description should be placed into. In their research they present an example [11]. The example assumes that the rule is attempting to classify an object as a *widget* or *gadget.* There are two characteristics of the objects: size and shape. For size there are values of {*small*, *medium*, *large*} and for shape there is {*sphere*, *cube*, *brick, tube*}. Each value for the characteristics and conclusions are represented by a binary value in the Individual's genes. An Individual containing the following gene sequence, 11110000, would represent the following rule:

111		1000		0
If object is *small, medium* or *large*	And	object is *sphere*	Then	object is *widget*

GABIL uses variable-length rule sets meaning that multiple rules could be joined together to form an individual. This is unusual for GAs that normally have fixed length individuals. A set of training data was used to determine the best rules. GABIL is used to classify medical data on patients to determine if they have breast cancer.

A second GA classification method is from Booker, Goldberg and Holland [12]. This is a hybrid algorithm that combines a GA with a bucket brigade algorithm. This algorithm used a GA to discover rules. The bucket brigade algorithm is used to evaluate and assign credit to rules generated by the GA. This algorithm was designed for domains that are very fluid, in which new information is constantly coming in. An example given is to guide a robot that is to locate certain objects in the environment while trying to avoid other objects. In the example the robot moves and the object moves giving an ever changing domain space to optimize.

The Booker, Goldberg, Holland [12] algorithm defined individuals much differently than GABIL. This algorithm assigns meaning to binary values of 0 and 1, but also introduces a new symbol #. This new symbol is used when the value of the corresponding position is not included in the rule. If the genes of an individual represent movement, size and color respectively, then 1#0 would represent the following condition:

1		#		0
Object is moving	AND	object size is irrelevant	AND	object is black

These conditions then get associated with an action that the robot can take, which typically is some type of movement. As stated earlier these results of the GA are passed to another algorithm, bucket brigade. This algorithm determines the effectiveness of each rule.

The two GAs presented in this literature review represent two types of domains to be optimized. GABIL represents optimization problems in which the domain is

unchanged. The Booker, Goldberg, Holland algorithm contains an ever changing domain. The problem of classifying stocks exhibit characteristics of both types of optimization.

2.3 Genetic Algorithms as Financial Forecasters

There have been hundreds of systems developed to forecast financial data [13]. Most use some type of artificial intelligence technique from neural networks [20] to GP. Atsalakis and Valavanis [13] state that a goal of the research community is to produce the best results using the least amount of information about the stocks and by developing the least complex model. Many financial forecasting algorithms, including the one presented in this paper, attempt to meet these two conditions.

Mahfoud and Mani [3] developed a GA that classifies stocks by having each individual within a population represent a classification rule. They used a clustering NGA to develop classification rules that were applied to a large number of stocks. The goal of the research was to make a prediction about each stock by classifying them into groups to buy or sell.

The Mahfoud and Mani [3] research provides a novel way to represent stock classification rules for GAs. Each individual represents a classification rule. Each piece of data used in the experiment is given a number of genes in the individual. The first two bits represent the numerical condition: 00 is >; 01 is <; 10 is = and 11 is !=. The remaining genes store the value for the rule that corresponds to the data about the stock. This can be any number of genes based upon the precision needed. Finally a bit used to represent if the stock meets the condition should be bought or sold. Now rules can be created like:

If Price < 15 and EPS > 1 Then Buy [3]

The Mahfound and Mani [3] research uses 15 attributes of the stocks used and covers a 12 week period. The GA searches for optimal rules. The evaluations rules are then applied to each stock. Results show that the algorithm correctly predicts stock movement 47.6% of the time. It makes no prediction 45.8% of the time. And it incorrectly predicts the direction of a stock only 6.6% of the time [3]. The research presented in this paper is very similar to Mahfoud and Mani with a few exceptions. The method defined in this research attempts to select a single stock to purchase each week. It also uses a specialized NGA.

A variety of other research uses GAs for financial forecasting. Tsang, Markose and Er [4] use a GA to attempt to locate temporary misalignment between options and futures. When these conditions happen within a market investors can position themselves to profit. Wagman [5] uses a GA to evaluate entire portfolios. Other methods create hybrid approaches [14] by combining GAs with other methods. GAs are probably not the most popular method to predict stock prices, however results for many methods show promise.

2.4 Niche Genetic Algorithms

NGAs are specialized GAs that attempt to find multiple optima within a domain. NGAs are often used for finding local maximums and minimums of functional optimization problems, while traditional GAs are used for global maximums and minimums. NGAs generally fall into one of two categories: Fitness Sharing and Crowding. NGA Fitness Sharing methods alter the fitness function to prevent global convergence. This is done through adjusting an individual's fitness based upon how close it is to other individuals [15]. NGA Crowding methods replace individuals from one generation with ones from a previous generation [8, 16, 17]. Most of these methods forum groups, called clusters, of individuals that are within a predetermined radius. Typically the strongest member of each cluster is moved into the new generation. This promotes genetic diversity. NGAs are useful in searching domains in which multiple optima exist.

The Dynamic-radius Species-conserving Genetic Algorithm (DSGA) is a recently develop NGA framework that performs very well in a variety of domains [18]. DSGA enhances a traditional crowding NGA, SCGA [16], with a tabu list [19] that is used to encourage exploration. The tabu list stores investigated areas of the domain. Exploratory techniques are used within the algorithm to encourage the population to investigate other areas of the domain. Previous results of DSGA show that it competes very well against other NGAs [18].

3 DSGA and Financial Forecasting

The DSGA is a clustering algorithm framework that uses a tabu list and varies the radius during execution. The tabu list stores optimal areas of the domain that have already been investigated. Future generations are encouraged to explore other areas of the domain. As in most clustering algorithms a radius parameter is used to determine the area of a cluster in the domain. DSGA varies the radius at fixed intervals as the algorithm runs. This helps mitigate poor radius choices.

3.1 Algorithm Overview

This research uses two sets of data. DSGA is run against training data and produces a set of rules. The rules are then applied to the second set of data, the test data. The set of rules are used to select a stock to purchase in the future. By applying rules to data from week x, a single stock is selected to be purchased in week $x + 1$. This is done by summing up the number of rules that indicate to buy the stock and subtracting the number of rules that indicate to sell the stock.

$$\text{Purchase\_Indicator}(stock) = \#\text{\_buy\_rules}(stock) - \#\text{\_sell\_rules}(stock) \quad (1)$$

This is referred to as the *purchase indicator*. This can be seen in the Data Flow Diagram in Figure 1.

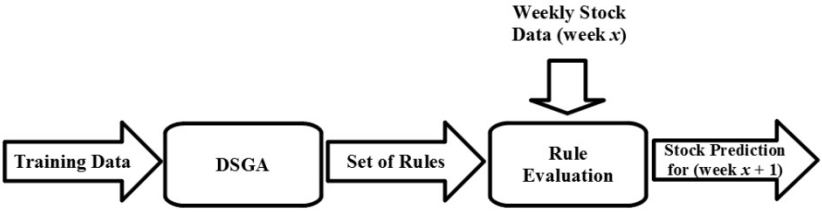

Fig. 1. The DSGA Financial Forecasting algorithm accepts training data and uses DSGA to locate a set of rules based upon the training data. The rules are used in a Rule Evaluation component that takes weekly stock data and predicts a stock to purchase the following week.

The algorithm maintains a list of rules, which is the tabu list, as the NGA runs. Periodically the algorithm analyzes the current generation and looks for convergence. Areas in the domain that the generation converges to are included on the list of rules. During this phase of the algorithm the seeds are also placed on the tabu list. This is discussed in a future section. The algorithm goes through multiple phases with each phase having the opportunity to add rules to the tabu list through convergence or seeds.

3.2 Individuals and Genes

The genes used to create individuals form a rule that can be applied to certain stocks. The first bit represents the decision to buy the stock or sell the stock. Arbitrarily, 1 represents a decision to buy and 0 represents a decision to sell. The remaining genes are divided into sections for each characteristic in the rule. This research uses four stock characteristics which are described in Table 1.

Table 1. Characteristic of stocks

Characteristic	Description
Percent stock changed	The percent change in the stock price for week *x*, which is the week prior to the week that the algorithm attempts to predict the stock for.
Percent volume changed	The percent change in volume during week *x* compared to week *x - 1*. Volume is the number of shares of a stock sold.
Days to next dividend	The number of days until the next dividend. If the stock does not have a dividend, this value is the maximum integer.
Percent return of next dividend	The percent return based upon the stock price of week *x* of the amount of the dividend. If the company does not give out dividends, this value is 0.

Each section begins with a bit to indicate if the characteristic is used in the rule. This allows rules to be made up of different combinations of characteristics. The next two bits indicate the condition for the characteristics (<=, <, > or >=). The remaining bits in the section determine the value for the rule. These are binary numbers and have implied decimal places where appropriate. In some cases there is a sign bit. When there is a sign bit it is always the first digit. All numbers are stored in little-endian format, with least significant digits to the left. Table 2 shows the actual 43 genes.

The following is an example of the gene sequence 1111001001001001010110111111011010101010100. This individual corresponds to the following rule:

1	1	11	0100100	1	00	10101101	1	11	1101101	0	10	1010100
Buy Rule	Use in	>=	-1.8	Use in Rule	<	90	Use in Rule	>=	91	Do Not Use in Rule	>	0.21

This rule states the following: Buy stock if all of the following conditions are met: percent change of stock price during week x >= -1.8 and percent change volume of throughout week x < 90 and days to the next dividend >= 91

Table 2. Positions of genes within individual

Position(s)	Description
1	1 = buy, 0 = sell
2	1 = include percent change price in the rule, 0 = don't include it
3-4	00 <; 01 <=; 10 >; 11 >= for percent change price
5-12	number for percent change price rule with one implied decimal place. First bit is sign bit 0 -; 1 +
13	1 = include percent volume change in the rule, 0 = don't include it
14-15	00 <; 01 <=; 10 >; 11 >= for percent volume change
16-23	number for percent change volume rule. First bit is sign bit 0 -; 1 +
24	1 = include days to next dividend in the rule, 0 = don't include it
25-26	00 <; 01 <=; 10 >; 11 => for days to next dividend
27-33	number for days to next dividend (max 127 days)
34	1 = include percent return on dividend in the rule, 0 = don't include it
35-36	00 <; 01 <=; 10 >; 11 => percent return on dividend
37-43	number for percent return with 2 implied decimal places.

3.3 Distance

There are a number of ways to determine distance in a GA. There is genetic difference, where the number of genes that have different values. There is also Euclidean distance. When rules for this algorithm are applied to data sets, they return order pairs consisting of a stock and week. A rule may return zero or more (stock, week) pairs. This research uses the intersection of the (stock, week) pairs between two individuals. If the rules for two individuals retrieve many of the same (stock, week) pairs, the distance between the individuals is very small. This research uses the intersection of the two sets of (stock, week) pairs to determine distance. The distance function is defined as:

$$\text{distance}(i_1, i_2) = 1 / |(\text{sw}(i_1) \cap \text{sw}(i_2))| \quad (2)$$

The function sw returns the (stock, week) pairs for the conditions encoded in the individual. A distance function is needed for all GAs that use fitness sharing.

3.4 Fitness Function

In order to choose individuals for the selection process, each individual's fitness must be computed. The fitness function should correspond to the goal of the GA. In this case the goal is the highest percent return in the following week, week $x + 1$. The fitness function retrieves all data for (stock, week) combinations that match the rule. The fitness is the average of the stock price percent returns for the following week. This fitness function encourages the GA to locate stocks that should produce favorable returns in the following week.

The first bit of the individual indicates if the rule suggests buying or selling the stock. If the individual corresponds to a rule to sell, then high fitness would have a negative return for the following week. So, for the rules that indicate selling the stock, the fitness function returns -1 times the average of the price percent return for the following week.

There were some minor manipulations for the fitness function. It is possible that by the fitness function defined above, there could be a negative fitness. This can cause issues with selection, so all fitness values were increased by 10. Also, having rules that only correspond to a small amount of data in the training data set is not very useful. So a parameter, w, was introduced. If a rule does not retrieve at least w (stock, week) pairs in the training data set, the fitness is 0. This encourages the GA to locate rules that apply to larger sets of data and not focus in on global optima. These modifications to the standard fitness function allow the algorithm to locate better rules.

Unlike other GAs DSGA does not use the fitness function for selection. It uses a shared fitness function. The fitness function value is altered to encourage exploration in new areas of the domain. The tabu list stored potential optimal areas of the domain that have already been explored. The shared fitness function will decrease the fitness of individuals close to these areas. The shared fitness for an individual i is, $\text{sf}(i) = \text{fitness}(i) / m_i$. The value for m_i is calculated as follows. The variable TLj is the jth individual on the tabu list.

$$m_{i=} \sum_{j=1}^{TabuSize} \left(1 - \left(\frac{distance(i,TLj}{0.1}\right)\right) \quad (3)$$

Shared fitness encourages the algorithm to explore areas of the domain not on the tabu list.

3.5 Parameters

There are a variety of parameters used in DSGA. Some are found in all GAs like population size and mutation rate. Some are found in other NGAs like radius. Some are unique to DSGA. Table 3 shows the parameters used in DSGA along with a description of their purpose.

Table 3. DSGA parameters

Parameter	Variable	Description
Population Size	N	The number of individuals in each generation.
Number of Generations	NG	The number of generations before the GA terminates.
Mutation Rate	M	The probability that a gene will be mutated.
Seed Radius	IS	The size of the radius at initialization.
Radius Delta	SD	The size of the change of the radius.
Reevaluation Loop Count	RLC	The number of generations in the intervals for deciding candidates for the tabu list.
Convergence Limit	CL	The number of identical individuals to be placed on the tabu list.
Weight	W	The number of (stock, week) pairs that must be returned by a rule in order to have a fitness other than 0.

3.6 DSGA Algorithm

DSGA is a typical clustering GA. Within the basic selection, crossover and mutation steps of a GA, DSGA incorporates additional steps of seed selection and seed conservation. A *seed* is the strongest individual within a population for some area of the domain. Seeds are selected by sorting the population by fitness. In the case of DSGA the sorting is done by the shared fitness as described in the section above. Individuals are evaluated for seed selection from the fittest individual to the least fit. If no other individuals within a predefined radius are seeds, then the individual is a seed. When a

new generation is created, seeds from the previous generation replace individuals in the next generation. Each seed replaces the weakest individual within the predefined radius. If no individuals are within the radius, the seed replaces the globally weakest individual. This ensures that seeds are carried into the next generation. In each generation seeds are evaluated, so a seed in one generation may not be a seed in the next generation. Many clustering GAs work in this way [16].

After *RLC* number of generations, DSGA evaluates the current generation. DSGA puts all current seeds on the tabu list. It also analyzes the current generation for convergence. If there are *CL* or more identical individuals within the generation, one of them is placed on the tabu list also. Then all individuals placed on the tabu list are replaced within the population with randomly generated individuals. The radius is then changed by the Sigma Delta, *SD*. In this research the radius is always increased by *SD*. Changing the radius helps the algorithm locate other optima. A known limitation to clustering algorithms is poor selection of the radius parameter [21]. Varying the radius helps mitigate this limitation. The algorithm then generates another *RCL* number of generations before it performs the evaluation again. Table 4 shows pseudocode for the DSGA algorithm.

DSGA has shown promise in locating local optima [18]. Tests against other NGAs show that it is very competitive. DSGA is especially good at locating arbitrarily close optima.

Table 4. DSGA pseudocode

Line	Pseudocode
1	Initialization
2	While not termination condition
3	For (int $r = 1$; $r < RLC$; r++)
4	Seed Selection
5	Selection
6	Crossover
7	Mutation
8	Seed Conservation
9	End for loop
10	If there exists an individual d with CL or more identical individuals then
11	Add d to tabu_list
12	Replace all individuals identical to d with randomly generated individual
13	End if
14	Add the seeds of the current generation to the tabu_list
15	Alter radius by *SD*
16	End while loop
17	Output the tabu_list – these are the generated classification rules

4 Results

Data for this research was obtained from the Dow Jones Index stocks, which are 30 of some of the largest American companies. The training data came from the first quarter of 2011 and the test data came from the second quarter of 2011. In the testing data, one stock was purchased each week based upon data from the previous week and the rules generated from DSGA. It was assumed that a constant amount of money was used to purchase each stock. Calculations in this section are based upon the stock being purchased at the opening price at the beginning of the week, which is usually Monday. The calculations are based upon the stock being sold at the closing price on the last day of the week, usually Friday.

As mentioned in the previous section, it is possible in a week for two stocks to have the same purchase indicator. In situations like this a stock characteristic is used to break the tie and resolve the conflict. The characteristic of percent change in price was used in this research. If multiple stocks have the same purchase indicator for determining to purchase the stock in week $x + 1$, whichever stock had the greatest price percent gain in week x would be selected. Because two stocks rarely have the exact same price percent increase for a given week, this is a good characteristic to use

4.1 Parameter Settings

Table 5 shows the parameter settings for this research. Their definitions can be found in Table 3. The parameter values were determined through experimentation prior to running the eight trials.

Table 5. Parameter settings

Parameter	Variable	Value
Population Size	N	25
Number of Generations	NG	80
Mutation Rate	M	0.01
Seed Radius	IS	0.05
Radius Delta	SD	0.01
Reevaluation Loop Count	RLC	20
Convergence Limit	CL	2
Weight	W	10

4.2 Selected Stocks

Tables 6 shows the stocks selected for one of the trials. A stock was selected for the 13 weeks in the experiment. The second column for each trial shows the percent return of the stock in the following week. A total return and weekly return for each trial is shown in Figure 2.

Table 6. Stocks select for trial 4

Week	Stock	Return	Week	Stock	Return
1	T	-0.10%	8	PFE	2.15%
2	MRK	2.46%	9	PFE	-0.76%
3	MRK	-0.47%	10	DIS	-1.74%
4	DIS	1.79%	11	VZ	0.79%
5	MRK	-0.14%	12	T	-0.72%
6	T	0.67%	13	VZ	5.00%
7	DIS	0.58%	Total		9.51%

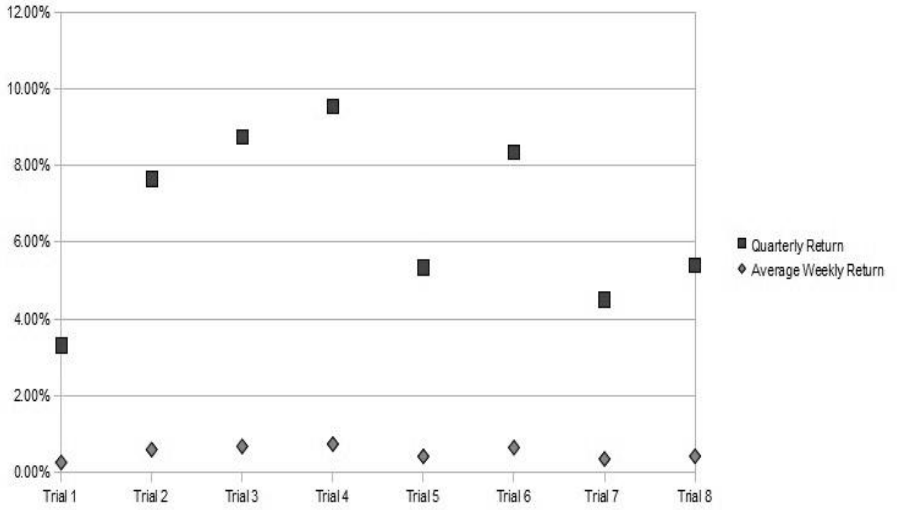

Fig. 2. Percent return per trial

4.3 Rates of Return

The results for this research show the rate of return for the quarter and average weekly rate of return. Since a constant amount of money is invested each week, the rate of return for the quarter is the sum of each weekly return. The following graph shows the percent return for each of the 8 trials.

The results for DSGA are compared to other indicators during this period. The data shown in the table below for DSGA is the average of 8 runs. *Maximum* is the true optima. This is the rate of return assuming the best performing stock is selected each

week. *Minimum* is the rate of return for selecting the worst performing stock each week. The *Dow Jones Index* shows the rate of return by investing an equal amount of money in each of the 30 stocks. Finally, the *Average of Stocks* shows the rate of return for averaging all stocks in the index. The Dow Index and Average of Stocks are realistic returns that many investors obtain. Table 7 shows the results for DSGA and the indicators.

Table 7. DSGA results compared against other indicators

Methods/Indicators	Quarter	Week
DSGA	7.075%	0.54%
Maximum	55.90%	4.30%
Minimum	-57.02%	-4.39%
Dow Jones Index	1.66%	0.13%
Average of Stocks	2.50%	0.19%

Of the 8 runs of the DSGA algorithm the average rate of return for the quarter was 7.075% or 28.3% a year. This corresponds to a 0.54% return per week. The standard deviation was 2.166.

4.4 Discussion of Results

While the return of DSGA was not optimal, 55.9%, it did do very well against the Dow Jones Index. It outperformed the Dow Jones Index by more than a factor of three. Even one standard deviation away from the average still beat the Dow Jones Index by almost 4%. All of the trials outperformed the Dow Jones Index. The minimum return was 3.3% which outperformed the Dow Jones Index 1.64%. The maximum return of the trials was 9.5% which outperformed the Dow Jones Index by 7.84%. These results show that the DSGA algorithm did very well against the Dow Jones Index.

To see if the results could be a coincidence the T-test was performed. The T-test was performed with the following values, $\mu = 1.66$, $n = 8$, average = 7.075, $s = 2.166$ and $\alpha = 0.001$. In the T-test μ is the expected value if the algorithm had no ability to select stocks that would increase in value. This comes from the performance of the Dow Jones Index. The value n is the number of trials. 7.075 is the average of the results of the experiment. The value s is the standard deviation of the results. Finally, α is the confidence level being tested. When calculated the t value comes out to be 7.1 which is well within the 0.001 confidence value. The T-test shows that within a 0.001 confidence level the results of this research were not a coincidence. The T-test is a good statistical analysis test for experiments with a low number of trials, normally considered $n < 30$.

5 Conclusion and Future Work

The DSGA algorithm did very well in predicting single stock selection for a week of the 30 Dow Jones Index stocks. It produced returns many times greater than the Dow Jones Index, which is often considered a safe and lucrative investment selection. The Dow Jones Index stocks make a great set of stock for forecasting systems because if the system predicts a stock incorrectly losses are normally minimal. DSGA produces these results by examining only four stock characteristics: change in stock price, change in stock volume, days until the next dividend and return of next dividend.

The results of this research suggest many other areas of future work. Future research could be to use the DSGA algorithm on other stock data sets, like the S&P 500, NASDAQ 1,000 and FTSE Eurotop 100. This researched used a three month timeframe for training data and test data. This was an arbitrary timeframe. Research in other timeframes could be another area of future research. Although the parameters for the algorithm produced very good results there may be better parameter values. The DSGA algorithm could also be used for other data classification problems. Future research in these areas could produce better returns and expand our understanding of stock forecasting.

Portfolio management is another possible area for future research. Instead of using the algorithm to predict a stock to purchase, the algorithm could be used to evaluate portfolios of already purchased stock. That algorithm could then be used to recommend changes to the portfolio.

Stock forecasting is a very complex problem because it is based upon many factors. Some of these factors are human bias which cannot be represented by mathematical models. NGAs seem capable of locating rules that produce very attractive returns. The DSGA algorithm presented in this paper produced returns many times greater than the stock index that it was based upon. Results indicate that it is suitable for stock forecasting.

References

1. Graham, B., Dodd, D.: Security Analysis. McGraw-Hill, New York (1934)
2. Wang, J.J., Wang, J.Z., Zhang, Z.G., Guo, S.P.: Stock Index Forecasting Based on a Hybrid Model. Omega 40(6), 758–766 (2012)
3. Mahfoud, S., Mani, G.: Financial Forecasting Using Genetic Algorithms. Applied Artificial Intelligence 10, 543–565 (1996)
4. Tsang, E., Markose, S., Er, H.: Chance Discovery in Stock Index Option and Future Arbitrage. New Mathematics and Natural Computation 1(3), 435–477 (2005)
5. Wagman, L.: Stock Portfolio Evaluation: An Application of Genetic-Programming-Based Technical Analysis (2003)
6. Bremermann, H.J.: The Evolution of Intelligence. The Nervous System as a Model of its Environment (Technical Report, No.1, Contract No. 477, Issue 17). Seattle WA: Department of Mathematics, University of Washington (1958)
7. Cavicchio, D.J.: Adaptive Search Using Simulated Evolution. Unpublished doctoral dissertation. University of Michigan, Ann Arbor (1970)

8. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems (Doctoral dissertation, University of Michigan). Dissertation Abstracts International, 36(10), 5140B (University Microfilms No. 76-9381) (1975)
9. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
10. Dolled-Filhert, M., Ryden, L., Cregger, M., Jirstrom, K., Harigopal, M., Camp, R.L., Rimm, D.L.: Classification of breast cancer using genetic algorithms and tissue microarrays. Clinical Cancer Research 12, 6459–6468 (2006)
11. De Jong, K.A., Spears, W.M., Gordon, D.F.: Using Genetic Algorithms for Concept Learning. Machine Learning 13(2-3), 161–188 (1993)
12. Booker, L.B., Goldberg, D.E., Holland, J.H.: Classifier Systems and Genetic Algorithms. Artificial Intelligence 40, 235–282 (1989)
13. Atsalakis, G.S., Valavanis, K.P.: Survey Stock Market Forecasting Techniques - Part II: Soft Computing Methods. Expert Systems with Applications 36, 5932–5941 (2009)
14. Armano, G., Marchesi, M., Murru, A.: A Hybrid Genetic-Neural Architecture for Stock Indexes Forecasting. Information Sciences 170(1), 3–33 (2005)
15. Goldberg, D.E., Richardson, J.: Genetic Algorithms with Sharing for Multimodal Functional Optimization. In: Proceedings of the Second International Conference on Genetic Algorithms and their Application, Cambridge Massachusetts, pp. 41–49 (1987)
16. Li, J.P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A Species Conserving Genetic Algorithm for Multimodal Function Optimization. Evolutionary Computation 10(3), 207–234 (2002)
17. Ling, Q., Wa, G., Yang, Z., Wang, Q.: Crowding Clustering Genetic Algorithm for Multimodal Function Optimization. Applied Soft Computing 8, 88–95 (2008)
18. Brown, M.S.: A Species-Conserving Genetic Algorithm for Multimodal Optimization (Doctoral dissertation). Available from Dissertations and Theses database, UMI No. 3433233 (2010)
19. Glover, F.: Tabu Search – Part I. ORSA Journal on Computing 1(3), 190–206 (1989)
20. Cao, Q., Parry, M.E.: Neural Network Earnings Per Share Forecasting Models: A Comparison of Backward Propagation and the Genetic Algorithm. Decisions Support Systems 47(1), 32–41 (2009)
21. Ando, S., Kobayashi, S.: Fitness-based Neighbor Selection for Multimodal Function Optimization. In: Proceeding of the 2005 Conference on Genetic and Evolutionary Computation, Washington DC, pp. 1573–1574 (2005)