Trading Rule Optimization

By Yichi Zhang, Xinyan Hu

Overview

- The idea of the project
- Description of the market strategy and market data
- Genetic algorithm, PSO algorithm and adjusted PSO algorithm
  - Genetic Algorithm Description
  - PSO & adjusted PSO
- Analysis of the results
- Suggestions for further improvements

The Idea of the Project

Background of Genetic Algorithm(GA)

- After Abhishek Gupta's introduction in the morning about machine learning from Math works, we want to concentrate on one simple method in machine learning——Genetic Algorithm, to explain how it helps our optimization in financial area. MATLAB has strong ability on simulation, so we also offer the results of our algorithm with the help of MATLAB.

Our Trading Methodology

- **Core Idea:**

  Stock prices move in certain trends in short periods. Therefore, we can make profit in current period from experience of the previous period.

- **Brief Structure:**

  We create a group of trading strategies. Each includes indices that determine the entry and exit point.

  We are using historical data iteratively modify and filter the strategies with help of AI algorithms

Description of the Market Strategy and Market Data

Market Strategy

The goal is to automatically create strategies for stock that fetch high returns with the following indices:

- Relative Strength Index (RSI)
- Moving Average(MA)
- Exponentially-Weighted Moving Average(EWMA)
- Stochastic Oscillator (KD)
- EMPTY

Market Data

- The data is the close prices of 30 randomly selected stocks.

Apple Inc.
Bank of America Corporation
Bed Bath & Beyond Inc.
BioTelemetry, Inc.
Citigroup Inc.
The Coca-Cola Company
DELL Problem
Delta Air Lines Inc.
First Solar, Inc.
General Electric Company
Google Inc.
The Home Depot, Inc.
International Business Machines Corporation
JPMorgan Chase & Company
Macy's, Inc.

McDonald's Corporation
Microsoft Corporation
Netflix, Inc.
The New York Times Company
Pepsico, Inc.
The Procter & Gamble Company
SAP AG
Seagate Technology Public Limited Company
Sony Corporation
Time Warner Cable Inc.
US Airways Group Inc.
Volkswagon AG
Wal-Mart Stores Inc.
Western Digital Corporation
Yahoo! Inc.

- The data frequency used in the project is hourly from Bloomberg. There are 7 hours in one day.

Date	Last Price	Volume	SMAVG (15)
03/21 15:30	82.12	1.450M	0.429M
03/21 15:00	82.12	0.335M	0.342M
03/21 14:30	82.02	0.227M	0.330M
03/21 14:00	81.77	0.240M	0.325M
03/21 13:30	81.91	0.184M	0.317M
03/21 13:00	81.89	0.214M	0.316M
03/21 12:30	82.11	0.137M	0.312M
03/21 12:00	82.12	0.143M	0.317M

- We use three years of historical data.

Difference between MATLAB GA algorithm and our GA algorithm

1	0	1	0	0

- Advantages:

  The change of one unit will alter its meaning in a random direction. It allows more flexible change in phenotype.

- Character:

  (1)Normal genes combine units together to explain one meaning.

  (2)In MATLAB toolbox, each 'string' or 'double vector' is one gene. The combinations of genes explain different meanings.

Index	Index’s Level

- Advantage:

  The change of one unit will lead to a more predictable result for one gene.

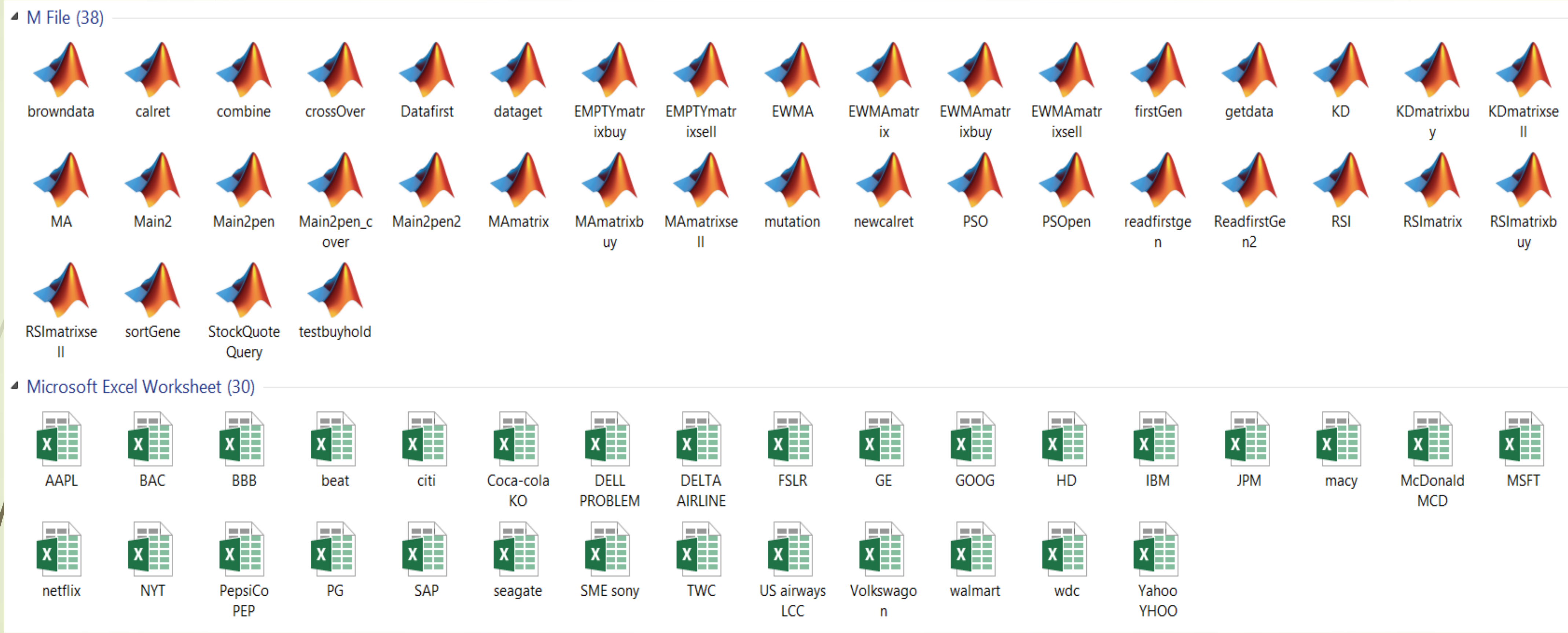
- Character:

  (1)We package the strategies with its coefficient(or level of index) as one gene, while MATLAB GA treats each unit as a gene.

  (2)One example of our gene is as above, which are a symbol of strategy and its level. They serve as a gene together. In other words, they are packaged. One gene explains one meaning.

Genetic Algorithm, PSO Algorithm and Adjusted PSO Algorithm

Programming Using Matlab



Programming Using Matlab

```matlab
function [ret ce]= ReadfirstGen2(port,p,aa,rsi,ma,ewma,down,up,i,method)
    [m n]=size(port);
    [mm nn]=size(method);
    [mmm nnn]=size(p);
    method2{1,1}='EMPTYmatrixbuy(p)';
    method2{2,1}='EWMAmatrixbuy(p,aa,down,up,port(i,j+n/4))';
    method2{3,1}='KDmatrixbuy(p,port(i,j+n/4))';
    method2{4,1}='MAmatrixbuy(p,ma,port(i,j+n/4))';
    method2{5,1}='RSImatrixbuy(p,rsi,port(i,j+n/4))';
    method2{1,2}='EMPTYmatrixsell(p)';
    method2{2,2}='EWMAmatrixsell(p,aa,down,up,port(i,j+n/4))';
    method2{3,2}='KDmatrixsell(p,port(i,j+n/4))';
    method2{4,2}='MAmatrixsell(p,ma,port(i,j+n/4))';
    method2{5,2}='RSImatrixsell(p,rsi,port(i,j+n/4))';


    ce=cell(2,n/4);
for j=1:(n/4)
    for l=1:mm
        if port(i,j)==l
            url=method2{l,1};
            ce{1,j}=zeros(mmm,nnn);
            ce{1,j}=eval(url);
        end
    end
end

for j=1:(n/4)
    for l=1:mm
        if port(i,j+n/2)==l
            url=method2{l,2};
            ce{2,j}=zeros(mmm,nnn);
            ce{2,j}=eval(url);
        end
    end
end
```

```matlab
[wholeq wholerq]=dataget;
stocknumber=10;
whole=wholeq(:,stocknumber);
wholer=wholerq(:,stocknumber);
[m n]=size(whole);
wholer=[ones(1,n);wholer;];


portnum=300;
%Cross-Over
crosspercent=0.2;
kill=0.2;
%Mutation
changerange=0.4;
%Circulation
generation=1;
dayy=26;
step=10;

%consider top 'top_rank' strategies at final step
top_rank=5;

%-----manage the size of the method matrix, column # is fixed
METHOD_STRUCT = {'EMPTY',0,0,0,0;
                 'EWMA',-0.005,0.005,0.005,0.005,;
                 'KD',20,10,20,10;
                 'MA',-0.020,0.020,0.020,0.020;
                 'RSI',0.4,0.1,0.6,0.1};

% METHOD_STRUCT = {'RSI',0.4,0.1,0.6,0.1
%                  'EWMA',-0.005,0.005,0.005,0.005,;
%                  'MA',-0.020,0.020,0.020,0.020;
%                  'KD',20,10,20,10;
%                  };


%------ controling the effect;
THE_GENE_LENGTH=2;
%------

learnlength=500*7;
```

Why GA ?

- Black Box Optimization:

  In science and engineering, a black box is a device, system or object which can be viewed in terms of its input, output and transfer characteristics without any knowledge of its internal workings.

  Typical problems of financial can be reformulated as problems of global optimization, where objective function is a black box.

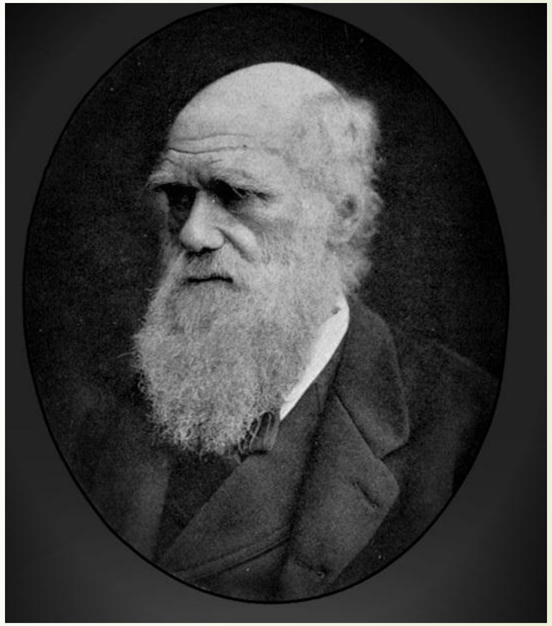
- Adaptive Systems:

  This optimization algorithm allows to create adaptive systems, where the selection of the optimization method is determined by the input data. So forget about the availability of distribution.

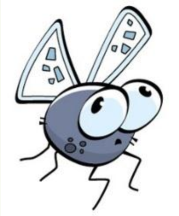
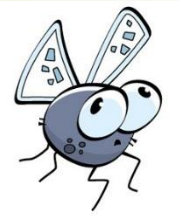
- Efficient Hybrids: The development of our GA algorithm was done under a novel framework of deriving global optimization methods based on the approach similar to smooth optimization. Instead of the derivative of the objective function, the approach is based on the randomization of the original problem. It can be shown that the parameters of the algorithm do not depend on the type of randomization. This approach allows to better understand the parameters of the optimization algorithms and can be used for constructing efficient hybrids based on available MATLAB modules.

How Does the Algorithm Work？

- **<u>Key words:</u>**
- Iteration
- Individual
- Generation
- Fitness
- Extinction/kill
- Crossover
- Mutation

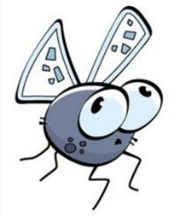


An Example of the Process

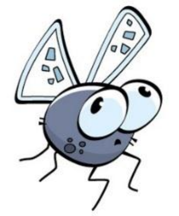
		Disguise Color	Strong Wings	Enemy Detection	Long Proboscis
	45%	✓	✗	✓	✗
	60%	✗	✗	✓	✓
	15%	✗	✗	✗	✓

An Example of the Process

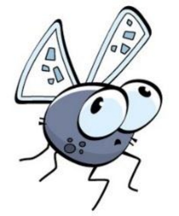
An Example of the Process



45%



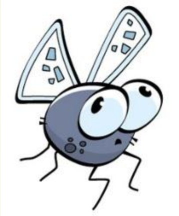
60%



70%

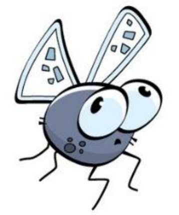
	Disguise Color	Strong Wings	Enemy Detection	Long Proboscis
45%	✓	✗	✓	✗
60%	✗	✗	✓	✓
70%	✓	✗	✓	✓

An Example of the Process



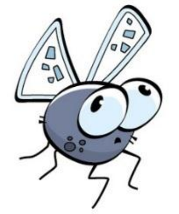
Disguise Color	Strong Wings	Enemy Detection	Long Proboscis
✓	✗	✓	✗

45%



Disguise Color	Strong Wings	Enemy Detection	Long Proboscis
✗	✓	✓	✓

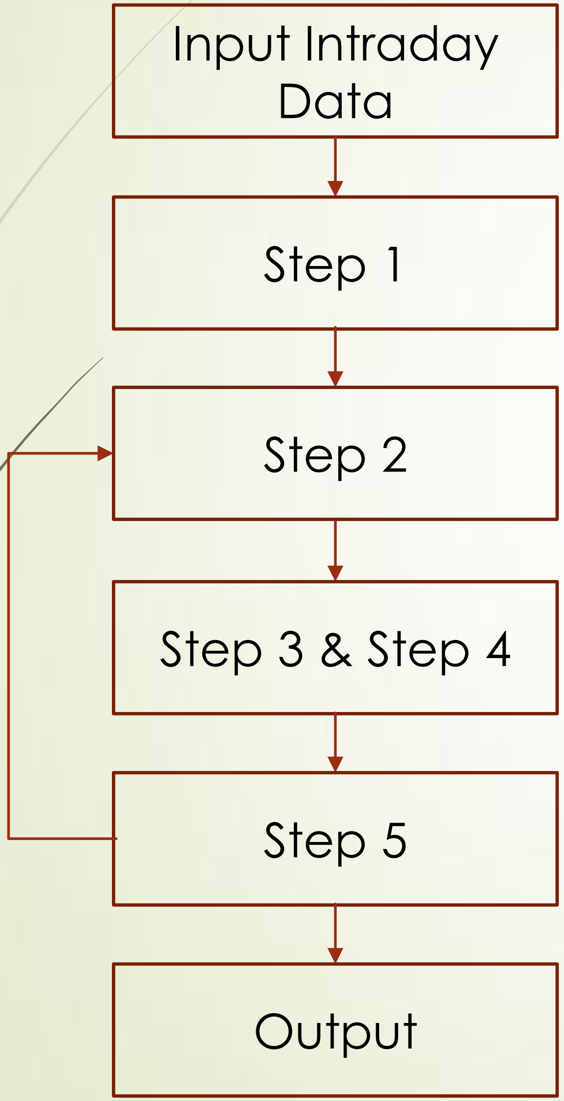
65%



Disguise Color	Strong Wings	Enemy Detection	Long Proboscis
✓	✗	✓	✓

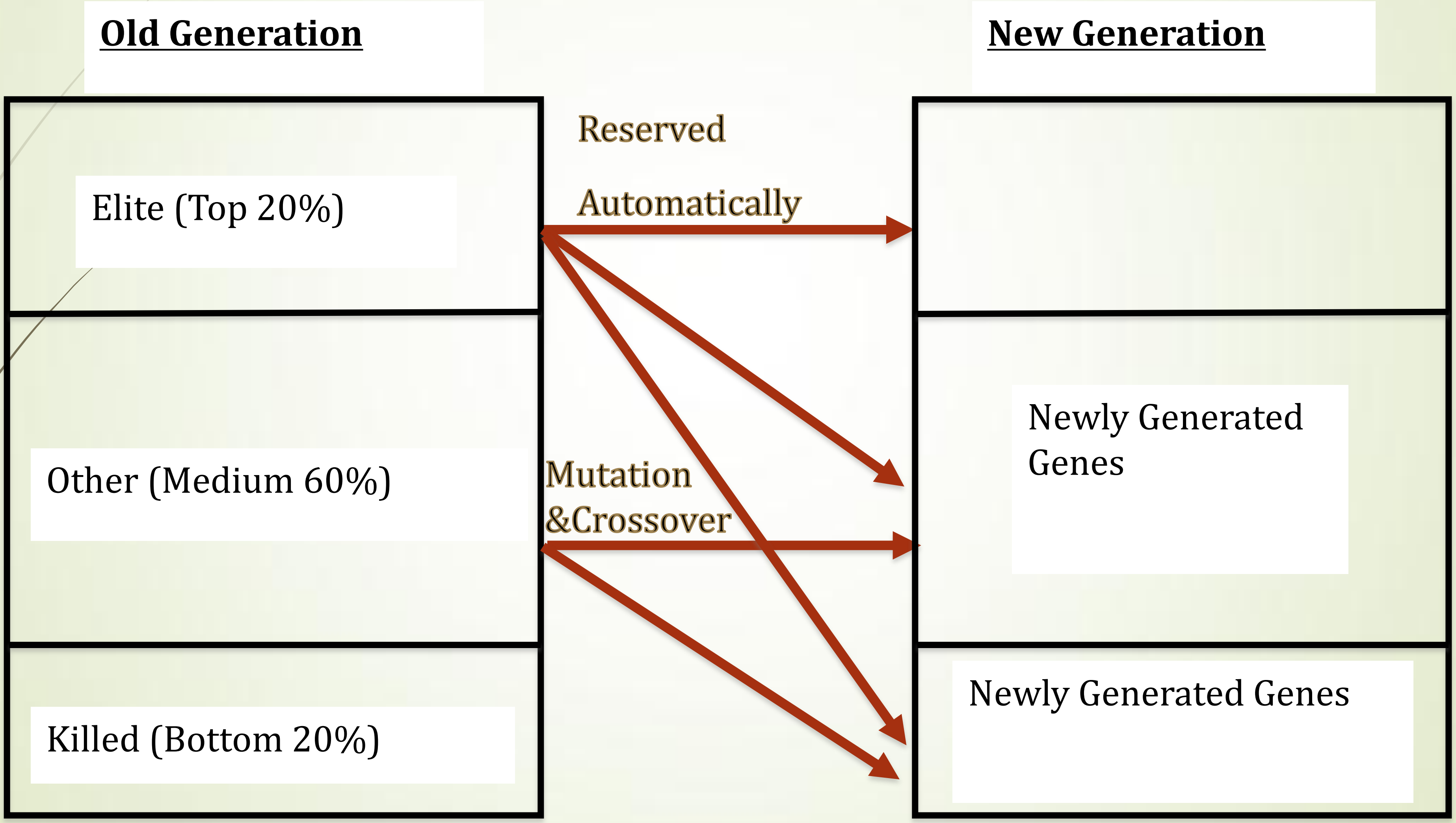
70%

Steps of the Algorithm



- 1. Randomly generate a certain sized group of individuals;
- 2. Evaluate scores of each individual through a fitness function;
- 3. Rank the individuals in the generation according to fitness value in descending order;
- 4. Take top genes(say top 20%) as elite, directly reserve them into the next generation. Kill the worst genes (say bottom 20%), as they are less favorable for improving the overall population fitness.
- 5. Generate new individuals through crossover and mutation.
- 6. Go back to step 2. The exit condition is that the time reaches the goal period.

How Does the Algorithm Work？



How Does the Gene Look Like？

A	**B**	**A’s Level**	**B’s Level**	**C**	**D**	**C’s Level**	**D’s Level**

Entry Strategy **Exit Strategy**

- In our project, there are four different indices for determine the timing of buy/sell the stock.
- Each gene selects two from the four indices for entry strategy and two for exit strategy
- A,B,C,D contains an indicator of the indices the gene uses.

PSO and adjusted PSO in the Project

- One of the reason we chose PSO is because of its iterative nature
- The Swarm here is the group of individual strategies
- The Particles are the candidate strategies
- Use the first entry and first exit strategy gene to divide the particles into groups. In each group, the candidates' corresponding index levels are adjusted towards best one in the category.

- Formula for PSO and adjusted PSO:

  PSO: P_new=(P_best-P_move)*ChangeRate+P_move

  Adjusted-PSO: P_new=(P_best-P_move)*ChangeRate*Condition+P_move

**Condition* can take values 0 or 1.

Analysis of the Results

Assumption Test

GA					
	Returns of top 5 strategies in testing periods				
Period 1	0.00%	0.00%	0.00%	0.00%	0.00%
Period 2	-1.62%	-1.62%	-1.62%	-1.26%	-4.19%
Period 3	-0.07%	-0.07%	-0.84%	0.80%	-0.15%
Period 4	3.13%	3.18%	9.20%	0.33%	0.33%
Period 5	0.00%	-6.84%	-6.84%	0.00%	-0.32%

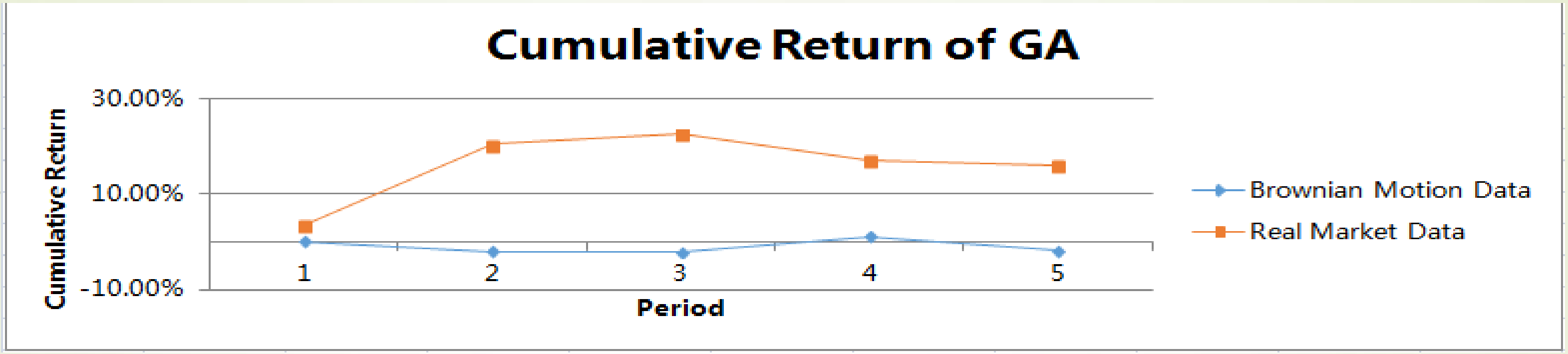
GA-PSO					
	Returns of top 5 strategies in testing periods				
Period 1	0.00%	-1.45%	-1.16%	-2.01%	-1.34%
Period 2	-1.82%	-0.98%	0.00%	0.00%	0.00%
Period 3	0.26%	-3.09%	0.00%	0.23%	-4.90%
Period 4	-1.25%	-4.21%	-6.10%	-3.84%	0.44%
Period 5	0.00%	0.00%	1.41%	0.00%	0.20%

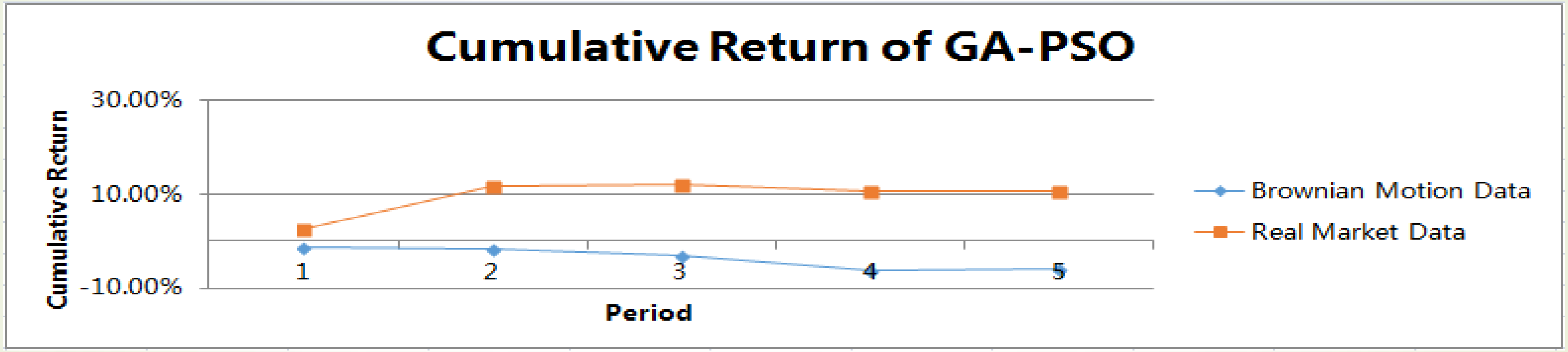
GA-adjusted PSO					
	Returns of top 5 strategies in testing periods				
Period 1	0.90%	0.24%	0.24%	0.24%	0.24%
Period 2	-4.50%	-4.50%	-4.81%	-2.47%	-4.59%
Period 3	3.63%	3.63%	3.63%	0.57%	3.63%
Period 4	0.18%	3.30%	3.13%	3.13%	3.13%
Period 5	0.00%	0.00%	-0.85%	-0.85%	-6.84%

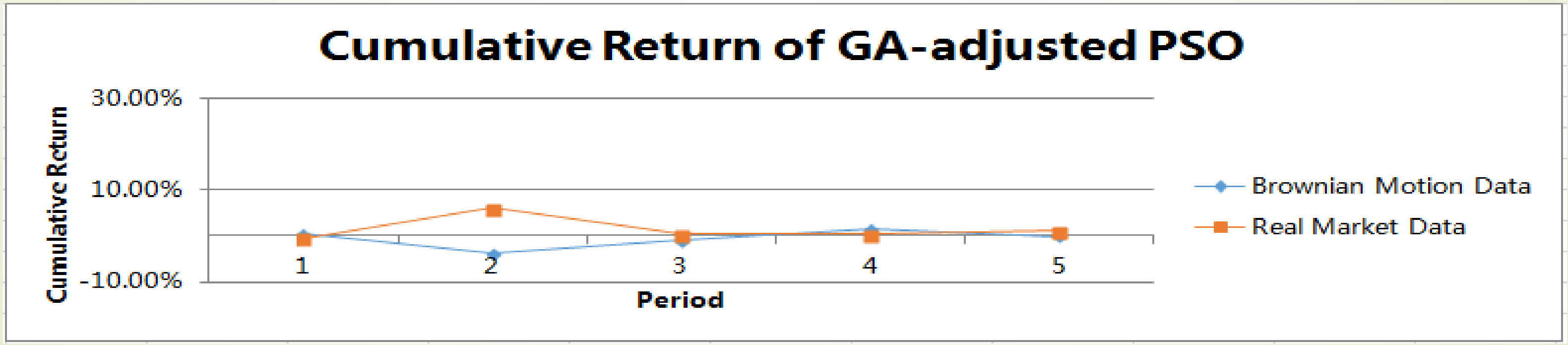
As a reference, we can consider the case when stock returns have no serial dependence: the Brownian motion(with a zero drift).

The returns of average top 5 strategies generated using each three methods.

Portfolio					
	1	2	3	4	5
GA	3.51%	16.35%	1.79%	-4.36%	-0.93%
GA-PSO	2.46%	9.06%	0.49%	-1.46%	0.18%
GA-adjuste	-0.46%	6.67%	-5.38%	-0.11%	0.77%


Cumulative Return of GA
30.00%
10.00%
-10.00%
Cumulative Return
1
2
3
4
5
Period
Brownian Motion Data
Real Market Data


Cumulative Return of GA-PSO
30.00%
10.00%
-10.00%
Cumulative Return
1
2
3
4
5
Period
Brownian Motion Data
Real Market Data


Cumulative Return of GA-adjusted PSO
30.00%
10.00%
-10.00%
Cumulative Return
1
2
3
4
5
Period
Brownian Motion Data
Real Market Data

Learning with Single Stock

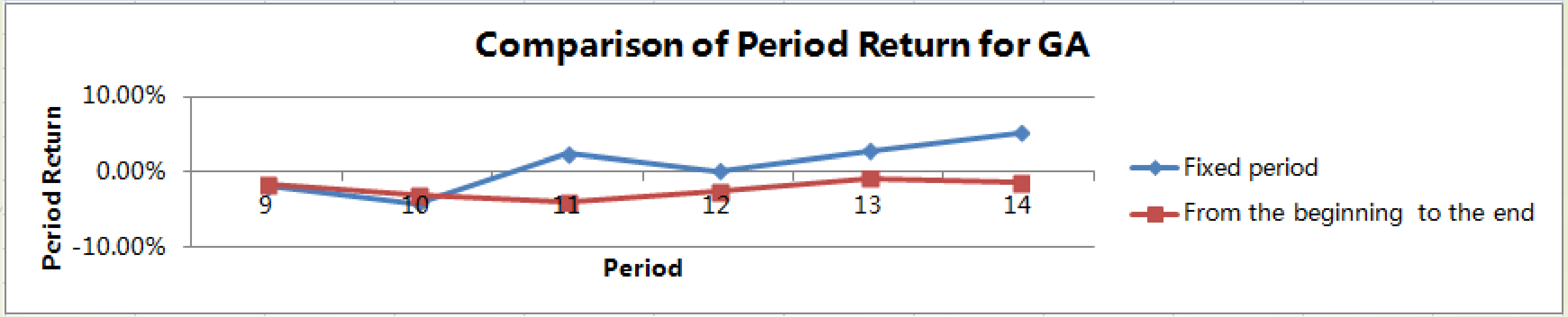
GA						
Test period number	9	10	11	12	13	14
Fixed period	-1.89%	-4.11%	2.46%	0.16%	2.76%	5.14%
From the beginning to the end	-1.64%	-3.06%	-3.96%	-2.56%	-0.82%	-1.32%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-PSO						
Test period number	9	10	11	12	13	14
Fixed period	1.54%	2.14%	2.17%	-1.22%	1.07%	-0.28%
From the beginning to the end	0.60%	6.21%	1.24%	-1.39%	0.48%	-1.22%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-adjusted PSO						
Test period number	9	10	11	12	13	14
Fixed period	-0.31%	-3.96%	-0.62%	-0.42%	1.54%	4.43%
From the beginning to the end	4.43%	-4.27%	-4.36%	-2.23%	0.09%	-0.45%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%

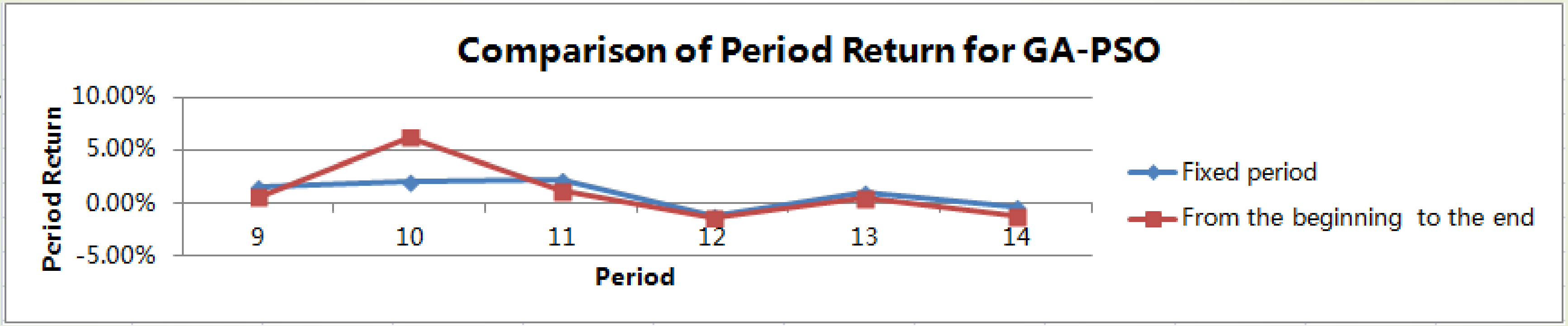
1.Fixed period has a specific length. In the simulation, we pick 8 periods as the training set.
2.From the beginning to the end means start from the first period until current time period.
3.Buy-hold means equal-weighted portfolio of all 30 stocks.

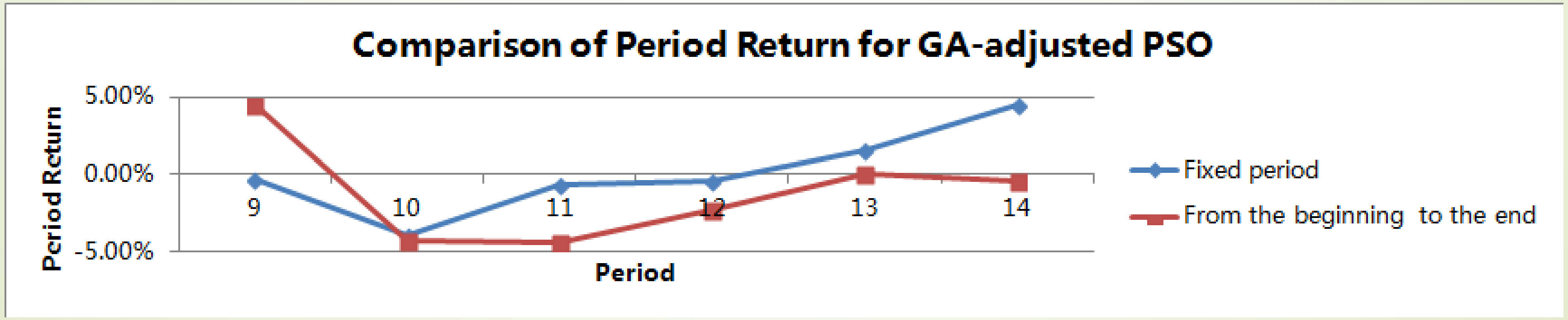
Learning with Single Stock

GA						
Test period number	9	10	11	12	13	14
Fixed period	-1.89%	-4.11%	2.46%	0.16%	2.76%	5.14%
From the beginning to the end	-1.64%	-3.06%	-3.96%	-2.56%	-0.82%	-1.32%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-PSO						
Test period number	9	10	11	12	13	14
Fixed period	1.54%	2.14%	2.17%	-1.22%	1.07%	-0.28%
From the beginning to the end	0.60%	6.21%	1.24%	-1.39%	0.48%	-1.22%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-adjusted PSO						
Test period number	9	10	11	12	13	14
Fixed period	-0.31%	-3.96%	-0.62%	-0.42%	1.54%	4.43%
From the beginning to the end	4.43%	-4.27%	-4.36%	-2.23%	0.09%	-0.45%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%

Prediction power seems to deteriorate with longer learning periods. It may be due to old information being less relevant.


Comparison of Period Return for GA
10.00%
0.00%
-10.00%
Period Return
9
10
11
12
13
14
Period
Fixed period
From the beginning to the end

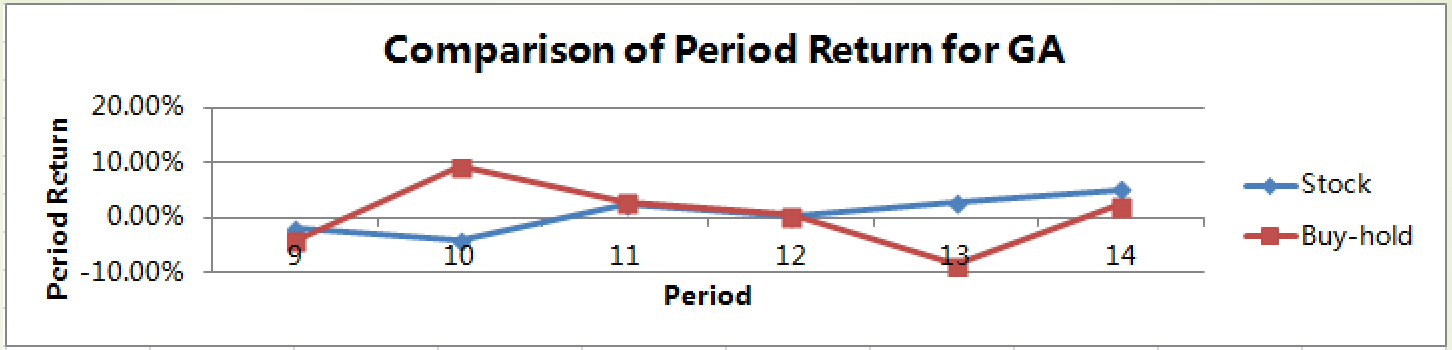

Comparison of Period Return for GA-PSO
10.00%
5.00%
0.00%
-5.00%
Period Return
9
10
11
12
13
14
Period
Fixed period
From the beginning to the end

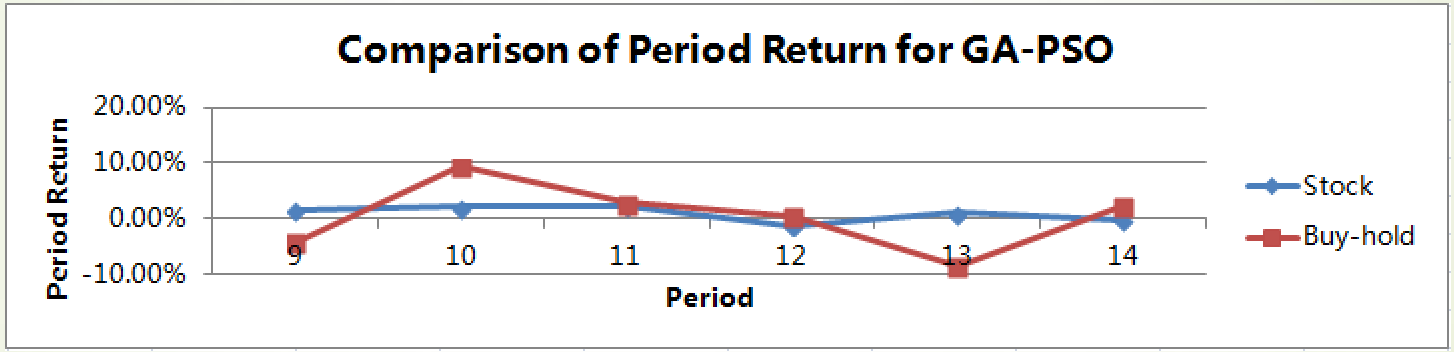

Comparison of Period Return for GA-adjusted PSO
5.00%
0.00%
-5.00%
Period Return
9
10
11
12
13
14
Period
Fixed period
From the beginning to the end

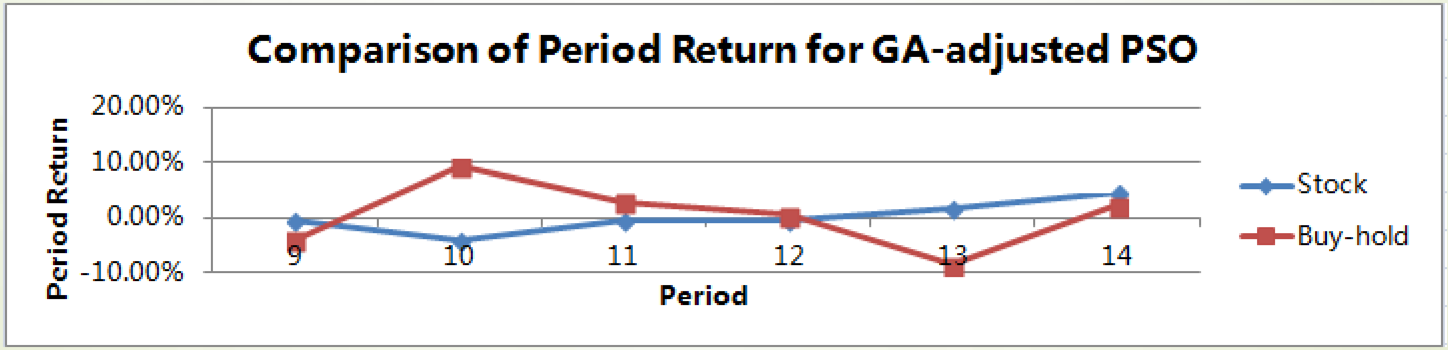
Learning with Single Stock

GA						
Test period number	9	10	11	12	13	14
Fixed period	-1.89%	-4.11%	2.46%	0.16%	2.76%	5.14%
From the beginning to the end	-1.64%	-3.06%	-3.96%	-2.56%	-0.82%	-1.32%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-PSO						
Test period number	9	10	11	12	13	14
Fixed period	1.54%	2.14%	2.17%	-1.22%	1.07%	-0.28%
From the beginning to the end	0.60%	6.21%	1.24%	-1.39%	0.48%	-1.22%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
GA-adjusted PSO						
Test period number	9	10	11	12	13	14
Fixed period	-0.31%	-3.96%	-0.62%	-0.42%	1.54%	4.43%
From the beginning to the end	4.43%	-4.27%	-4.36%	-2.23%	0.09%	-0.45%
Buy-hold	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%

There seems to be a delay for the method to recognize the up trend. The strategies are conservative——they won't trade if they don't pick up a trend.


Comparison of Period Return for GA
20.00%
10.00%
0.00%
-10.00%
Period Return
9
10
11
12
13
14
Period
Stock
Buy-hold


Comparison of Period Return for GA-PSO
20.00%
10.00%
0.00%
-10.00%
Period Return
9
10
11
12
13
14
Period
Stock
Buy-hold


Comparison of Period Return for GA-adjusted PSO
20.00%
10.00%
0.00%
-10.00%
Period Return
9
10
11
12
13
14
Period
Stock
Buy-hold

Learning with Portfolio

GA			
Testing period	Portfolio	Stock	Buy-hold
9	3.51%	-1.89%	-4.03%
10	16.35%	-4.11%	9.29%
11	1.79%	2.46%	2.83%
12	-4.36%	0.16%	0.46%
13	-0.93%	2.76%	-8.53%
14	0.16%	5.14%	2.39%

GA-PSO			
Testing period	Portfolio	Stock	Buy-hold
9	2.46%	1.54%	-4.03%
10	9.06%	2.14%	9.29%
11	0.49%	2.17%	2.83%
12	-1.46%	-1.22%	0.46%
13	0.18%	1.07%	-8.53%
14	1.67%	-0.28%	2.39%

GA-adjusted PSO			
Testing period	Portfolio	Stock	Buy-hold
9	-0.46%	-0.31%	-4.03%
10	6.67%	-3.96%	9.29%
11	-5.38%	-0.62%	2.83%
12	-0.11%	-0.42%	0.46%
13	0.77%	1.54%	-8.53%
14	0.09%	4.43%	2.39%

Here we show average returns of 5 top strategies.

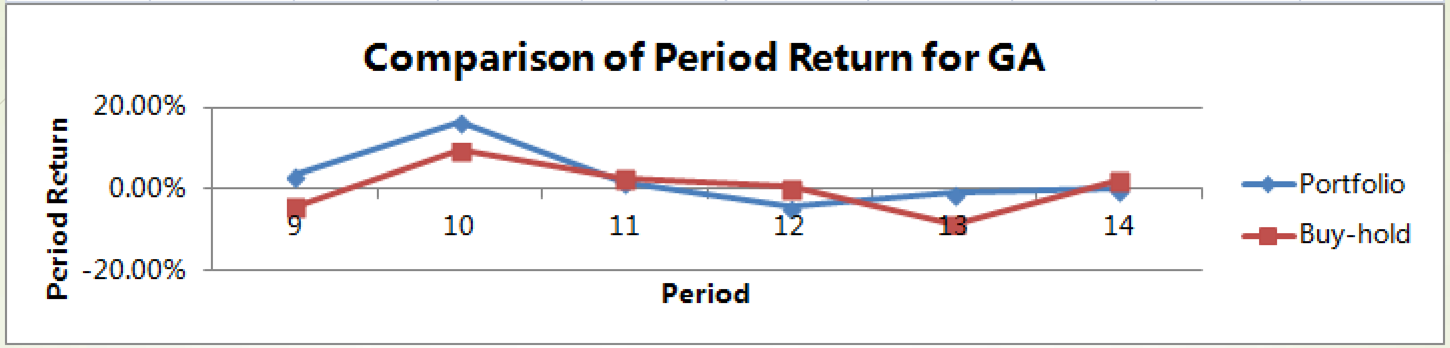
Learning with Portfolio

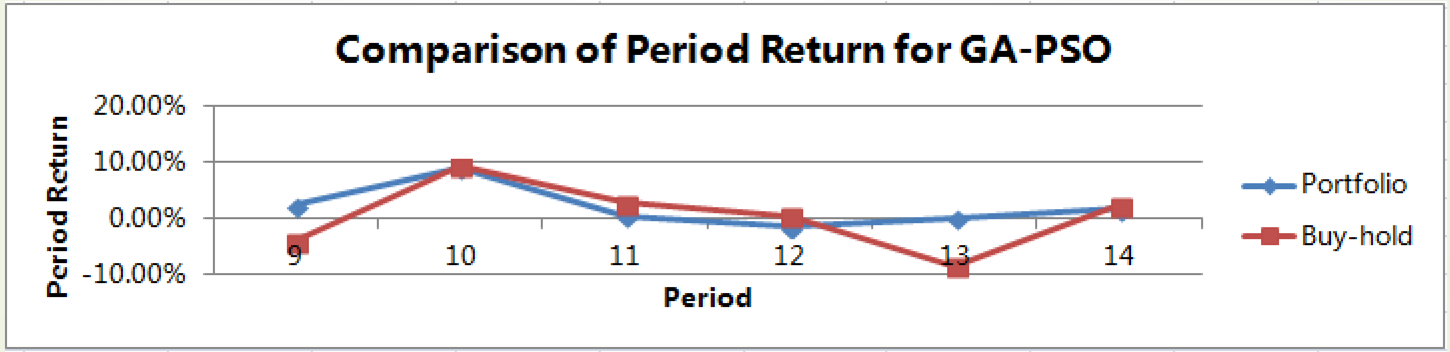
GA			
Testing period	Portfolio	Stock	Buy-hold
9	3.51%	-1.89%	-4.03%
10	16.35%	-4.11%	9.29%
11	1.79%	2.46%	2.83%
12	-4.36%	0.16%	0.46%
13	-0.93%	2.76%	-8.53%
14	0.16%	5.14%	2.39%

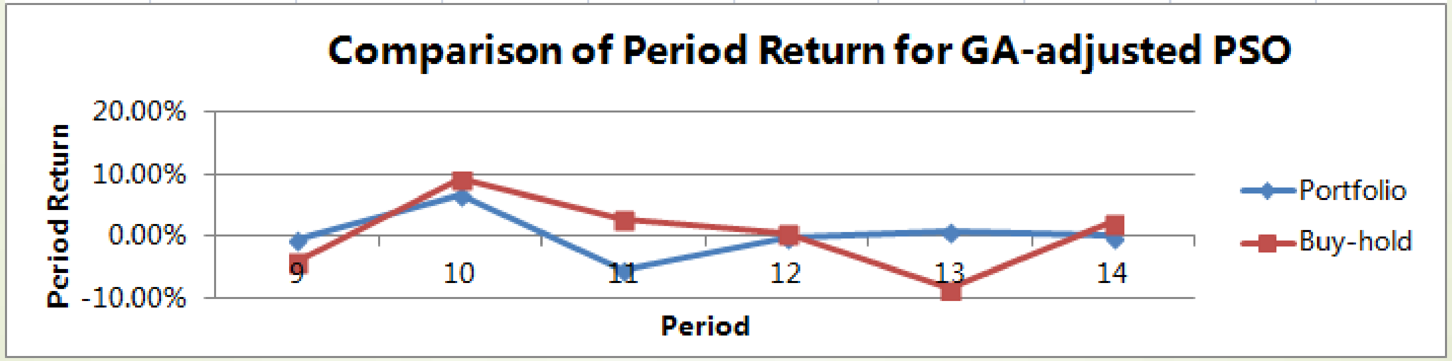
GA-PSO			
Testing period	Portfolio	Stock	Buy-hold
9	2.46%	1.54%	-4.03%
10	9.06%	2.14%	9.29%
11	0.49%	2.17%	2.83%
12	-1.46%	-1.22%	0.46%
13	0.18%	1.07%	-8.53%
14	1.67%	-0.28%	2.39%

GA-adjusted PSO			
Testing period	Portfolio	Stock	Buy-hold
9	-0.46%	-0.31%	-4.03%
10	6.67%	-3.96%	9.29%
11	-5.38%	-0.62%	2.83%
12	-0.11%	-0.42%	0.46%
13	0.77%	1.54%	-8.53%
14	0.09%	4.43%	2.39%

Portfolio receives an obviously better result than with single stock.


Comparison of Period Return for GA
Period Return
20.00%
0.00%
-20.00%
9
10
11
12
13
14
Period
Portfolio
Buy-hold


Comparison of Period Return for GA-PSO
Period Return
20.00%
10.00%
0.00%
-10.00%
9
10
11
12
13
14
Period
Portfolio
Buy-hold


Comparison of Period Return for GA-adjusted PSO
Period Return
20.00%
10.00%
0.00%
-10.00%
9
10
11
12
13
14
Period
Portfolio
Buy-hold

Why does neither PSO perform as good as GA？

PSO and adjusted PSO try to achieve improvement in every step. In this way, strategies with same indices become more similar. In this way, they lose their variety in some degree. They have less choice for the next generation.

How to Measure Diversity?

Phenotypes (ptype) The number of unique fitness values in the population, divided by size of the population.

Standard deviation (stddev) The standard deviation of fitness values in a population:

$$stddev(P) = \sqrt{\frac{\sum_{i=1}^{N} (f_i - \overline{f})^2}{N-1}},$$

where N is the population size and f_i is fitness of the ith individual.

Genotypes (gtype) The sum of the *Hamming distances* between any two genotypes(individual strings). The Hamming distance between genotype u and v is defined as:

$$Hamming(u,v) = \sum_i |sgn(u[i] - v[i])|,$$

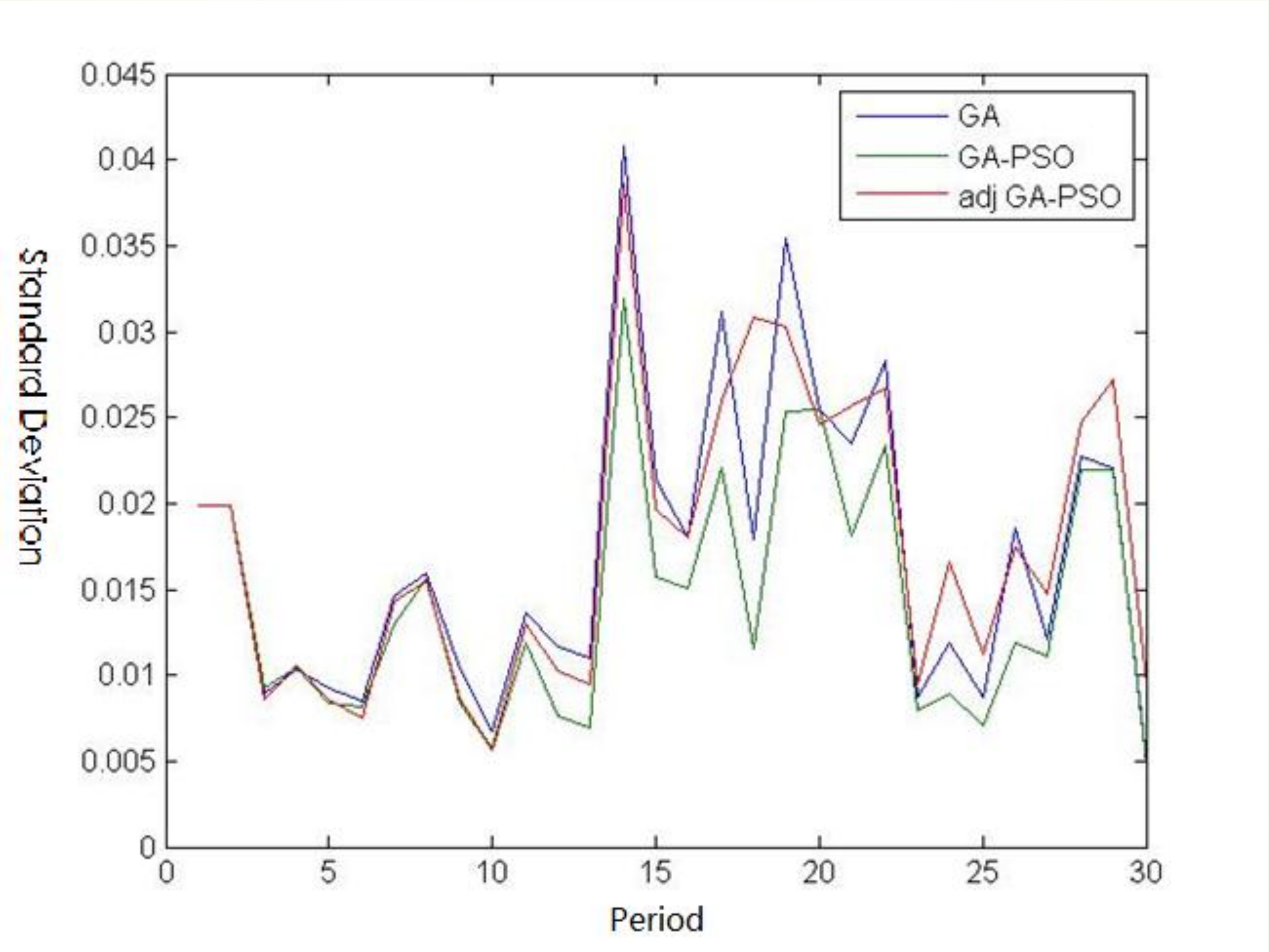
$$\mathrm{sgn}(u[i] - v[i]) = \begin{cases} \frac{1}{4} & \text{if } u[i] \neq v[i] \\ \frac{1}{4} \times \text{difference of index level} & \text{if } u[i] = v[i] \end{cases}$$

where $u[i]$ and $v[i]$ are the ith gene (integer element) of the u and v, respectively. And the Hamming based population diversity of population P is thus:

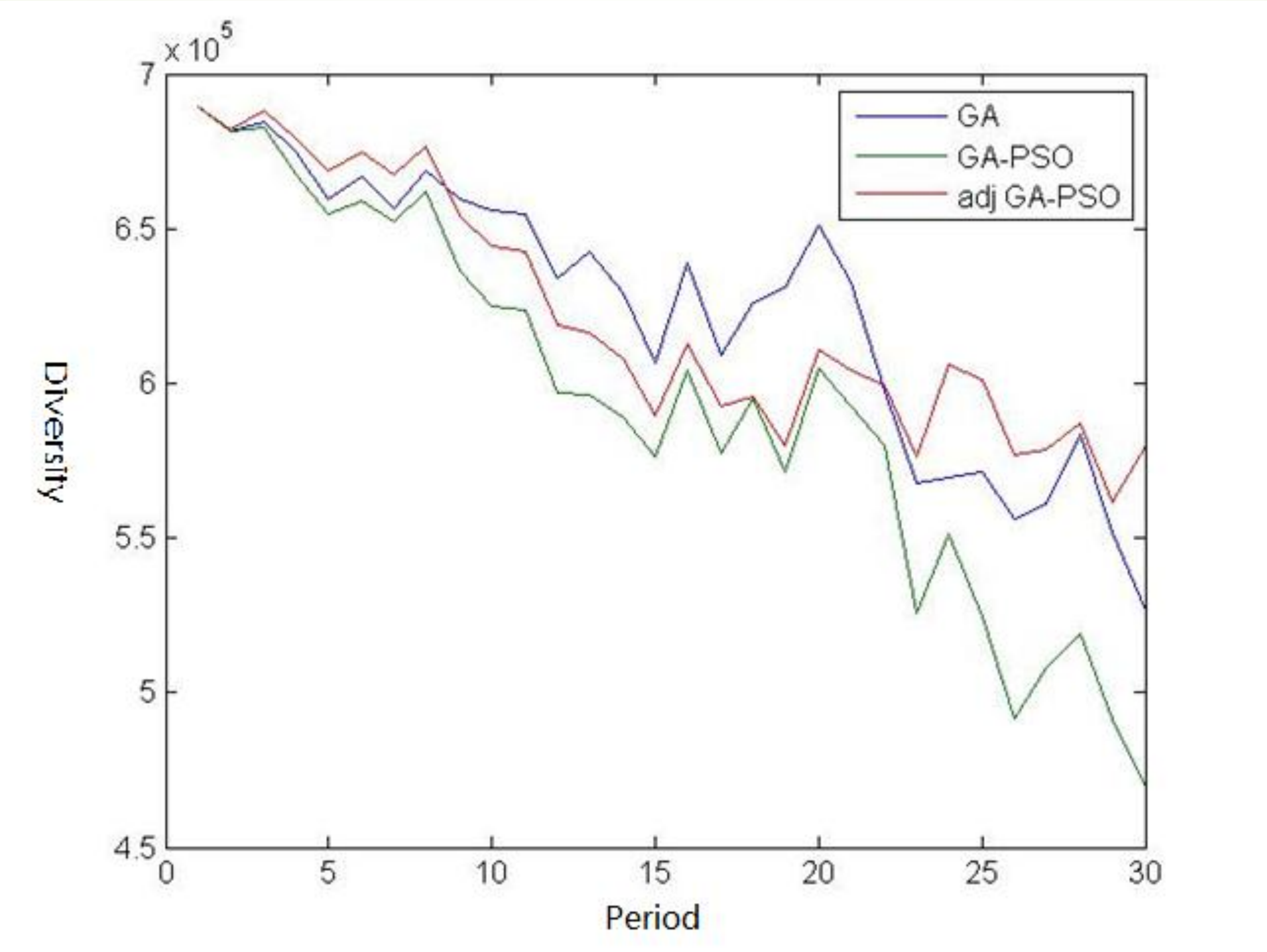
$$gtype(P) = \frac{1}{2} \sum_{i \neq j} Hamming(P[i], P[j])),$$

where $P[i]$ and $P[j]$ are the ith and jth genotypes in P. We normalize $gtype(P)$ by dividing it with $(N^2 K)/2$.

Phenotype Diversity



Genotype Diversity



Why Learning with Portfolio Performs Better Than That in Stocks ?

The difference in learning mechanism for single stock and for the portfolio:

Applying the same filter to all stocks in the portfolio allows to select stocks with more extreme values of the indicators.

On an upward trending market portfolio learning picks up only the more profitable stocks.

Measures of Risk Adjusted Returns

Time Period	9	10	11	12	13	14
Returns(every period)	5.62%	15.49%	-1.75%	0.34%	1.08%	0.05%
Buy-hold Return	-4.03%	9.29%	2.83%	0.46%	-8.53%	2.39%
Volatility(every period)	11.42%	5.37%	6.08%	2.39%	0.49%	0.93%
Sharpe Ratio	0.4909	2.8831	-0.2903	0.1380	2.1593	0.0418
Information Ratio	1.0581	1.3591	-0.8068	-0.0691	2.6573	-0.7553
Omega Ratio	0.6426	2.6031	0.6980	1.0064	8.4286	1.3559
Max Drawdown	8.98%	3.63%	5.34%	4.44%	0.22%	1.05%
			Total Returns			21.62%
			Total Buy-hold Return			2.41%
			Average Volatility			4.45%
			Annualized Total Returns			31.23%
			Annualized Volatility			12.84%
			Average Sharpe Ratio			0.9038
			Average Information Ratio			0.5739
			Average Omega Ratio			2.4558
			Average Max Drawdown			3.94%

*Hourly returns are used.
**Many calculations of ratio are done by MATLAB built-in function

Suggestions for Further Improvements

- Combination of similar stocks can improve the amount of learning data. So we can research on how to create portfolios to better perform in GA-PSO.

- More indices can be included to create more strategies.

- Another function of ranking can be adjusted to create more aggressive or conservative strategies.

Reference

- [1] Kaplinskii, A. I. and Propoi, A. I., On stochastic approach to nonlinear programming problems, *Avtom. Telemekh.*, 1970, no. 3, pp. 122-133.
- [2] Kaplinskii, A. I. and Propoi, A. I., First-Order Nonlocal Optimization Methods based on Potential Theory, *Avtom. Telemekh.*, 1994, no. 7, pp. 94-103.
- [3]Kaplinskii, A. I. and Propoi, A. I., Second-Order Refinement Conditions based on Potential Theory, *Avtom. Telemekh.*, 1994, no. 8, pp. 104-113.
- [4] Kaplinskii, A. I. and Propoi, A. I., Design of Nonlocal Optimization Algorithms: A Variational Approach, *Preprint of Inst. of Syst. Res.*, 1986.
- [5] Tarek A. El-Mihoub, Adrian A. Hopgood, Lars Nolle, Alan Battersby ,Hybrid genetic algorithms:A review, Engineering Letters, 13:2, EL_13_2_11
- [6] Daan Wierstra, Tom Schaul, Jan Peters, Juergen Schmidhuber, Natural evolution strategies, Engineering Letters, 13:2, EL_13_2_11
- [7] Kenny Q. Zhu, Population Diversity in Genetic Algorithm for Vehicle Routing Problem with Time Windows

THE END

Appendix

More about GA,PSO, black box optimization and their implementation in Matlab

- Black Box Optimization:

  In science and engineering, a black box is a device, system or object which can be viewed in terms of its input, output and transfer characteristics without any knowledge of its internal workings. Optimization algorithms that only use values of the objective function at certain points (and no derivatives) can be referred as black box methods. Such algorithms can work with algorithmically defined, non smooth and multi modal functions. Another example of black box function is a third –party software which should not be changed.

- Optimization Hybrids using Matlab functions:

  For example, consider Matlab (Financial toolbox) function **fvfix(**Rate, NumPeriods, Payment, PresentVal, Due) which computes future value of fixed cash flows. It can be viewed as black box function, so its output should be controlled by input parameters: given a desired level of future value, find such values of Rate, NumPeriods, Payment, PresentVal and Due that this function value is attained. Here we have 5 dimensional problem that requires an optimization algorithms such as Nelder Mead (**fminsearch()** function in core Matlab)or coordinate descent. In many cases only one parameter is sought and other parameters are fixed: for example, given future value of the cash flows, find their present value. In this case one dimensional search can be used, such as bisection.

More about GA,PSO, black box optimization and their implementation in Matlab

- Typical problems of financial engineering (calibration, pricing, hedging) can be reformulated as problems of global optimization, where objective function is a black box.
- Adaptive Systems:

  Our approach allows to create adaptive systems, where the selection of the optimization method is determined by the input data.
- Efficient Hybrids:

  The development of our GA algorithm was done under a novel framework of deriving global optimization methods based on the approach similar to smooth optimization [1]. Instead of the derivative of the objective function, the derivative of the potential function is computed at each iteration step. The approach is based on the randomization of the original problem. It can be shown that the parameters of the algorithm do not depend on the type of randomization. This approach allows to better understand the parameters of the optimization algorithms and can be used for constructing efficient hybrids based on available MATLAB modules, such as **fminsearch()** and **fzero()**.

Steps of generic PSO algorithm

1. Set $k = 1$ and evaluate $f(\mathbf{x}_j^k)$ for $j = 1, \ldots, M$. Set $pbest_j = +\infty$ for $j = 1, \ldots, M$.

2. If $f(\mathbf{x}_j^k) < pbest_j$ then set $\mathbf{p}_j = \mathbf{x}_j^k$ and $pbest_j = f(\mathbf{x}_j^k)$.

3. Update position and velocity of the j-th particle, with $j = 1, \ldots, M$, as

$$\mathbf{v}_j^{k+1} = w^{k+1}\mathbf{v}_j^k + \mathbf{U}_{\phi_1} \otimes (\mathbf{p}_j - \mathbf{x}_j^k) + \mathbf{U}_{\phi_2} \otimes (\mathbf{p}_{g(j)} - \mathbf{x}_j^k)$$
$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1}$$

   where $\mathbf{U}_{\phi_1}, \mathbf{U}_{\phi_2} \in \mathbb{R}^d$ and their components are uniformly randomly distributed in $[0, \phi_1]$ and $[0, \phi_2]$ respectively, the symbol $\otimes$ denotes component-wise product and $\mathbf{p}_{g(j)}$ is the best position in a neighborhood of the j-th particle.

4. If a convergence test is not satisfied then set $k = k + 1$ and go to 2.

- **Velocity update in PSO algorithm**
  - provides search directions
  - Includes deterministic and probabilistic parameters.
  - Combines effect of current motion, particle own memory, and swarm influence.

New velocity

$$\mathbf{v}_{k+1}^{i} = w\,\underbrace{\mathbf{v}_{k}^{i}}_{\text{current motion}} + c_1\, rand\, \underbrace{\frac{\left(\mathbf{p}^{i} - \mathbf{x}_{k}^{i}\right)}{\Delta t}}_{\text{particle memory influence}} + c_2\, rand\, \underbrace{\frac{\left(\mathbf{p}_{k}^{g} - \mathbf{x}_{k}^{i}\right)}{\Delta t}}_{\text{swarm influence}}$$

inertia factor

0.4 to 1.4

self confidence

1.5 to 2

swarm confidence