THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

EIE557

Computational Intelligence and its Applications

Lab 1 Report

Date: 20/4/2018

Laboratory 1: Optimization Using Genetic Algorithms

1.Objectives:

(1) To gain experience in solving optimization problems using GA's
(2) To be familiar with a GA optimization tool implemented in MATLAB

2. Exercises

2.1 Evolving Investment Strategies Using the GA

2.1.1 Introduction

A best rule will be found by using the simple genetic algorithm to make monthly decision on whether to make investment to the Hangseng Index (HSI) stocks. The decision rules will make use of the HK macroeconomic data and data from other markets in the previous month to decide the investment for the current month. Two strategies are implemented:

Investment Strategy 1:
A decision is made on whether to be in and out of the HSI stocks. We assume that the return for a month when we do not invest is zero by ignoring the bank interest. The fitness is calculated by accumulating the return of the 'investment' months.

Investment Strategy 2:
A decision is made on whether to long and short the HSI stocks. The fitness is calculated by a summation of the return of those 'long' months plus the summation of the return of those 'short' months. Note that an HSI gain/lose in a month will be a positive/negative return to a 'long" month and a negative/positive return for a 'short' month.

Table 2-1: Data sets to be used in constructing decision rules for investment strategies

Variable	Description	Data File	Remarks
1	Dow Jones Index	dj.mat	Monthly average of daily close prices
2	S&P 500 Index	s&p.mat	Monthly average of daily close prices
3	Nasdaq Index	nasdq.mat	Monthly average of daily close prices
4	Spot Gold Price	gold.mat	Monthly average of daily close prices
5	Japanese Yen/US dollar	jpy.mat	Monthly average of daily close prices
6	10-Year Treasury Bond Yield	10y-yeild.mat	Monthly average of daily close prices
7	HK Property Trading	property-trade-	Monthly data

	Volume	volume.mat	
8	HK Consumer Price Index	cpi.mat	Monthly data
9	HK M3 (money	money.mat	Monthly data
10	HK Saving	saving.mat	Monthly data
11	HK Number of Visitors	visitors.mat	Monthly data
12	HK Retail Index	retail-index.mat	Monthly data
13	HK Trading (Export)	trade-export.mat	Monthly data
14	HK Unemployment Rate	unemployment-rate.mat	Monthly data
15	Monthly Change of Variable 1	dj_c.mat	
16	Monthly Change of Variable 2	s&p_c.mat	
17	Monthly Change of Variable 3	nasdq_c.mat	
18	Monthly Change of Variable 4	gold_c.mat	
19	Monthly Change of Variable 5	jpy_c.mat	
20	Monthly Change of Variable 6	10y-yeild_c.mat	

Table 2-1 shows Hong Kong macroeconomic data sets and the data from the other markets that may be used for building up decision rules. A combination of three variables will be used to construct the decision rules using the GA. Example decision rules look like:

Investment Strategy 1:
IF Variable i is LESS THAN X, AND Variable j is GREATER THAN OR EQUAL TO Y, AND Variable k is LESS THAN Z, THEN invest in common stocks ELSE keep the cash.

Investment Strategy 2:
IF Variable i is LESS THAN X, AND Variable j is GREATER THAN OR EQUAL TO Y, AND Variable k is LESS THAN Z, THEN long common stocks ELSE short the common stocks.

In the implementation, 32 cutoff values are used for each variable. Eight logical groupings and eight combinations of “LESS THAN” and “GREATER THAN OR EQUAL TO” for three variables are used. Therefore, a solution chromosome will be a 21-bit binary string.

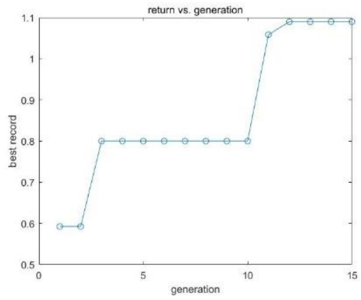
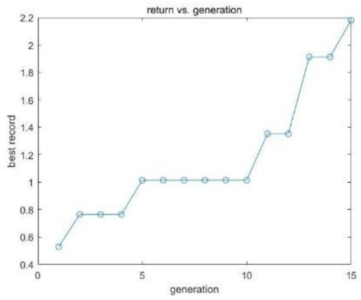
2.1.2 Problems

(1) Use a combination of three variables to run the GA to find the best rule. Some possible combinations are (a) Variables 1, 4, 9; (b) Variables 2, 4, and 9; (c) Variables 3, 4, and 9; (d) Variables 7, 8, and 9, and (e) Variables 9, 15, 18. A suggested parameter set for the GA is given in Table 2-2. Record the best rule that produces the best return for each of two investment strategies. Show the return vs. generation graph.

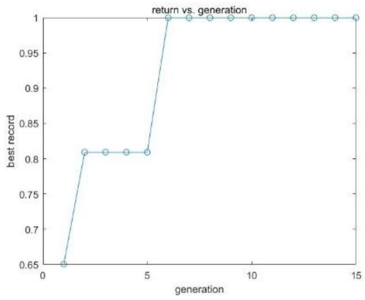
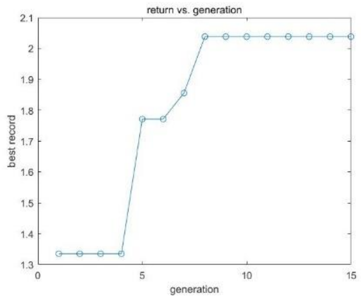
Table 2-2: Suggested parameters for running the GA

Population Size	No. of Generations	Selection Function	Probability of the simple crossover	Probability of binary mutation
10	15	Roulette wheel select	0.6	0.05

(a) Variables 1, 4, 9

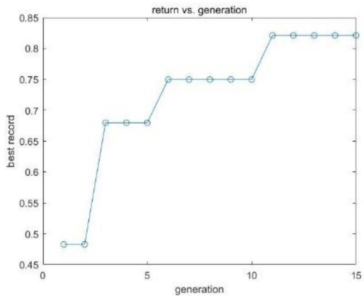
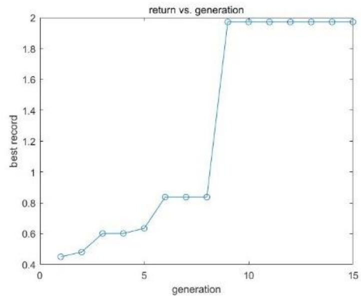
	Strategy 1	Strategy 2
best_rule	1001111111001010111101	1001101111101010001111
final_gain_train	1.0898	2.1792
final_gain_test	1.0402	1.0319
return vs. generation graph		

(b) Variables 2, 4, 9

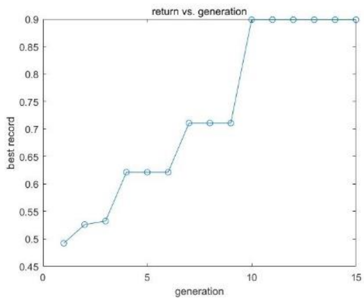
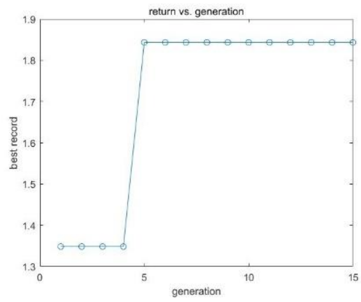
	Strategy 1	Strategy 2
best_rule	1110101010011100100101	1110010000010000000101
final_gain_train	0.9997	2.0380
final_gain_test	0.9747	1.0231
return vs. generation graph		

(c) Variables 3, 4, 9

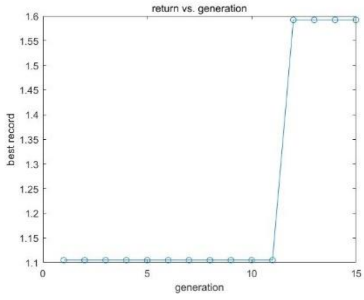
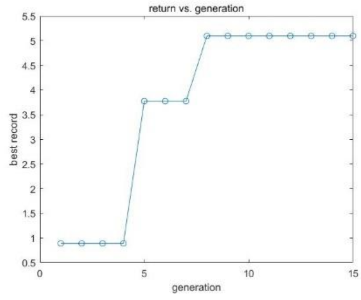
	Strategy 1	Strategy 2
best_rule	0010110001010011111101	0001001010000100100001
final_gain_train	0.8211	1.9711
final_gain_test	1.0366	0.9938

return vs. generation graph		

(d) Variables 7, 8, 9

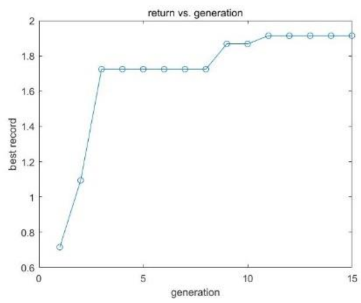
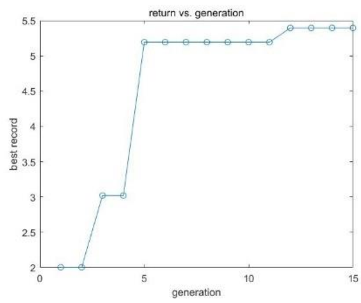
	Strategy 1	Strategy 2
best_rule	110100101100010010111	111000101100010100111
final_gain_train	0.8988	1.8434
final_gain_test	0.9906	1.0249
return vs. generation graph		

(e) Variables 9, 15, 18

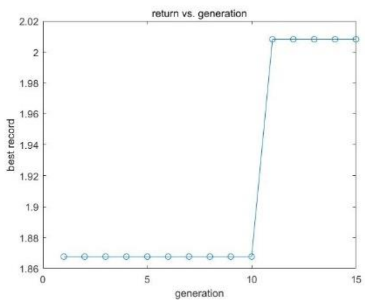
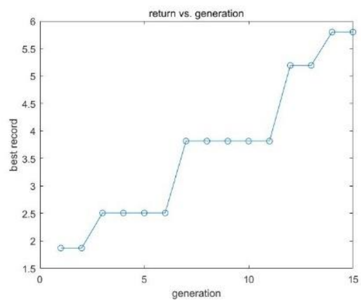
		Strategy 1	Strategy 2
best_rule		010010110001110100110	000010110010011001111
final_gain_train		1.5921	5.0946
final_gain_test		0.9767	0.8843
return vs. generation graph			

(f) Variables 15, 16, 17

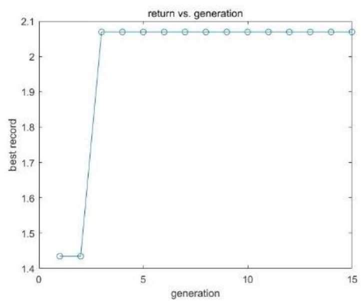
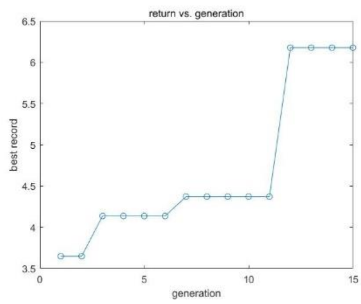
	Strategy 1	Strategy 2
best_rule	110010000101010110101	110000000101010011001
final_gain_train	1.9137	5.3952
final_gain_test	1.1036	1.2906

return vs. generation graph		

(g) Variables 15, 17, 20

	Strategy 1	Strategy 2
best_rule	01011010101001110011 1	010110101010000100111
final_gain_train	2.0082	5.8000
final_gain_test	1.0981	1.2778
return vs. generation graph		

(h) Variables 16, 17, 20

	Strategy 1	Strategy 2
best_rule	011110100010010100111	101100101011110000110
final_gain_train	2.0691	6.1771
final_gain_test	1.0270	1.2778
return vs. generation graph		

Until now, [16, 17, 20] has the best results.

(2) Use the same three-variable combination as in (1) and change the GA parameters to run the GA. Use Table 2-3 to report the simulation results. Explain your findings. Use graphs and other data if necessary.

Table 2-3: Experimental results for the combination of three variables you have chosen (the gain of the buy-andhold strategy is 26% for the training period of 01/2000 – 12/2003 and -6% for the test period of 01/2004 – 06/2005).

Simulation run	GA parameters used including no. of generation, population size, selection function, probabilities for crossover and mutation	The best rule obtained	Gain (lose) in the training period (01/2000 – 12/2003)		Gain (lose) in the test period (01/2004 – 06/2005)	
			Strategy 1	Strategy 2	Strategy 1	Strategy 2
1	no. of generation = 75 (Other values unchanged)	11111010101 1110100110 0101101010 11110100010	2.0955	6.1771	1.1447	1.4056
2	population size = 50 (Other values unchanged)	1101001010 11111000110 01110010001 0011100111	2.0955	6.3101	1.0981	1.1207
3	selection function = tournSelect (Other values unchanged)	11110010101 1111100110 1100001010 11111100110	2.0955	6.1771	1.1492	1.2778
4	probabilities for mutation = 0.9 (Other values unchanged)	0110101010 01100100111 11101010101 1110010110	2.0082	6.1771	1.0981	1.4056
5	probabilities for crossover = 0.25 (Other values unchanged)	0110101010 01111100111 0001001010 10011100110	2.0082	5.1930	1.0981	1.2778

(3) Use different combinations of three variables to go through the GA simulation in order to find the best combination of three variables. You may fix the GA parameter set. Report and discuss your findings. Use Table 2-4 to report the experimental results.

Table 2-4: Experimental results for different three-variable combinations

Simulation run	GA parameters used including no. of generation, population size, selection function, probabilities for crossover and mutation	The best rule obtained	Gain (lose) in the training period (01/2000 – 12/2003)		Gain (lose) in the test period (01/2004 – 06/2005)	
			Strategy 1	Strategy 2	Strategy 1	Strategy 2
1 Variables: 16, 17, 20	no. of generation = 30 population size = 30 selection function = tournSelect	0100001010 11111100010 0000101010 11111000010	2.0955	6.4115	1.1574	1.4308
2 Variables: 15, 17, 20	same as above	0000101010 11110100010 0010101010 11110100010	2.0955	6.1771	1.1529	1.4056
3 Variables: 15, 16, 17	same as above	0010101100 0101001001 1 01110100010 1010010101	1.9875	5.8522	1.0981	1.2778
4 Variables: 9, 15, 18	same as above	0000101100 10100001111 0001001100 10011001111	1.8945	5.0946	0.9134	0.8843
5 Variables: 1, 4, 9	same as above	10011100100 1101000100 10011101110 1101100100	1.1093	2.4733	0.9809	1.1057

(4) Explain the simulation results in the context of investment performance and discuss whether the strategies could work in real situations.

The results showed that the following three variables are most relevant to HSI stocks: (16) Monthly Change of S&P 500 Index, (17) Monthly Change of Nasdaq Index and (20) Monthly Change of 10-Year Treasury Bond Yield.
The strategies couldn't work in real situations because this module is based on the relationship between HSI stocks and other indices in the same period. If you want to predict future HSI stocks you need future other indices. It can only be used as a model for checking past values.

2.2 Optimization of Single-Variable Functions
2.2.1 Introduction

The function to be maximized is $f(x) = xsin(10\pi \cdot x) + 2.0$ ($x \in [1,2]$).
Use a binary string to represent a solution. Assume that a precision of 10-5 should be achieved (19-bit strings should be used).

2.2.2 Problems
(1) Decide a fitness function to be used.

The function $f(x) = xsin(10\pi \cdot x) + 2.0$ ($x \in [1,2]$) *itself can be a fitness function.*

(2) Run the GA using different GA parameter setting to find the best solutions. Use Table 2-5 to report the simulation results.

Table 2-5: Experimental results for the optimization of the single-variable function

Simulation run	GA parameters used including no. of generation, population size, selection function, probabilities for crossover and mutation	The best variable value	The function value
1	no. of generation = 15, population size = 10, selection function = normGeomSelect, probabilities for crossover and mutation = 0.6 and 0.05	1.8505	3.8503
2	no. of generation = 30, population size = 20, selection function = normGeomSelect, probabilities for crossover and mutation = 0.6 and 0.05	1.8506	3.8503
3	no. of generation = 30, population size = 20, selection function = roulette, probabilities for crossover and mutation = 0.6 and 0.05	1.8505	3.8503
4	no. of generation = 30, population size = 20, selection function = tournSelect, probabilities for crossover and mutation = 0.6 and 0.05	1.8506	3.8503
5	no. of generation = 30, population size = 20,	1.8506	3.8503

	selection function = normGeomSelect, probabilities for crossover and mutation = 0.9 and 0.05		

(3) Is a real-value encoded chromosome suitable for this problem?

Yes. For the continuous function optimization problem, real-value encoded chromosome can avoids the instability caused by the discontinuities binary codes.

2.3 Multi-Variable Functions

2.3.1 Introduction

The function to be maximized is: $f(x_1, x_2, \ldots, x_{10}) = (x_1 x_2 x_3 x_4 x_5)/(x_6 x_7 x_8 x_9 x_{10})$ where $x_i \in [1,10], i = 1,2, \ldots, 10$.

2.3.2 Problems

Use 10-gene real-value coded chromosomes and parameters shown in Table 2-5 to conduct the simulation. Set the number of generation to 200 and the population size to 30. Run the GA with different selection, crossover and mutation functions and different parameters. Use Table 2-6 to report the simulation results. Discuss your findings.

Table 2-5: Suggested parameters used for real-value coded multi-variable function optimization

Name	Parameters
Uniform mutation	5 (the number of mutations of the type to be carried out in each generation with randomly selected parents)
Boundary mutation	5 (as the above)
Simple crossover	5 (the number of crossover operations of the type to be carried out in each generation with randomly selected parents)
Arithmetic crossover	5 (as the above)
Normalized geometric selection	0.08 (the probability for the best individual)
Tournament selection	5 (the tournament size)

Table 2-6: Experimental results for the optimization of the multi-variable function

Simulation run	GA operators (selection, crossover, and mutation functions) and parameters used	Best variable values	The function value
1	normGeomSelect, simpleXover+arithXover, boundaryMutation+unifMutation	10 10 10 10 10 1 1 1 1 1	100000
2	normGeomSelect, arithXover, unifMutation	9.8539 9.7745 9.9545 9.8284 9.8862 1.0567 1.4658 1.0911 1.0367 1.0225	5.2002e+04
3	normGeomSelect, simpleXover, unifMutation	9.8665 9.4690 9.9954 9.8683 9.9050 1.0594	6.7815e+04

		1.0759 1.0218 1.0450 1.1060	
4	normGeomSelect, simpleXover, boundaryMutation	10 10 10 10 10 1 1 1 1 1	100000
5	tournSelect, simpleXover, unifMutation	9.9393 9.5881 9.6992 9.8239 9.9933 1.0154 1.1240 1.0110 1.0309 1.0798	7.0646e+04
6	tournSelect, arithXover, unifMutation	8.8396 9.4569 9.8748 9.9904 9.5687 1.0451 1.0162 1.0505 1.2217 1.0145	5.7067e+04
7	normGeomSelect, simpleXover+arithXover, unifMutation	9.7575 9.8792 9.8503 9.7909 9.8851 1.0227 1.0304 1.0954 1.0519 1.2281	6.1623e+04
8	normGeomSelect, simpleXover, boundaryMutation+unifMutation	10 10 10 10 10 1 1 1 1 1	100000
9	normGeomSelect, simpleXover, unifMutation no. of generation = 300	9.6915 9.8853 9.9913 9.9917 9.9589 1.0565 1.0083 1.0025 1.0177 1.0144	8.6399e+04
10	normGeomSelect, simpleXover, unifMutation population size = 90	9.8989 9.5501 9.8683 9.9725 9.9845 1.1355 1.2161 1.0180 1.0455 1.0728	5.8914e+04

As shown above, for selection methods, Normalized geometric selection and Tournament selection had similar results. For crossover methods, Simple crossover performed better than Arithmetic crossover. For mutation methods, Boundary mutation was much better than Uniform mutation. Besides, by increasing no. of generations and population size, this module could perform better, but there will be more calculations.