Computational Financial Modeling

ENHANCING TECHNICAL ANALYSIS WITH GENETIC ALGORITHM

SAIKIRAN |DEEPAK SHARMA | PRANJAL JAIN

23RD NOV. 2012

How Genetic Algorithm can be used to improve the performance of a particular trading rule ?

We need to optimize the parameters used by the trading rule

HOW TO DO THIS NOW?

Consider a Simple Moving Average system, commonly used in trading simulators, which has two set of parameters

The lengths of two moving averages

OUR MOTIVE

To investigate how Genetic Algorithms, a class of Algorithms in evolutionary computation can be used to improve the performance of a particular trading rule

How changes in the design of the GA itself can affect the solution quality obtained in context of technical trading system.

The GA

- Search and global optimization procedure
- Based on the natural biological evolution.
- Stochastic in nature
- Characterized by speed

ALGORITHM

```
Genetic Algorithm ()
{
  Generate initial selection operator
  Evaluate the fitness value
 While ( do not satisfy a stop condition)
{
    Execute the selection operator
    Execute the crossover operator
    Execute the mutation operator
    Evaluate the fitness value
 }
}
```

The Way GA Works

- Beginning with random generation
- GA goal is to improve the fitness of solutions as the generation passes by.
- Stopping Condition:
  - Until some fixed number of iterations or
  - Achievement of satisfactory fitness level.

SOME USEFUL TERMS

Crossover

If two parents are represented by a high level of fitness, then crossing them will lead to a better offspring (solution).

Mutation

As search space is large, some amount of randomness is maintained through generations by mutation operator applied infrequently.

Enhancing Moving Average System

A moving average for n days is given by:

$$\frac{1}{n}\sum_{i=0}^{n} P_{t-i}$$

$\theta 1 < \theta 2$

If $\quad \frac{1}{\theta_1}\sum_{i=0}^{\theta_1} P_{t-i} > \frac{1}{\theta_2}\sum_{i=0}^{\theta_2} P_{t-i} \quad$ $Buy\ signal$

Else $\quad$ Sell $signal$

CONTINUED…

- The lengths of MAs are chosen instinctively by the trader.
- If say we consider 500 days of data, then the number of possible ways of choosing lengths becomes 500x500

That makes the search space exponentially larger!

FOCUS

Optimizing our trading system means maximizing our Fitness Function such as Profit, given by

$$\boldsymbol{TR_f = \prod_{i=1}^{f}(1 + DR_i)}$$

$$\boldsymbol{DR_i = (P_i - P_{i-1}) \times \delta}$$

TR_f - Total return for the sample period

DR_i - Daily return for the day *i*,

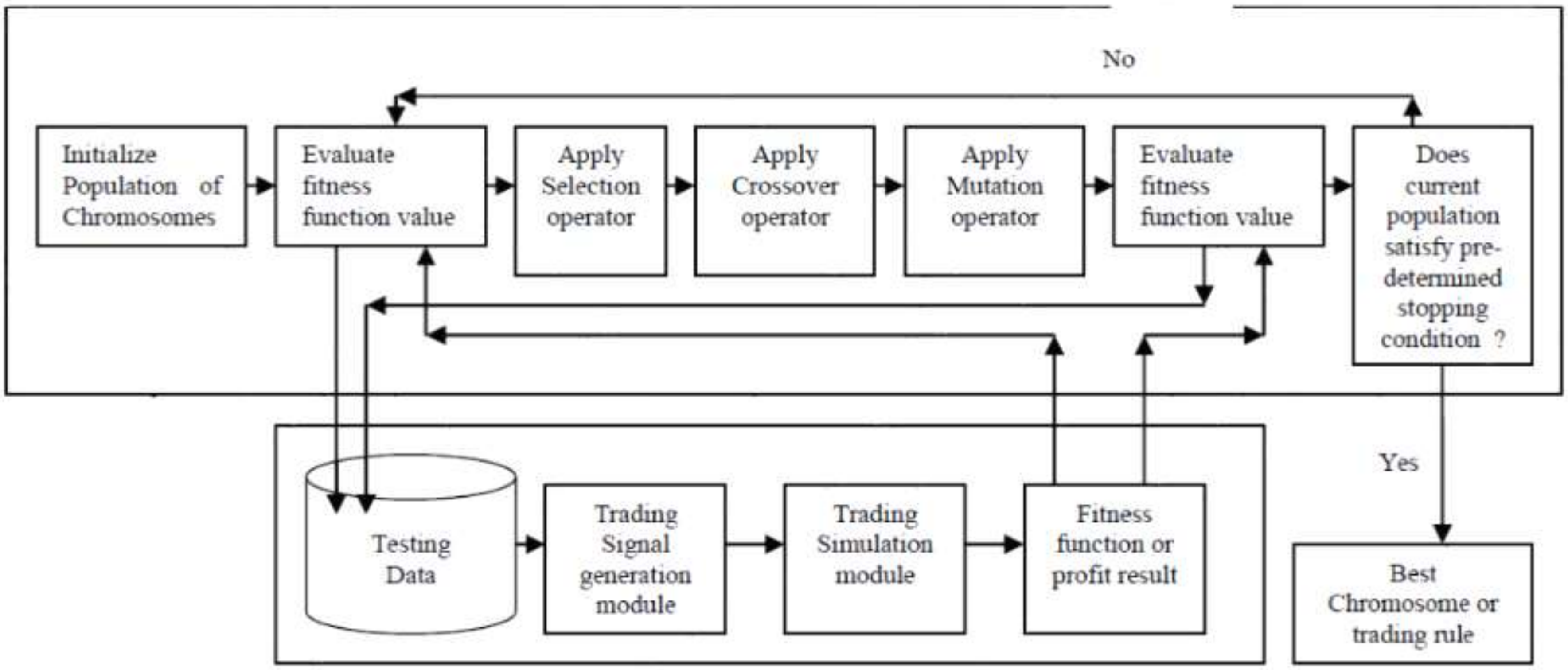
P_i denotes the stock price for the day *i*.

δ - dummy variable which generates value *1* for buy and -1 for sell

ENCODING

- Each parameter in Genetic Algorithm is encoded as binary string and concatenated to form a chromosome.
- Shorter moving average goes from 1 session to 256 session i.e. string length of 8 bits ex. 01100101]
- Longer moving average goes from 1 session to 512 session i.e. string length of 9 bits [ex. 110010101] Thus total search space is of 17 bits [ex. 01100101; 110010101] i.e. 217.

Simulation Process



I/P Used while Applying GA

We have used TCS data from National Stock Exchange from 1st July 2010 to 31st July 2012.

Implementation using R

Library used: genal

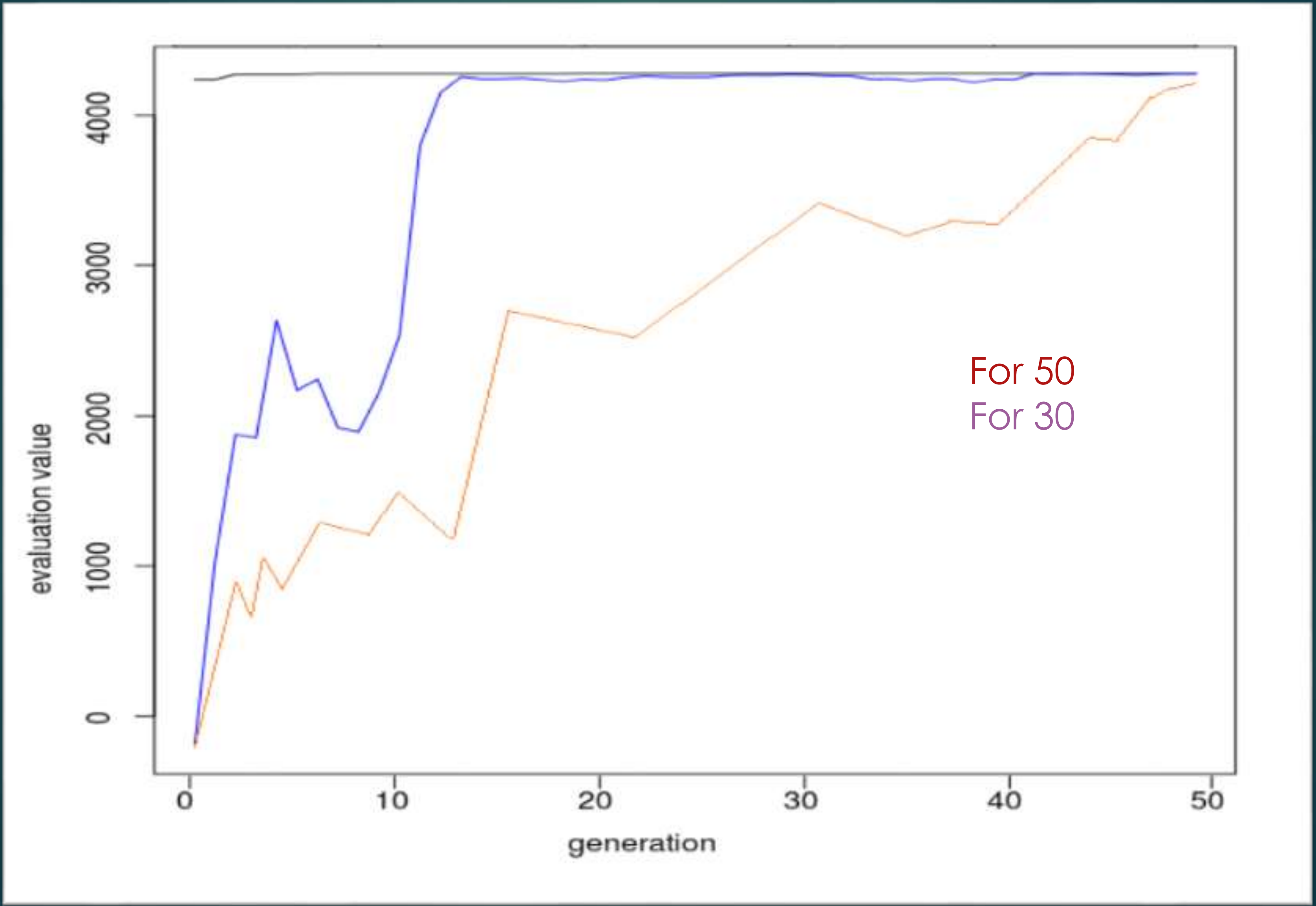
CODE:

```
avg <- function(data,min,max)
{
 sum <- 0;
 for(i in min:max)
{
  sum <- sum+data[i,]
 }
 sum <- sum/(max-min+1)
 return (sum);
}
```

```
evaluate <- function(string=c()) {
    returnVal = 0;
    if (length(string) == 2) {
      theta1 <- string[1];
      theta2 <- string[2];
      for(i in (1:(dim(data))){
        if(i > max(theta1,theta2)+1){
          delta <- 0
          avg1 <- avg(data,max(i-theta1,1),i)
          avg2 <- avg(data,max(i-theta2,1),i)
          if(avg1>avg2){
            delta = 1;
          }
          else{
            delta = -1;
          }
          returnVal <- returnVal+(delta*(data[i,]-data[i-1,]));
        }
      }
    }
    else {
      stop("Expecting a chromosome of length 2!");
    }
    return (returnVal+max((theta1-theta2)*1000,0));
  }
```

Maximum fitness function performance for GA applied for different population sizes

RESULT



evaluation value
0
1000
2000
3000
4000
0
10
20
30
40
50
generation
For 50
For 30

RESULT

Population Size	50	30
$\theta 1, \theta 2$	11,50	91,114
Max Profit	Rs. 217.65/-	Rs. 198.24/-
Max Return	127.48%	125.04%

Results given in paper

Genetic Algorithm: An Application to Technical Trading System Design by V. Kapoor, S.Dey, A.P. Khurana

Population size	**20**	**50**	**70**	**100**
$\theta 1/\theta 2$	235/414	4/40	12/173	9/212
Max. Profit	Rs. 834.35	Rs.932.84/-	Rs.936/-	Rs.947/-
Avg. Return	Rs. 709.46/-	Rs. 745.9/-	Rs. 732.14/-	Rs. 672.02/-
Max. Return	116.60%	141.27%	160.39%	150.38%
Std. dev. of Profits	83.029836	98.232199	99.66883	109.248246
Max. profit./ St.dev.	10.048797	9.4962752	9.3911005	8.66833139

INFERENCES

- As long as the population size is increased, the best possible solution obtained is higher.
- In case of average fitness, it increases after population size of 20, till 50 and then decreases, showing that the solution series becomes noisy in nature.
- Standard deviation is the measure of diversity in the population.
- Population size increase the solution quality.

CONCLUSION

- Genetic Algorithm performs better than the moving average lengths derived from rules in finance literature.
- To a great extent we are able to beat "Efficient Market Hypothesis"(EMH) (Any public information is reflected in the stock price and it is difficult to beat market.)

REFERENCES

- Genetic Algorithm: An Application to Technical Trading System Design (V. Kapoor, S.Dey, A.P. Khurana)
- Genetic Algorithm: Wikipedia
- NSE Site

THANK YOU
Deepak Sharma
IIT Mandi