Using Genetic Algorithm with Group Evolution Powered with Local Search for Finding Fibonacci Levels in Stock Market Prices

Naiem Khodabandehlooyeganeh[1] Pr. Carlo Lucas[2] Nima Darabi[3] Sirous Rezaii[4]

[1]Computer Engineering Group, Islamic Azad University Qazvin Branch
IAU Qazvin, Barajin Road, Qazvin, Iran
E-mail: Naiem.khodabandehloo@valiyacom.ir

[2]Faculty of Engineering, Tehran University, Tehran, Iran
E-mail: lucas @ipm.ac.ir

[3]Department of Mathematics, Sharif University of Technology, Tehran, Iran
E-mail: nima@cs.sharif.edu

[4]Department of Electrical Engineering, Khajeh Nasir University, Tehran, Iran
E-mail: sirousrezaee@mycom-int.com

Abstract

There are many issues that their main goal, is not just to find a global optimum point; In such issues, local optimum points are as important as global ones. For example; in stock diagrams there are levels known as Fibonacci Levels which are very important in technical analysis of stocks. The consequential point in this matter is to identify all possible local optimums in a domain as well as the global optimum. For solving this problem, we used Genetic Algorithm and we have made a change in the evolutional part of this algorithm which we will call Genetic Algorithm with Group Evolution. To increase the speed and efficiency of the algorithm we are using normal local search. Finally, we will show the benefits of our algorithm in finding all optimums in an acceptable time.

Keywords:

Genetic Algorithm, Group Evolution, Local Search, Fibonacci Levels

Introduction

There are many issues in the real world that have different optimum solutions in limited amplitude. Usually there is only one solution that is the global optimum.

Genetic Algorithm has been used in solving optimum-making issues as an effective way and has proven its efficiency. The main privilege of the Genetic Algorithm in compromise with other search and optimizing methods is that genetic algorithm does not stop in local optimum point and always keeps on searching for a better answer. Our method can be used in problems which main goal is to find all optimum points, instead of finding only the global one.

Finding the Fibonacci levels in stock diagrams is one of these problems in which we are not looking for one specific optimum point but all the interesting local optimum points. Since the Fibonacci levels are basis of many financial technical analyses, finding these levels in automatic technical analysis systems are very useful. Usually, along the diagram, different levels will appear each one describing the stock's cost behavior in a specific period of time.

Since the levels are mapped between two different points in a diagram, our Fitness Function amplitude is 3D and limited. Because ordinary search method will take a lot of energy for this kind of problems, accidental search methods are commonly used.

Using genetic algorithm along with local search is a method that has been used frequently [4]. By doing this, we can use the speed and efficiency of local search as well as preserving the advantages of Genetic Algorithm by finding a fast and suitable way to search locally around a specific point.

Group Evolution is the change we have made in Genetic Algorithm. In this case we can say we are breaking one Genetic Algorithm issue to a few sub issues. We can also eliminate the limitations caused by population decrease, using local search.

Approaches and Methods

Genetic Algorithm Powered with Local Search:

Local search has been used to solve optimization problems from long ago. The search usually starts with one point and keeps going while finding a better solution than the previous one. There is a variety of local search methods. The main problem was the need to know the system's process function and the bigger problem was that almost all local search methods were stuck in local optimum points.
Accidental Evolutional search methods have solved many search problems. These methods would not get stuck in the local optimum points and also there is no need to know the system's process functions. Even in Multifunctional Evolutional methods [3], we can do the optimum point finding, for different functions, simultaneously.
Accidental and Evolutional search methods might have slow and inaccurate functionality. To overcome this problem, mixing them with local search methods is an appropriate suggestion; we will also use this method in our paper.
While mixing Genetic Algorithm with Local Search, we have to notice to some points to gain the optimum solution; some of them are mentioned in [6]: first doing the local search only on chosen points, not on all of them; and second stopping local search before all points near the chosen one are being searched. These two tips have major effects on algorithms speed.
However we have not paid too much attention on optimizing these two algorithms and to ease the issue we have used a simple recursive local search in calculating the fitness function, in a way that the best neighbor points from north, south, east, and west, changes place with the point and this will repeat until the best neighbor is not recognizable.

Fitness Function

In fact, the fitness function is our goal deal. The population individuals are points placed on this function. Fitness function has been described, about Fibonacci levels, in this way:

1- We consider a price diagram of a stock in a specific period of time (figure 1- Iran Khodro stock price diagram).
2- We take two price levels in a way that in that special area most of the prices are formed around 0%, 16%, 31%, 50%, 62%, 100%, and… levels.
3- There maybe many levels with these conditions, which the price analyzer will choose one of them, according to his needs.

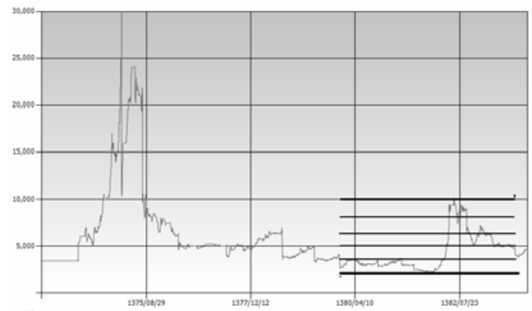

Figure 1 – Iran Khodro stock price and a Fibonacci level on it.

We noticed that Fibonacci Levels need two different price levels, to form. These two points can be practiced better on price histogram which shows status of aggregation of points on a level, in a better way.

$$X_i=\{x \mid f(x)=i\} \quad (1)$$

$$P(i)=n(Xi) \quad (2)$$

$$D(i)=\sum_{j=1}^{n}\prod_{k=j-m/2}^{j+m/2}\frac{m}{2(x_j-x_k)} \quad (3)$$

$$S(i)=\prod_{j=1}^{n}\frac{\sum_{k=j-m/2}^{k=j+m/2}(f(x_i)-f(x_i+k))}{f(x_i)} \quad (4)$$

$$\begin{aligned}F(i,j)=\,&\gamma D(i).S(i).P(i)+\gamma D(j).S(j).P(j)+\\&\delta(P(i+0.23(i-j)).P(i+0.38(i-j)).P(i+0.50(i-j))\\&.P(i+0.61(i-j)).P(i+1.61(i-j)).P(i+2.61(i-j))\\&.P(i+4.23(i-j)))\end{aligned} \quad (5)$$

In (1) Xi is a set insisting all x points which $f(x)$ is equal to i (all days that the stock price has been i). In (2) $n(Xi)$ gives the count of values in set Xi referred as n in later equations. In (3) m is a constant usually set as 15. It lets us to monitor the price behavior for a limited duration (e.g. 2 weeks). In (5) γ and δ are constants used to prioritize extent levels from inside levels in Fibonacci Levels.
In above formulas $f(x)$ function is considered as stock's graph. Histogram function for diagram is $P(i)$; it defines the number of points in the graph which their price is i. $D(i)$ is a measure for the

density of these *P(i)* points. *S(i)* is the sharpness function; it gets higher value if the diagram usually shapes local maximum or minimums for price *i*. This function is somehow similar to derivative.
F(i,j) is the final fitness function.

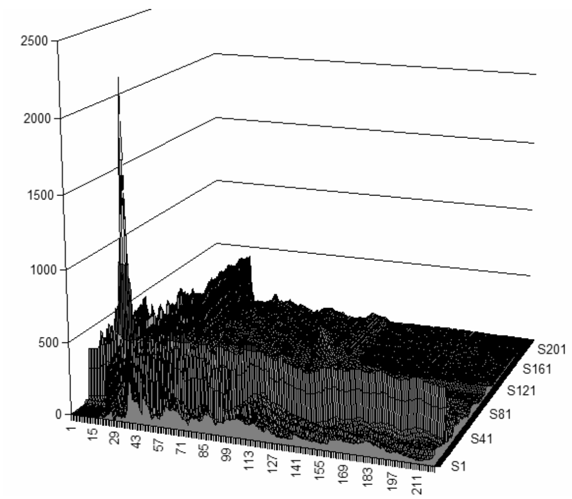

Figure 2 – Fitness function graph for Iran Khodro Company

Genetic Algorithm with Group Evolution for finding all local and global optimums

As it was mentioned before, in Genetic Algorithm, there is a primal society which evolves via generations using Mutation and Crossover processors. In each generation, individuals with higher fitness value have more chance to reproduce through evolution of generations. This leads society closer to optimum values. The algorithm won't stop searching by finding a local optimum and will always be in search of better results.
The steps of a genetic algorithm can be perused as below:

1- Primal society will be picked up accidentally within the problem domain.
2- Crossover processor will be applied to the society's individuals with a specific probability.
3- Mutation processor will be applied to the society's individuals with a specific probability.
4- The fitness value will be calculated for each individual.
5- The new society will be chosen, individuals with better fitness have more chance to survive. Selection can be done with a method like Round Robin.
6- The process will be repeated from step 2.

As it was mentioned before, by combining genetic algorithm with local search and calculating fitness for some specific points with better values using a local search method, we will get a better answer.
The above method is suitable for finding unique answers. Here, we suggest a novel way to find all local optimum points. In Genetic Algorithm with Group Evolution, the whole society should be divided to distinct primal groups first of all. Then we run normal genetic algorithm with local search up to specific numbers of generations. In fact in this situation, there have been a genetic algorithm with local search ran on each group while the whole society is shared for all groups but primal society is chosen from a limited part of society called a group. In this situation, the Genetic Algorithm will search for optimum points with higher probability in the group, but it can vast its search limits beyond that.
After passing a certain number of generations, groups will be arranged again. Our algorithm in this manner is similar to Tabu Search along with genetic algorithm [5]. It is noticeable that number of new groups does not have any relation to the old ones. New groups will be created around local optimum values which are called group leaders. In fact, number of local optimum points or group leaders in the one that elucidates the quantity and population of the groups. It is obvious that the more the groups are, their population is less and in such situation a limit must be set to avoid further problems for genetic algorithm. In situations which the number of local optimum values is too much, size of society must be greater than ordinary situations. Also, it is better to set limited amplitude; otherwise there should be a limitation on number of possible groups.
When new groups have been shaped, new query will begin and the genetic algorithms will act recently shaped groups as primal societies. At the end, after generations of evolutions within groups, many group leaders (optimum points) will be discovered.
Genetic algorithm with group evolution can be described as below:

1- Primal society will be chosen accidentally within the whole domain.

2- The primal society will be divided to a specific number of groups as primal groups.
3- Genetic algorithm with local search will be performed for each group until specific number of generations (search in all groups can be perform simultaneously)
4- Best points of each group will be found and if their advantages are noticeable, they can be group's leader.
5- The primal society will be divided among group leaders and each leader is granted a group bounded by half from neighbour leaders. Unavoidably the number of leaders must be limited.
6- The algorithm will continue from stage 3, for a definite numbers.

Results and Discussion

Regarding to Fitness Function in section 3, by using Genetic Algorithm with local search, we can find the optimum answer in Fibonacci Levels on limited domain of the problem, faster and easier than using the generic genetic algorithm.
In our problem, each Fibonacci level is described with two extents which are level 0 and level 100.
The genetic algorithm which was used in this problem is Aucsd-1.4 [1]. In solving this problem we have used the usual parameters of genetic algorithm. We used gray code to code our genome and two point crossover. We also used round robin method for choosing population.
Our genetic algorithm processors are binary processors using a common Genetic Algorithm way – two point Crossover on a binary string with one point Mutation. For avoiding continuous points and unexpected changes by Mutation and Crossover processors, and because using usual binary coding makes the importance of bits different from each other, we used Gray coding to perform genome calculations.
Length of used genome in this query depends on the expected accuracy in calculating levels. Normally, we divide prices into 2000 numbers between the maximum and minimum values; for more accuracy we can divide them in more pieces. Practically we found out that for most of the stocks with rather long histories. Least proper division for price amount or histogram function domain is 700 and this amount can be increased to 5000 for more accuracy; though amounts more than 5000 does not have any practical advantages or even may reduce accuracy of the answer. Concerning size of domain and expected accuracy, a 32 bit long genome has been used. The population of primal society was set to 16000 (including all groups).
The primal population has been divided to 16 different groups with same size domains so each group received around 1000 elements to search.
The utmost number of generations for each group in growth levels was 50 generations; with increasing the accuracy and length of local search, we can use less number of generations for each group.
Number of levels for group evolution was chosen as 10.
Crossover probability was 0.50 and Mutation probability was 0.05. Approximately high rate of mutation is for not limiting search domain to the groups.
Maximum number of groups was 49; this was because more Fibonacci Levels are not useful in diagrams and very weak Fibonacci Levels are not used in technical analyses. Also for 49 groups, population of each society will be 326 elements at average which is a low number for running genetic algorithm procedures and should not be less than this amount.

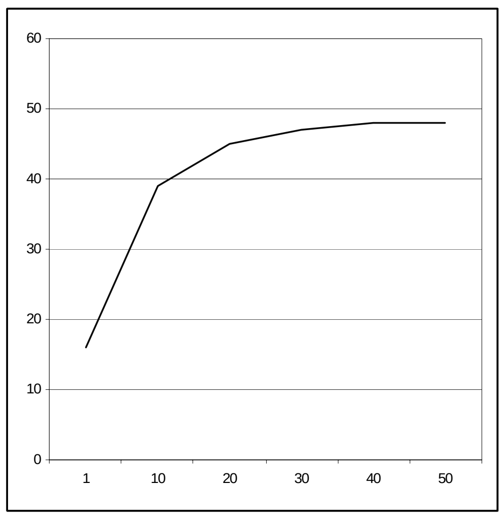

Figure 3 – Changes in number of groups (vertical) versus evolution of groups (horizontal)

In total, the number of all genetic algorithm generations is 500 Also we have to take group selection process into consideration. The group selection process selects maximum 49 elements with better fitness values. Of course if there are no better elements than those 49 (and by better we mean a lot better) number remains 49. Also if a few of better elements are in a square domain with 1.20 altitudes for each side of the square, only the best element will be chosen as leader. At end, the whole altitudes will be divided between group leaders by half of distance from the neighbor leaders.

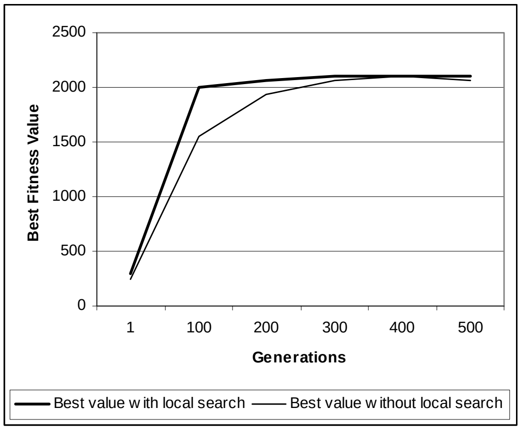

Figure 4 – Comparison of the best fitness value with and without local search

The Local Search process has been applied on algorithm in calculating the fitness function part, in a recursive way with reviewing four neighbors (east, west, north, and south) and with exchanging the recent elements with better one. This task should be kept on until no better neighbour is found.

Finally, performing the above algorithm on a Pentium 2000 Full cache system took less than 2 minutes.

Conclusion

Genetic algorithm and the methods inspired from the nature are having a lot of usage in different issues. Using these algorithms we can accidentally search in a vast domain and find proper answer. Finding Fibonacci Levels in stock diagram is an important issue; it is the basis of technical analysis. These levels have a lot of guidelines hidden in them such as Support and Defiance Levels, Eliot Waves, Correcting Waves, etc. which all shape around these levels.

To find all existing Fibonacci Levels in diagram we cannot use generic Genetic Algorithm; because generic Genetic Algorithm simple algorithms search for only one optimum value. Genetic Algorithm is the proper tool to find the right answer. At the end we suggested a way to extend the genetic algorithm so we can find a group of optimum points in a query while using all advantages of Genetic Algorithm. We called this method as Genetic Algorithm with Group Evolution.

To have a fast search feasible with low population of the society, we added local search to our algorithm.

We solved the query with these ways with founding the proper answer in a reasonable time.

Major problem of this method is that it needs a lot of society's elements. Method of dividing group domains between leaders is also challenging which is very important in accuracy and speed of finding the answer.

References

[1] - Schraudolph, N.N.; Grefenstette, J.J. 1991. A user's guide to GAucsd 1.4. UCSD

[2] –Brooks, C. 2001.: Artificial Intelligence Programming, Local Search and Genetic Algorithms, Technical Report, Department of Computer Science, University of San Francisco

[3] – Zitzler E.; Laumanns M.; Bleuler S. 2002. A Tutorial on Evolutionary Multiobjective Optimization, MOMH, Paris, France

[4] - Bouhmala N. 2004. Combining Local Search and Genetic Algorithms with the Multilevel Paradigm for the Traveling Salesman Problem, HG, Valencia, Spain

[5] - Mary Raja Slochanal S.; Sudhakaran M. 2003. Integrating Genetic Algorithms and Tabu Search For Combined Heat and Power Economic Dispatch, Tencon, Banglore, India

[6] – Ishibuchi H.; Yoshida T.; Murata T. 2003. Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms, *IEEE Transaction on Evolutionary Computation 7*.