Engineering Applications of Artificial Intelligence 74 (2018) 1–9

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Fibonacci indicator algorithm: A novel tool for complex optimization problems

A. Etminaniesfahani, A. Ghanbarzadeh *, Z. Marashi

Department of Mechanical Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran

ARTICLE INFO

Keywords:
Benchmark function
Mechanical benchmark problem
Fibonacci indicator
Stock market
Meta-heuristic algorithm
Number of function evaluation

ABSTRACT

In this paper a new meta-heuristic algorithm is introduced. This optimization algorithm is inspired by the very popular tool among the technical traders in the stock market called the Fibonacci Indicator. The Fibonacci Indicator uses to predict possible local maximum and minimum prices, and periods in which the price of a stock will experience a significant amount of movement. The proposed Fibonacci Indicator algorithm is validated on several Benchmark functions up to 100 dimensions to have a comparison to algorithms such as DE extensions, PSO extensions, ABC, ABC-PS, CS, MCS and GSA in the ability of convergence and finding the global optimum in different research areas. Finally two engineering design problems are used to show the performance of the algorithm. Application of the proposed Fibonacci Indicator Algorithm in a wide set of benchmark functions has asserted its capability to deal with difficult optimization problems.

1. Introduction

In the last few decades, the use of meta-heuristic algorithms has been much improved to approach the optimized solution of nonlinear functions. A heuristic algorithm is a method to find the solution to an optimization problem by "trial-and-error". How ever, these algorithms may not find the global best solution to the problem and might get trapped in the local optimum points. On the other hand, the meta-heuristic algorithms finds the optimum solution by higher-level strategies employing trial-and-error, exploration, and exploitations. Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995), Evolutionary Algorithms (EA) including Genetic Algorithm (GA) (Holland, 1975), Ant Colony Optimization (ACO) (Bilchev and Parmee, 1995; Dorigo and Blum, 2005), and the Bee Algorithm (BA) (Pham et al., 2006) are among the most popular metaheuristic algorithms. Evolutionary algorithms and swarm intelligence-based algorithms are two main categories of population-based optimization (Karaboga and Akay, 2009). Genetic algorithms, Differential Evolution (DE) (Storm and Price, 1995), (Meng and Pan, 2016; Meng et al., 2018) and evolutionary strategy (ES) (Rechenberg, 1965; Schwefel, 1965) have been the most popular techniques in evolutionary computation. Particle swarm optimization and Bees Algorithm are the most popular examples of swarm intelligence optimization. Two advantages of the different categories, i.e. evolutionary algorithms and swarm intelligence-based algorithms are presented below:

1. Particularly useful in multi-modal and multi-objective optimization problems;
2. Hybridize algorithms to each other;

A number of optimization algorithms can combine with each other and produce the hybrid algorithm with the synergy of both algorithm's advantages and elimination of their disadvantages.

Global optimization can be applied to various branches of science, economics and engineering (Bomze et al., 1997; Gergel, 1997; Horst and Tuy, 1996; Li et al., 2015; Rizk-Allah et al., 2016, 2018). Generally, solving nonlinear optimization problems can be classified into deterministic and stochastic methods (Li et al., 2015; Arora et al., 1995; Pardalos et al., 2000; Younis and Dong, 2010). In deterministic methods, optimization problems are solved by creating deterministic progression of convergence at the global optimal solution. This method requires unfailing mathematical specification and responsively depends on the initial conditions. On the contrary, in the stochastic methods including heuristic and meta-heuristic methods, new points are randomly generated (Younis and Dong, 2010). The efficiency of the optimization algorithms is usually determined by their ability in finding the global best solution by the minimum cost usually corresponding to the number of function evaluations. Exploration and exploitation are two main strategies to find the global best solution. Poor exploring and very fast convergence of algorithms increase the chance of getting trapped in local minima. Furthermore, very slow converging and increasing function evaluations

* Corresponding author.
E-mail address: Ghanbarzadeh.A@scu.ac.ir (A. Ghanbarzadeh).

https://doi.org/10.1016/j.engappai.2018.04.012
Received 22 May 2017; Received in revised form 8 April 2018; Accepted 10 April 2018

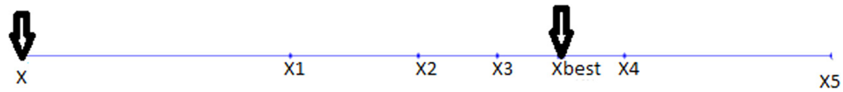

Fig. 1. Generating new points using FI method.

is not economical. The balance between exploration and exploitation is crucial to improve the efficiency of optimization algorithms (Li et al., 2015). Over the past decade, meta-heuristic algorithms such as GA (Price et al., 2005; Yang et al., 2007; Chelouah and Siarry, 2000) ACO, PSO (Jiang et al., 2007) and the artificial bee colony (ABC) have shown considerable successes in optimization algorithms (Ghanbari and Rhati, 2017). Previous researches show that ABC and GA have better exploration and slower convergence. However, ACO and PSO converge faster with more possibility of getting trapped in local optima (Alshamlan et al., 2015; Premalatha and Natarajan, 2009; Fidanova et al., 2014; Meng and Pan, 2016). A nonlinear optimization problem can be formulated as a D-dimensional problem of the following type: (Gergel, 1997; Nguyen et al., 2014).

$$f(x) = \begin{cases} \min & f(x) \\ s.t. & 1 \le x \le u \end{cases} \tag{1}$$

The objective function is defined by $f(x)$ and the D dimensional vector of variables is $x = (x1, x2, \dots, xD)$; lower and upper limits of variables are defined by $l = (l1, l2, \dots, lD)$ and $u = (u1, u2, \dots, uD)$.

2. Fibonacci indicator in the stock market

2.1. Fibonacci ratios

Leonardo Fibonacci is an Italian mathematician who found a sequence in 12–13th century. In this sequence, each number is generated by summing two previous numbers as follows: $\{0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots\}$. In this sequence, each number is approximately 61.8% greater than the preceding number. By dividing one number in the sequence by the two and three places to the right, 38.2% and 23.6% will be found, respectively. Eqs. (2)–(7) are used to generate Fibonacci ratios.

$$F(1)/F(2) = 0 \tag{2}$$

$$F(2)/F(3) = 100\% \tag{3}$$

$$F(3)/F(4) = 50\% \tag{4}$$

$$\lim_{n\to\infty} (F(N)/F(N+1)) = 61.8\% \tag{5}$$

$$\lim_{n\to\infty} (F(N)/F(N+2)) = 38.2\% \tag{6}$$

$$\lim_{n\to\infty} (F(N)/F(N+3)) = 23.6\% \tag{7}$$

2.2. Fibonacci retracement and Fibonacci time zone

Fibonacci retracement and Fibonacci time zone are two well-known indicators used in finance technical analysis to predict possible maximum and minimum price of each stock and suggests suitable time to buy or sell in the future. Fibonacci retracement predicts the possible local minimum and maximum price of each stock by taking the minimum and maximum points on a chart and dividing the vertical distances proportional to Fibonacci ratios. For example assume m and M are minimum and maximum price of a specific stock, respectively.

$$\text{Fibonacci percentages} = [0, 23.6\%, 38.2\%, 50\%, 61.8\%, 100\%] \tag{8}$$

$$\text{Possible local minimum or maximum price } (i) = m + (M - m) \times \text{Fibonacci percentages } (i), i = 1, 2, \dots, 6 \tag{9}$$

Fibonacci time zone in summary predicts the periods in which the price of a specific stock will get a significant amount of movement by choosing the starting point and dividing the horizontal axis corresponding to Fibonacci series.

3. Fibonacci indicator algorithm

Here, we present a novel evolutionary optimization algorithm using a combination of two popular tools of technical traders in the stock marketing, Fibonacci retracement and Fibonacci time zone. Assume Fibonacci ratios are on arbitrary axis as shown in Fig. 1, so adding 50% to Fibonacci ratios arranges points more symmetrically around 100%. In this article, the act of generating new points proportional to Fibonacci series + 50% between the arbitrary point (x) and the independent variable of the best gained solution is named FI method. Fig. 1, shows FI method between x and x best.

$$\text{Fibonacci percentages} + 50\% = [50, 73.6, 88.2\%\ 100\%\ 111.8\%\ 150\%] \tag{10}$$

Assume the independent variable is time and the fitness of benchmark function is its price. In the real world, price chart of each stock is available and traders use Fibonacci indicators to determine the proper time to speculate in the stock markets by predicting local minimum and maximum prices.

In FIA the algorithm tries to find the minimum fitness of problem in variable intervals without using chart. The population size is the number of points that algorithm uses to start searching. Evaluating the benchmark cost functions in chosen points determines their fitness; all points save and sort based on fitness value. The best point sets as “xbest” and its fitness sets as “gbest”. However, in each step the data of “xbest” and “gbest” are shared among all traders.

3.1. Searching phase

Searching phase is the main part of Fibonacci indicator algorithm. In this phase algorithm generates new points between the first chosen points from the best to the worst fitness, respectively and the shared xbest. As soon as a better fitness is met, the worst point should replace with xbest. This procedure continues until all the first chosen points are renewed after replacing all of them with new founded points it is suggested to do not omit any xbest in iterations. After making decision about adding or replacing newly founded points, algorithm cares on objective function. In single variable problems new points are generated from the worst point (worst xbest in previous iterations) to the newest xbest but in multivariable problems this procedure should just carry out for the specific percent of iterations (p%). Also, in $(100-p)$% of iterations, one point (assume “x”) should generate by crossing over between xbest of previous iterations; it means each variable gains from xbest of one of the previous iterations randomly.

For example for a 3 dimensional function if m xbest saved, x generates by crossing over as you see below.

$$\text{xbest} = \text{xbest}^m = (\text{xbest}1^m, \text{xbest}2^m, \text{xbest}3^m) \tag{11}$$

$x = (\text{xbest}1^i, \text{xbest}2^j, \text{xbest}3^k)$ and i, j, k are integer numbers randomly chosen from 1 to m.

New points are generated between “x” and xbest by FI method. Crossing over and using xbest of previous iterations from the worst to the best fitness helps the algorithm to do not trap in local optimums. In both case of single variable problems and multi variable problems, as soon as