Modifications of Particle Swarm Optimization Techniques and Its Application on Stock Market: A Survey

Razan A. Jamous
Pure Mathematics - Department of Mathematics, Faculty of Science, Ain Shams University
Cairo, Egypt

Assem A. Tharwat
Departments of Operations Research and Decision Support, Faculty of Computer and Information,
Cairo University

EssamEl.Seidy
Pure Mathematics - Department of Mathematics, Faculty of Science, Ain Shams University
Cairo, Egypt

Bayoumi Ibrahim Bayoum
Pure Mathematics - Department of Mathematics, Faculty of Science, Ain Shams University
Cairo, Egypt

***Abstract*—Particle Swarm Optimization (PSO) has become popular choice for solving complex and intricate problems which are otherwise difficult to solve by traditional methods. The usage of the Particle Swarm Optimization technique in coping with Portfolio Selection problems is the most important applications of PSO to predict the stocks that have maximum profit with minimum risk, using some common indicators that give advice of buy and sell. This paper gives the reader the state of the art of the various modifications of the PSO and study whether had been applied over the stock market or not.**

Keywords*—Computational intelligence; Particle Swarm Optimization; modification; Stock Market; Portfolio Selection

I. INTRODUCTION

Computational Intelligence (CI) is the study of adaptive mechanisms to enable or facilitate intelligent behavior in complex and changing environments. Studies of social animals and social insects have resulted in a number of computational models of swarm intelligence. Biological Swarm Systems that have inspired computational models include ants, bees, spiders, and bird flocks [1] The authors in]. The objective of computational swarm intelligence models is to modeling the simple behaviors of individuals, and the local interactions with the environment and neighboring individuals, in order to obtain more variant behaviors that can be used to solve complex problems, mostly optimization problems. Swarm intelligence (SI) originated from the study of colonies, or swarms of social organisms. Studies of the social behavior of organisms (individuals) in swarms prompted the design of very efficient optimization and clustering algorithms. For example, simulation studies of the graceful, but unpredictable, choreography of bird flocks led to the design of the Particle Swarm Optimization (PSO) algorithm [2]. However, in that short period, PSO has gained widespread appeal amongst researchers and has been shown to offer good performance in a variety of application domains. The usage of PSO in stock market and Portfolio Selection is very common today, whereas, the average person's interest in the stock market has grown exponentially. This demand coupled with advances in trading technology has opened up the markets, so that nowadays anybody can own stocks, and use many types of software to perform the aspired profit with minimum risk. Consequently, a lot of attention has been devoted to the analysis and prediction of future values and trends of the financial markets, and due to large applications in different business transactions, stock market prediction has become a hot topic of research. Nowadays, more software such as PSO are used to guide person to manage his portfolio and get successful investment, this motivates researchers to develop these software to give more accuracy and efficiency for successful portfolio management. In this paper we survey the state of the art of the various modifications of the PSO and study whether had been applied over the stock market or not. The rest of the paper is organized as follows. Section 2 gives historical study of particle swarm optimization. Section 3 briefly reviews the stock market. In section 4, the basic particle swarm optimization is presented. Section 5 gives the Variations of Particle Swarm Optimization and explains how it can modified PSO and the terms which we can modify it. Also, it gives the different forms for the Modifications of the original PSO. Finally, in Section 6 we conclusion this paper by the summary of main points.

II. HISTORICAL BACKGROUND OF PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart introduced particle Swarm Optimization (PSO) in 1995 as a stochastic optimization algorithm based on social simulation model [3]. Since its inception in 1995, research and application interest in PSO have increased, resulting in an exponential increase in the number of publications, Parsopoulos and Vrahatis provided statistical study about the exponential increase in number of publications about PSO during the year 2000 to 2013, in this work the statistical study was completed for the next two years, and the increase in number of PSO publications is still exponential as shown in Figure1.

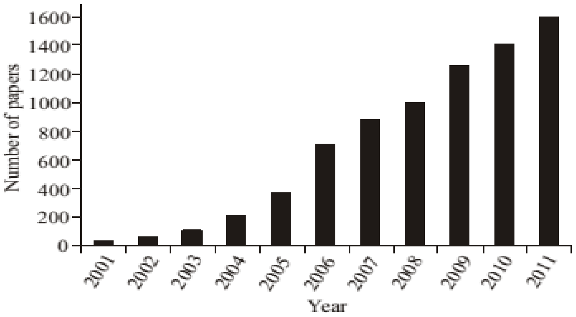

Fig. 1. Number of journal papers with the term “particle swarm” in their titles, published by three major publishers, namely Elsevier, Springer, and IEEE, during the years 2000-2011

Research in PSO has resulted in a large number of new PSO algorithms that improves the performance of the original PSO and enables application of PSO to different optimization problem types (e.g., unconstrained optimization, optimization in dynamic environments, constrained optimization, multi-objective optimization and finding multiple solutions). Elaborate theoretical studies of PSO dynamics have been done, and PSO parameter sensitivity analyses have resulted in a better understanding of the influence of PSO control parameters. PSO applications vary in complexity and cover a wide range of application areas. The PSO algorithm simulates the behaviors of bird flocking, the flight of a bird flock can be simulated with relative accuracy by simply maintaining a target distance between each bird and its immediate neighbors. This distance may depend on its size and desirable behavior. PSO learned from this and used it to solve the optimization problems. Therefore in PSO, each single solution (a bird in the search space) is called a particle, and each particle has fitness value which is evaluated by the objective function to be optimized, and has a velocity which directs the flying of the particle. All particles fly through the problem space by following the current optimum particle.

III. Stock Market Review

Stock market is, without a doubt, one of the greatest tools ever invented for building wealth. Stocks are a part, if not the cornerstone, of any investment portfolio[4]. This demand coupled with advances in trading technology has opened up the markets so that nowadays nearly anybody can own stocks, and use many types of software to perform the aspired profit with minimum risk. Consequently, a lot of attention has been devoted to the analysis and prediction of future values and trends of the financial stock markets, and due to large applications in different business transactions, stock market prediction has become a hot topic of research.

IV. Prediction of the Stock Market

A. Defining the prediction task

Before having any further discussion about the prediction of the market we define the task in a formal way. Following [15] Given a sample of N examples {(x_i, y_i), i=1,…,N} where $f(x_i)= y_i$, $\forall$i,return a function g that approximates f in the sense that the norm of the error vector E=($e_1,\dots,e_N$) is minimized. Each e_i is defined as $e_i=e(f(x_i), y_i)$ where e is an arbitrary error function”. In other words the definition above indicates that in order to predict the market you should search historic data and find relationships between these data and the value of the market. Then try to exploit these relationships you have found on future situations. This definition is based on the assumption that such relationships do exist.

B. Prediction Methods

The prediction of the market is without doubt an interesting task. In the literature there are a number of methods applied to accomplish this task. These methods use various approaches, ranging from highly informal ways (the study of a chart with the fluctuation of the market) to more formal ways (linear or non-linear regressions).We have categorized these techniques as follows:

- Technical Analysis Methods.
- Fundamental Analysis Methods.
- Traditional Time Series of Prediction Methods.
- Machine Learning Methods.

The criterion to this categorization is the type of tools and the type of data that each method is using in order to predict the market.

C. Technical Analysis

Following [5] “Technical analysis is the method of predicting the appropriate time to buy or sell a stock used by those believing in the castles-in-the-air view of stock pricing”. The idea behind technical analysis is that share prices move in trends dictated by the constantly changing attributes of investors in response to different forces. Using technical data such as price, volume, highest and lowest prices per trading period the technical analyst uses charts to predict future stock movements. Price charts are used to detect trends, and these trends are assumed to be based on supply and demand issues which often have cyclical or noticeable patterns. From the study of these charts trading rules are extracted and used in the market environment. The technical analysts are known as “chartists”. Most chartists believe that the market is only 10 percent logical and 90 percent psychological [5].

D. Fundamental Analysis

Following [5] “Fundamental analysis is the technique of applying the tenets of the firm foundation theory to the selection of individual stocks”. The analysts that use this method of prediction use fundamental data in order to have a clear picture of the firm (industry or market) they will choose to invest on. They are aiming to compute the “real” value of the asset that they will invest in and they determine this value by studying variables such as the growth, the dividend payout, the interest rates, the risk of investment, the sales level, tax rates and so on. Their objective is to calculate the intrinsic value of an asset (e.g. of a stock). Since they do so they apply a simple trading rule. “If the intrinsic value of the asset is higher than the value it holds in the market, invest in it. If not, consider it a bad investment and avoid it”. The fundamental analysts believe that the market is defined 90 percent by logical and 10 percent by physiological factors. This type of analysis is not possible to fit in the objectives of our study. The reason for this is that the data it uses in order to determine the intrinsic value of an asset does not change on daily basis. Therefore fundamental analysis is helpful for predicting the market only in a long-term basis.

E. Traditional Time Series Prediction

The Traditional Time Series Prediction analyzes historical data and attempts to approximate future values of a time series as a linear combination of these historical data .In econometrics there are two basic types of time series forecasting: univariate (simple regression) and multivariate (multivariate regression) [6].These types of regression models are the most common tools used in econometrics to predict time series. The way they are applied in practice is that firstly a set of factors that influence (or more specific is assumed that influence) the series under prediction is formed. These factors are the explanatory variables xi of the prediction model. Then a mapping between their values x_{it} and the values of the time series y_t(y is the to-be explained variable) is done, so that pairs $\{x_{it} ,y_t\}$ are formed. These pairs are used to define the importance of each explanatory variable in the formulation of the to-be explained variable. In other words the linear combination of xi that approximates in an optimum way y is defined. Univariate models are based on one explanatory variable (I=1) while multivariate models use more than one variable (I>1).

To sum up, it is possible to apply this methodology to predict the market on a daily basis. Additionally it is widely used by the economists and therefore it is a methodology that we can use for the purposes of the present study.

F. Machine Learning Methods

Several methods for inductive learning have been developed under the common label “Machine Learning”. All these methods use a set of samples to generate an approximation of the underling function that generated the data. The aim is to draw conclusions from these samples in such way that when unseen data are presented to a model it is possible to infer the to-be explained variable from these data. From these methods here are: The Nearest Neighbor and the Neural Networks Techniques. Both of these methods have been applied to market prediction; particularly for Neural Networks there is a rich literature related to the forecast of the market on daily basis [7].

V. Prediction Techniques

Many research papers have appeared in the literature using evolutionary computing tools such as genetic algorithm (GA), particle swarm optimization (PSO), bacterial foraging optimization (BFO) and genetic programming (GP) in developing forecasting models. Hassan et al. described a novel time series forecasting tool, their fusion model combines a Hidden Markov Model (HMM), Artificial Neural Networks (ANN) and Genetic Algorithms (GA) to forecast financial market behavior [12]. In another work, Aboueldahab et al. introduced a new Enhanced Particle Swarm Optimization (EPSO) to train the Sigmoid Diagonal Recurrent Neural Networks (SDRNN) weights and applied this technique in the forecasting of both NASDAQ100 and S&P500 stock market indices [9]. Majhiet al. used the standard particle swarm optimization (PSO) algorithm to develop an efficient forecasting model for prediction of S&P500 and DJIA stock indices [13]. The connecting weights of the adaptive linear combiner based model are adjusted by the PSO so that its mean square error (MSE) is minimized. In another publication [13] Majhi developed two new forecasting models based on bacterial foraging optimization (BFO) and adaptive bacterial foraging optimization (ABFO) to predict S&P500 and DJIA stock indices using technical indicators derived from the past stock indices. The structure of these models is basically an adaptive liner combiner, the weights of trained using the ABFO and BFO algorithms.

VI. Basic Particle Swarm Optimization

Individuals in a particle swarm follow a very simple behavior: to emulate the success of neighboring individuals and their own successes. The collective behavior that emerges from this simple behavior is that of discovering optimal regions of a high dimension al search space. PSO algorithm maintains a swarm of particles, where each particle represents a potential solution. In analogy with evolutionary computation paradigms, a swarm is similar to a population, while a particle is similar to an individual. In simple terms, the particles are “flown” through a multidimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors.

The following description of the PSO algorithm is adapted from [14]. Let $X_i(t)= (x_{i1}, x_{i2}, \ldots, x_{id})$ denote the position of particle i in the search space at time step t, $V_i(t) = (v_{i1},v_{i2}, \ldots, v_{id})$ denote the velocity particle i in the search space at time step t, $P_i= (p_{i1},p_{i2}, \ldots, p_{id})$ denote the best solution achieved so far by the particle itself, $P_g= (p_{g1},p_{g2}, \ldots, p_{gd})$denote the best solution achieved so far by the whole swarm. The new position of the particle is changed by adding a velocity to the current position, as follows:

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \qquad (1)$$

$$v_{id}^{(t+1)} = w.V_{id}^{(t)} + c_1 r_1 \left(P_{id} - X_{id}^{(t)}\right) + c_2 r_2 \left(P_{gd} - X_{id}^{(t)}\right) \qquad (2)$$

Where c_1 and c_2 are two positive constants, r_1 and r_2 are two random numbers in the range [0, 1]; w is the inertia weight. The velocity vector drives the optimization process, and reflects both the experiential knowledge of the particle and socially exchanged information from the particle’s neighborhood. The experiential knowledge of a particle is generally referred to as the cognitive component, which is proportional to the distance of the particle from its own best position (referred to as $pbest_i$). The socially exchanged information is referred to as the social component of the velocity equation (2), which is proportional to the distance of the particle from the best position found by the swarm (referred to as gbest).

A. Global Best PSO

For the global best PSO, or gbest PSO, the neighborhood for each particle is the entire swarm. The social component of the particle velocity update reflects information obtained from all the particles in the swarm. In this case, the social information is the best position found by the swarm.

B. Local Best PSO

For the local best PSO, or $pbest_i$ PSO, the neighborhood for each particle is small number of particles in the swarm. So that the social component reflects information exchanged within the neighborhood of the particle, reflecting local knowledge of the environment. In this case, the social information is the best position found by the experiential knowledge of the particle.

C. Velocity Components

The velocity calculation as given in equation (2) consists of three terms:

- The previous velocity, $V_i(t)$, which serves as a memory of the previous flight direction, i.e. movement in the immediate past. This memory term can be seen as a momentum, which prevents the particle from changing direction, and to bias towards the current direction. This component is also referred to as the inertia component.
- The cognitive component, c1r1(Pi–Xi), which quantifies the performance of particle irrelative to past performances. In a sense, the cognitive component resembles individual memory of the position that was best for the particle. The effect of this term
- is that particles are drawn back to their own best positions, resembling the tendency of individuals to return to situations or places that satisfied them most in the past.
- The social component, c2r2(Pg–Xi), which quantifies the performance of particle I relative to a group of particles, or the swarm. Conceptually, the social component resembles a group norm or standard that individuals seek to attain. The effect of the social component is that each particle is also drawn towards the best position found by whole the swarm.

VII. Modifications of the Original PSO

We divide different modifications on PSO into two main categories, external modifications and internal modifications.

A. External Modification Techniques

External Modification interests with all modifications which perform not on the basic components of PSO such as the method which use multi swarms or methods which split the swarm.

1) Dynamic multi-swarm particle swarm optimizer

The authors in [15] have proposed DMS-PSO based on the new neighborhood topology. In this method the whole of the population are divided into small sized swarm. Each sub-swarm uses its own members to search for better regions in the search space. In order to increase the diversity these sub-swarms are regrouped frequently to exchange the information among all particles. A local search is combined with the algorithm to improve the overall algorithm's local searching ability. The DMS-LPSO is tested on a set of benchmark functions and the results show that the proposed algorithm can find reasonable solutions for all of the problems.

2) Multi-swarm and multi-best particle swarm optimization algorithm

A new method named Multi-Best PSO (MBPSO) is proposed [16]. This method instead of using single global best position (gbest) and personal best position (Pbest), it uses the multi gbest and multi Pbest. So in the course of searching, other best values can help the best value trapped by local optimum fly out of local position. MBPSO divided the whole population into the sub-swarms and then calculates the several gbest and then combines all particles together and then calculates again taking the result as a new initial value.

3) Dynamic multi-swarm particle swarm optimizerwith sub-regional harmony search

DMS-PSO-SHS [33] is an extension of DMS-PSO that divided the whole of the population into the small sub-swarm with dynamic size to adopt each one the population of the harmony search algorithm. This method based on the DMS-PSO, generate new harmonies according to the current personal best solution and the nearer personal best solution is replaced with a new harmony with better fitness. The DMS-PSO-SHS enables the particles to have more diverse exemplars to learn from after we frequently regroup the swarms and allow the harmonies to search in a larger potential space among different sub-populations.

4) Multi-swarm Particle Swarm Optimization

In [16], the authors have proposed a Multi-swarm Particle Swarm Optimization (MPSO) to maintain the swarm diversity. This method applied a mixed local search behavior modes and information exchange among subswarms. When the premature convergence occurs in one sub-swarm then that particles should escape from the local area through the initialization their position in the search space.

5) Master-slave swarm evolutionary (MSSE-PSO)

The authors in [17] developed a shuffling master-slave swarm evolutionary algorithm based on particle swarm optimization (MSSE-PSO). The population is sampled randomly from the feasible space and partitioned into several sub-swarms (one masters warm and additional slave swarms), in which each slaves warm independently executes PSO. The master swarm is enhanced by the social knowledge of the master swarm itself and that of the slave swarms.

6) Heterogeneous partical swarm Optimization (HPSO)

The authors in [13] introduced a Heterogeneous PSO (HPSO). In the standard PSO and most of its modifications, particles follow the same behaviors. That is, particles implement the same velocity and position update rules and they exhibit the same search characteristics. In HPSO particles are allowed to follow different search behaviors in terms of the particle position and velocity updates selected from a behavior pool, thereby efficiently addressing the exploration-exploitation trade off problem. Two versions of the HPSO were proposed, the one static, where behaviors do not change, and a dynamic version where a particle may change its behavior during the search process if it cannot improve its personal best position.

7) BP algorithm

In optimizing the particle swarm optimization (PSO) that inevitable existence problem of Prematurity and the local convergence. Based on these aspects, [18] proposed a kind of modified particle swarm optimization algorithm, they take the gradient descent method (BP algorithm) as a particle swarm operator embedded in particle swarm algorithm, and at the same time they use to attenuation wall (Damping) approach to make fly off the search area of the particles of size remain unchanged and avoid the local optimal solution, with three input XOR problem to testing the improvement of the particle swarm.

B. Internal Modification Techniques

Internal Modification interests with the modifications which happened on the basic components of PSO.A number of basic modifications to the basic PSO have been developed to improve speed of convergence and the quality of solutions found by the PSO. These modifications include the introduction of an inertia weight, velocity clamping, velocity constriction, different ways of determining the global best and the local best positions, and different velocity models.

1) Velocity Clamping

One of the important aspects that determines the efficiency and accuracy of an optimization algorithm is the exploration–exploitation trade-off. Exploration is the ability of a search algorithm to explore different regions of the search space in order to locate a good optimum. Exploitation, on the other hand, is the ability to concentrate the search around a promising area in order to refine a candidate solution. A good optimization algorithm optimally balances these contradictory objectives. Within the PSO, these objectives are addressed by the velocity update equation. The velocity updates in equations (2) consist of three terms that contribute to the step size of particles. In the early applications of the basic PSO, it was found that the velocity quickly explodes to large values, especially for particles far from the local best and global best positions. Consequently, particles have large position updates, which result in particles leaving the boundaries of the search space (the particles diverge). To control the global exploration of particles, velocities are clamped to stay within boundary constraints [19]. If a particle's velocity exceeds a specified maximum velocity Vmax, the particle's velocity is set to the maximum velocity. However, the problem of finding a good value for each Vmaxin order to balance between moving too fast or too slow, and exploration/exploitation. Usually, the Vmax values are selected to be a fraction of the domain of each dimension of the search space Xmax and Xmin, and calculated as follows:

$$\mathrm{Vmax} = \delta(\mathrm{Xmax} - \mathrm{Xmin}) \tag{3}$$

Where Xmax and Xmin are respectively the maximum and minimum value of the domain, and δ ∈[0, 1]. The value of δ is problem dependent, as was found in a number of empirical studies of Shi and Eberhart [20].

a) Dissipative Particle Swarm Optimization

In order to prevent premature convergence [21] proposed Dissipative PSO (DPSO) by adding random mutation to PSO. This could be thought of as an inspiration for GA. DPSO introduces negative entropy through the addition of randomness to the particles. The results showed that DPSO performed better than standard PSO when applied to the benchmark problems.

b) Particle Swarm Optimization with passive congregation

The authors in [22] Presented a PSO with passive congregation (PSOPC) to improve the performance of Standard PSO (SPSO). Passive congregation is an important biological force preserving swarm integrity. By introducing passive congregation to PSO, information can be transferred among individuals of the swarm. This approach was tested with a benchmark test and compared with standard Gbest mode PSO, Lbest mode PSO and PSO with a constriction factor, respectively. Experimental results indicate that the PSO with passive congregation improves the search performance on the benchmark functions significantly.

c) Stochastic Particle Swarm Optimization

A new particle swarm optimizer, called Stochastic PSO (SPSO), which is guaranteed to convergence to the global optimization solution with probability one, is presented based on the analysis of the standard PSO [23]. In this approach, if the global best position is replaced by a particle's position in some interaction, this particles' position will be regenerated and if a particles' new position coincides with the global best position, its position will also be regenerated randomly. The authors have proved that this is a guaranteed global convergence optimizer and through some numerical tests this optimizer show edits good performance.

d) Cooperative Particle Swarm Optimization

A modified particle swarm optimizer named Cooperative PSO (CPSO) was proposed by Van denBergh and Engelbrecht [24]. The CPSO could significantly improve the performance of the original PSO by utilizing multiple swarms for optimizing different components of the solution vector by employ in cooperative behavior. In this method, the search space is partitioned by dividing the solution vectors into small vectors, based on the partition several swarms will be randomly generated in different parts of the search space and used to optimize different parts of the solution vector.

e) Particle Swarm Optimization with disturbance term

[25] presented PSO with disturbance term (PSO-DT) which add a disturbance term to the velocity updating equation based on the prototype of the standard PSO trying to improve (or avoid) the shortcoming of standard PSO. The addition of the disturbance term based on existing structure effectively mends the defects. The convergence of the improved algorithm was analyzed. Simulation results demonstrated that the improved algorithm have a better performance than the standard one.

f) Center Particle Swarm Optimization

It was presented by Liu and his colleagues based on introducing a center particle to the LDWPSO algorithm [25]. The center particle is proposed explicitly to visit the center of the swarm at every iteration. After N-1particles update their positions as the usual PSO algorithms at every iteration, a center particle is updated according the following formula:

$$X_{cd}^{(t+1)} = \frac{1}{N-1}\sum_{i=1}^{N-1} X_{id}^{(t+1)} \quad (3)$$

Unlike other particles, the center particle has no velocity, but it is involved in all operations the same as the ordinary particle, such as fitness evaluation, competition for the best particle, except for the velocity calculation. The center particle has opportunities to become the gbest of the swarm. Hence it can guide the whole swarm to promising region and accelerate convergence. The center particle and a randomly selected ordinary particle were recorded during the optimization process. It was clear that the center particle has higher probability to be gbest. Therefore, the center particle often guides the search process, and although it is only one particle, it imposes great effect on the swarm.

g) Mean Particle Swarm Optimization

Deep and Bansal presented MeanPSO algorithm based on a novel philosophy of modifying the formula of velocity update equation [8]. The two terms of original velocity update formula were replaced by two new terms based on the linear combination $\left(\frac{P_i+P_g}{2}\right)$ and$\left(\frac{P_i-P_g}{2}\right)$as follows:

$$V_i^{(t+1)} = V_i^{(t)} + c_1 r_1\left(\left(\frac{P_i+P_g}{2}\right) - X_i^{(t)}\right) + c_2 r_2\left(\left(\frac{P_i-P_g}{2}\right) - X_i^{(t)}\right)(4)$$

Where the first term represents the current velocity of the particle (can be thought as a momentum term). The second term is proportional to the vector $\left(\left(\frac{P_i+P_g}{2}\right) - X_i^{(t)}\right)$ and is responsible for the attraction of particle’s current position towards the mean of the positive direction of global best position (Pg) and positive direction of its own best position (Pi). The second term is proportional to the vector$\left(\left(\frac{P_i-P_g}{2}\right) - X_i^{(t)}\right)$and responsible for the attraction of particle’s current position towards the mean of the positive direction of its own best position (Pi) and the negative direction of the global best position (-Pg).

h) field-effect transistor (FET)

[26] introduces a modified particle swarm algorithm to handle multi objective optimization problems. In multi objective PSO algorithms, the determination of Pareto optimal solutions depends directly on the strategy of assigning a best local guide to each particle. In this work, the PSO algorithm is modified to assign a best local guide to each particle by using minimum angular distance information. This algorithm is implemented to determine field-effect transistor (FET) model elements subject to the Pareto domination between the scattering parameters and operation bandwidth. Furthermore, the results are compared with those obtained by the non-dominated sorting genetic algorithm-II. FET models are also built for the 3 points sampled from the different locations of the Pareto front, and a discussion is presented for the Pareto relation between the scattering parameter performances and the operation bandwidth for each model.

i) Fuzzy particle swarm optimization (FPSO)

Fuzzy particle swarm optimization (FPSO) [27] is a new variant of particle swarm optimization (PSO). Compared to PSO, each particle in FPSO is attracted by its previous best particle and other particles (not the global best particle) selected by a fuzzy mechanism. Although FPSO effectively slows down the attraction of the previous best particle and the global best particle, it shows slow convergence rate when solving complex optimization problems. To enhance the performance of FPSO, the authors propose an improved FPSO algorithm (IFPSO) which employs two strategies including generalized opposition-based learning (GOBL) and L´ evy mutation. In order to verify the performance of this approach, thirteen well-known benchmark functions and a real-world optimization problem are used in the experiments.

j) A Modified particle swarm optimization (MPSO)

The authors in [28] proposed a modified particle swarm optimization (MPSO) algorithm to solve the reliability redundancy optimization problems. The MPSO modifies the strategy of generating new position of particles. For each generation solution, the flight velocity of particles is removed. Whereas the new position of each particle is generated by using difference strategy. In addition, an adaptive parameter λ1 is used in MPSO. It can ensure diversity of feasible solutions to avoid premature convergence.

2) Inertia Weight

The inertia weight was introduced by Shi and Eberhart [11] as a mechanism to control the exploration and exploitation abilities of the swarm and as a mechanism to eliminate the need for velocity clamping. The inertia weight was successful in addressing the first objective, but could not completely eliminate the need for velocity clamping. The inertia weight, w, controls the momentum of the particle by weighing the contribution of the previous velocity, basically controlling how much memory of the previous flight direction will influence the new velocity. The value of w is extremely important to ensure convergent behavior, and to optimally tradeoff exploration and exploitation. For $w \geq 1$, velocities increase over time, accelerating towards the maximum velocity (assuming velocity clamping is used), and the swarm diverges. Particles fail to change direction in order to move back towards promising areas. For $w < 1$, particles decelerate until their velocities reach zero (depending on the values of the acceleration coefficients). Large values for w facilitate exploration, with increased diversity. A small w promotes local exploitation. However, too small values eliminate the exploration ability of the swarm. Little momentum is then preserved from the previous time step, which enables quick changes in direction. The smaller w, the more do the cognitive and social components control position updates.

a) Linear Decreasing Weight Particle Swarm Optimization

Linear Decreasing Weight particle swarm optimization (LDWPSO) algorithm was presented by Shi and Eberhart [23]. The inertia weight w is decreased linearly over the searching iterations, from an initial value to a final value as follows:

$$w = (w_{max} - w_{min}) \times \frac{(Max.Iter - Iter)}{Max.Iter} + w_{max} \quad (5)$$

Where w is the inertia weight that controls the velocity of particles, w_{max} is the initial inertia weight, w_{min} is the final inertia weight, Max. Iter is the maximum number of iterations, and Iter is the current iteration. LDWPSO algorithm uses equation (1) to update position, equation (2) to update velocity and equation (5) to update the inertia weight.

b) Particle Swarm Optimization with Dynamic Adaptation

The author in [28] proposed another dynamic inertia weight to modify the velocity update formula in a method called modified Particle Swarm Optimization with Dynamic Adaptation (DAPSO).

c) Exponential Particle Swarm Optimization

The authors in [14], Ghali and his colleagues presented Exponential particle swarm optimization (EPSO) algorithm based on simple update in the form of inertia weight formula, as follows:

$$w = (w - 0.4) \times e^{-\frac{(\text{Max.Iter} - \text{Iter})}{\text{Max.Iter}}} + 0.4 \qquad (6)$$

EPSO algorithm uses equation (1) to update position, equation (2) to update velocity and equation (6) to update the inertia weight.

d) C-Catfish PSO

Introduced chaotic maps into catfish particle swarm optimization. Swarm optimization (C-CatfishPSO) is a novel optimization algorithm proposed by [20].The [30] introduce a new parameter, called inertia weight, into the original particle swarm optimizer. Simulations have been done to illustrate the significant and effective impact of this new parameter on the particle swarm optimizer.

e) PSO with Nonlinear Decreasing inertia Weight (PSO-NDW)

Ultrasonic motor (USM) exhibits non-linearity that relates the input and output. It also causes serious characteristic changes during operation. PID controller has been widely used as the control scheme for USM. However, it is difficult for the fixed-gain type PID controller to compensate such characteristic changes and non-linearity of USM. [30] proposed a modified PSO with Nonlinear Decreasing inertia Weight (PSO-NDW) for optimal self-tuning of PID controller in positioning control of USM. A modified PSO employs the strategy that nonlinearly decreases the value of inertia weight from a large value to a small value. This strategy is to improve the performance of the standard PSO algorithm in global search and fine-tuning of the solutions. The performance of PSO-NDW based PID controller has been evaluated on the USM servo system. The results demonstrate that the proposed modified PSO can improve the accuracy of USM.

3) Acceleration Coefficients

A new approach was developed by Clerc and Kennedy, very similar to the inertia weight, to balance the exploration/exploitation trade-off, where the velocities are constricted by a constant χ, referred to as the constriction coefficient [31]. The velocity update equation changes to:

$$V_i^{(t+1)} = \chi\, [\, V_i^{(t)} + \varphi_1\big(P_i - X_i^{(t)}\big) + \varphi_2\big(P_g - X_i^{(t)}\big)\,] \qquad (7)$$

Where:

$$\chi = \frac{2k}{\left|2 - \varphi - \sqrt{\varphi(\varphi - 4)}\right|} \qquad (8)$$

With $\varphi = \varphi_1 + \varphi_2, \varphi_1 = c_1 r_1$ and $\varphi_2 = c_2 r_2$. Equation (5) is used under the constraints that $\varphi \geq 4$ and $k \in [0, 1]$. The constriction approach was developed as a natural dynamic way to ensure convergence to a stable point, without the need for velocity clamping. Under the conditions that $\varphi \geq 4$ and $k \in [0, 1]$, the swarm is guaranteed to converge. The constriction coefficient χ evaluates to a value in the range [0, 1] which implies that the velocity is reduced at each time step.

a) Constrained Particle Swarm Optimization CPSO

In the other study, a Constrained Particle Swarm Optimization (CPSO) is developed by [32]. In this method, constraint handling is based on particle ranking and uniform distribution. For equality constraints, the coefficient weights are defined and applied for initializing and updating procedure. This method applied to schedule generation and reserve dispatch in a multi-area electricity market considering system constraints to ensure the security and reliability of the power system. CPSO applied to three case studies and results showed promising performance of the algorithm for smooth and non-smooth cost functions.

Table (I), shows a summary of modifications of the PSO and whether had been applied over the stock market or not.

THE DISCUSSION OF TABLE (I)

After having a careful look at the papers we reviewed, it is concluded that there has been notably a lot of work done and remains much more scopes and areas to work on the PSO and application aspects of PSO over stock market. So the implantation these modified methods over the stock market and study the performance of these methods and how it effect to support the decision making in the stock market is very critical issue.

Certain parameters require tuning to make PSO algorithm works well. However, changing PSO parameters can have a relatively large effect. Unadjusted particles' velocity may exceed a maximum value. Particles with such speeds might explore the search space, but lose the ability to fine-tune a result. The inertia weight or the random values also control the performance of the PSO algorithm. The higher the inertia weight, the higher the particle speed. As with the maximum velocity coefficient, the setting of the inertia weight must compromise between having a good exploration of the search space and a good fine-tuning ability.

We can see that the most modifications in the papers we reviewed done on the velocity of PSO which has the most important impact on the improvement of the PSO performance. So in our work we propose a new modification on the velocity equation of PSO to improve the convergence behavior of the Particle Swarm Optimization algorithm. Then based on our new modified PSO, we develop a new effective prediction model for stock markets and use this new model for solving portfolio optimization problem to provides a better safety investment in stock market and high prediction accuracy.

As it seen from table1, the work of some researchers were interested to make internal modification. But no one interested to apply external and internal modifications on PSO at the same time.

To improve PSO performance certain parameters have to be controlled. So, the most papers we reviewed interested to modify one of the basic components of the PSO i.e. velocity clamping, inertia term, and acceleration coefficients. But no one apply multiple modifications on more than one of the basic components of the PSO.

All survived papers which covered in this survey used only pure PSO. No one merge these modified techniques with one of computational intelligence techniques to get hybrid techniques and evaluate the performance of new techniques.

VIII. Conclusion

In this paper, we explain in details the main concepts of basic particle swarm optimization algorithm and its various modifications. Also, we present a review of stock market and we surveyed the most published works since 1998 and until 2014. Then we study if these different forms of PSO are implanted over the stock market.

As we see from table1, the most modification (about 90%) has been happened as the internal modification i.e. on the basic components of PSO, whereas about 45% of modifications has been happened on the velocity clamping, while only about 20% interested to make external modification on the PSO, and other of modifications has been applied over inertia term, lastly 25% of modifications has been applied on acceleration coefficients.

As future work, we can suggest the following points:

Up to our knowledge, all the modifications on the PSO were applied as only external or internal modifications, so a lot of work can be done if we apply external and internal modifications at the same time.

- In the internal modifications, all the modifications on the PSO were applied on one of the basic components of the PSO i.e. velocity clamping, inertia term, and acceleration coefficients, so a lot of work can be done if we apply multiple modifications on more than one of the basic components of the PSO.
- All techniques which covered in this survey used PSO, so a lot of work can be done if we combine these techniques with one of computational intelligence techniques such as genetic algorithm, bacterial foraging optimization, the Nearest Neighbor and the artificial neural networks. etc., to get hybrid techniques.
- Lastly, only few techniques which covered in this survey applied over stock market, so a lot of work can be done if we study the implementation of remind techniques over the stock market.

TABLE I. A Summary of Modifications of the PSO

				External	Internal			Application
Author	**year**	**Context**	**Description**		**Velocity Clamping**	**Inertia term**	**Acceleration Coefficients**	**Stock market**
Shi et al	1998	LDWPSO	The inertia weight *w* is decreased linearly over the searching iterations, from an initial value to a final value			✓		✓
Van den Bergh and Engelbrecht	2002	GCPSO	Induce a new particle searching around the global best position found so far.		✓			
Xie et al.	2002	APSO	Improve swarm's local and global searching ability by inserting self-organization Theory.		✓			
He et al.	2004	PSOPC	Add a passive congregation part to the particle's velocity update formula.		✓			
Cui and Zeng	2004	SPSO	Particle i's position will be regenerated randomly if it is too close to the gbest.		✓			
Van den Bergh and Engelbrecht	2004	CPSO	Use multi-swarms to search different dimensions of the design space by employing cooperative behavior.		✓			
He and Han	2006	PSO-DT	Induce a disturbance term to the velocity update formula.		✓			
Yang et al.	2007	DAPSO	Use dynamic inertia weight to modify the velocity update formula.			✓		
Liu et al	2007	Center PSO	Introducing a center particle to the LDWPSO. The centre particle is proposed explicitly to visit the centre of the swarm at every iteration.		✓			✓
Zhao et al	2008	DMS-PSO	In this method the whole of the population are divided into small sized swarm.	✓				
Li and Xiao	2008	MBPSO	This method instead of using single global best position (gbest) and personal best position (Pbest), it uses the multi gbest and multi Pbest.	✓				
Ghali et al	2008	EPSO	Itupdates the form of inertia weight formula.			✓		✓

Deep et al	2009	MPSO	modifying the formula of velocity The two terms of original velocity update formula were replaced by two new terms based on the linear combination.		✓			✓
Zhao et al	2010	DMS-PSO	This method generate new harmonies according to the current personal best solution and the nearer personal best solution is replaced with a new harmony with better fitness.	✓				
Jie et al	2010	MPSO	This method applied a mixed local search behavior modes and information exchange among subswarms.	✓				
Azadani et al	2010	CPSO	Constraint handling is based on particle ranking and uniform distribution.				✓	
Jiang et al.	2010	MSSE-PSO	Population is sampled randomly from the feasible space and partitioned into several sub-swarms.	✓				
Engelbrecht	2010	HPSO	particles are allowed to follow different search behaviors selected from behavior pool	✓				
Chuang et al	2011	C-Catfish PSO	introduced chaotic maps into catfish particle swarm optimization			✓		
Ufuk et al.	2012	FET	The PSO algorithm is modified to assign a best local guide to each particle by using minimum angular distance information.		✓			
AlrijadjisDjoewahir at el.	2012	PSO-NDW	A modified PSO with Nonlinear Decreasing inertia Weight (PSO-NDW) employs the strategy that nonlinearly decreases the value of inertia weight from a large value to a small value.			✓		
Jie He et al.	2013		It take the gradient descent method (BP algorithm) as a particle swarm operator embedded in particle swarm algorithm, and at the same time they use to attenuation wall (Damping) approach to make fly off the search area of the particles of size remain unchanged and avoid the local optimal solution.	✓				
Yingsheng Su et al.	2013	FPSO	Each particle in FPSO is attracted by its previous best particle and other particles (not the global best particle) selected by a fuzzy mechanism.		✓			✓
Yubao Liu et al.	2014	MPSO	This algorithm modifies the strategy of generating new position of particles. For each generation solution, the flight velocity of particles is removed. Whereas the new position of each particle is generated by using difference strategy.		✓			

REFERENCES

[1] J. Kennedy, J. and Eberhart, C., “Particle Swarm Optimization”. Proceedings of the 1995 IEEE International Conference on Neural Networks, Australia, 1995, pp. 1942-1948.

[2] Y.Jiang, C. Liu, C. Huang and X. Wu, 2010. Improved particle swarm algorithm for hydrological parameter optimization. Appl. Math. Comput., 217: 3207-3215.

[3] J. Jie, W. Wang, C. Liu and B. Hou, 2010. Multiswarm particle swarm optimization based on mixed search behavior. Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, Jun. 15-17, IEEE Xplore Press, Taichung, pp: 605-610. DOI: 10.1109/ICIEA.2010.5517044.

[4] J. Li, and X. Xiao, 2008. Multi-swarm and multi-best particle swarm optimization algorithm. Proceeding of the 7th World Congress on Intelligent Control and Automation, June 25-27, IEEE Xplore Press, Chongqing, pp: 6281-6286. DOI: 10.1109/WCICA.2008.4593876.

[5] R. Majhi, Panda, G. Majhi, B. and Sahoo, G., “Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques”, Expert Systems with Applications. Vol. 36(6), 2009, pp. 10097-10104.

[6] J. Li, and X. Xiao, 2008. Multi-swarm and multi-best particle swarm optimization algorithm. Proceeding of the 7th World Congress on Intelligent Control and Automation, June 25-27, IEEE Xplore Press, Chongqing, pp: 6281-6286. DOI: 10.1109/WCICA.2008.4593876

[7] Y. Shi, and Eberhart, R.C. “Parameter selection in particle swarm optimization”, Proceedings of the 7th International Conference on Evolutionary Programming VII, March 25-27, 1998, p.591-600.

[8] R. Hassan, Nath, B. and Kirley, M., “A fusion model of HMM, ANN and GA for stock market forecasting”, Expert Systems with Applications, Vol. 33, 2007, pp. 171-180.

[9] T.Aboueldahab, and Fakhreldin, M, “Stock Market Indices Prediction via Hybrid Sigmoid Diagonal Recurrent Neural Networks and Enhanced Particle Swarm Optimization”, International Congress for global Science and Technology, ICGST, Vol. 10, 2010, pp. 23-30.

[10] G. Maddala, S., “Introduction to econometrics”. New York, Toronto: Macmillan Publishing Company, 1992.

[11] T. László Kóczy, Claudiu R. Pozna, "Issues and Challenges of Intelligent Systems and Computational Intelligence", New York: Springer, | ISBN-10: 3319032054,2014 .

[12] K. Deep, and Bansal, J.C. “Mean particle swarm optimization for function optimization,” Int. J. Computational Intelligence Studies, Vol. 1, No. 1, 2009, pp.72-92

[13] A. Engelbrecht, 2010. Heterogeneous particle swarm optimization. Proceeding of the 7th International Conference on Swarm Intelligence, pp: 191-202.

[14] N. I. Ghali, El-Desouki, N., Mervat, A. N. and Bakrawi, L., “Exponential Paticle Swarm Optimization Approach for Improving Data Clustering”, Proc. World Academy of Science, Engineering and Technology, Vol.32, 2008, pp. 56-60.

[15] Yubao Liu, Guihe Qin, “A Modified Particle Swarm Optimization Algorithm for Reliability Redundancy Optimization Problem ” in *Journal of Computers*, Vol 9, No 9 (2014), 2124-2131, Sep 2014, doi:10.4304/jcp.9.9.2124-2131.

[16] Y. Liu, Qin, Z., Shi, Z. and Lu, J., “Center particle swarm optimization,” Neurocomputing Vol. 70, 2007, pp. 672-679.

[17] T. Helstrom, and Holmstrom, K. “Predicting the stock market”. Published as Opuscula ISRN HEV-BIB-OP-26-SE. 1998.

[18] J. He , H. Guo, “A Modified Particle Swarm Optimization lgorithm ”, in TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 11, No. 10, October 2013, pp: 6209 - 6215 ,e-ISSN: 2087-278X

[19] R C. Eberhart, Simpson, P.K.and Dobbins, R.W., “Computational Intelligence PC Tools”. Academic Press Professional, first edition, 1996.

[20] B. Malkiel, G., “A random walk down wall street”. New York, London: W. W. Norton & Company, 1999.

[21] J. Kennedy, J. and Eberhart, C., “Particle Swarm Optimization”. Proceedings of the 1995 IEEE International Conference on Neural Networks, Australia, 1995, pp. 1942-1948.

[22] S. He, Q. Wu, J. Wen, J. Saunders and R. Paton, 2004. A particle swarm optimizer with passive congregation.Biosystems, 78: 135-147.

[23] Z. Cui, and J. Zeng, 2004. A guaranteed global convergence particle swarm optimizer. Proceedings of 4th International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden, pp: 762-767.

[24] U.¨ OZKAYA, F. G¨ UNES"A modified particle swarm optimization algorithm and its application to the multiobjective FET modeling problem", in Elec Eng & Comp Sci journal, Vol.20, No.2, 2012.

[25] Q. He, and C. Han, 2006. An improved particle swarm optimization algorithm with disturbance term. Comput. Intell. Bioinfo., 4115: 100-108.

[26] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola , “Kernel methods in machine learning”, in The Annals of Statistics Journal, Volume 36, Number 3 (2008), 1171-1220.

[27] X. Yang, J. Yuan and H. Mao, 2007. A modified particle swarm optimizer with dynamic adaptation. Appl. Math. Comput., 189: 1205-1213.

[28] S. Yingsheng and H. Fan,"AN IMPROVED FUZZY PARTICLE SWARM OPTIMIZATION FOR NUMERICAL OPTIMIZATION",in Dynamics of Continuous, Discrete and Impulsive Systems vol. 20, no. 2, pp. 173-188,2013.

[29] W.J. Xie, Zhang and Z.L. Yang, 2002a. Adaptive particle swarm optimization on individual level. 6th International Conference on Signal Processing, pp:1215-1218.

[30] L.Y. Chuang, S.W. Tsai and C.H. Yang, 2011. Chaotic catfish particle swarm optimization for solving global numerical optimization problems. Appl. Math. Comput., 217: 6900-6916.

[31] A. Djoewahir, T. Kanya, and M. Shenglin,"A Modified Particle Swarm Optimization with Nonlinear Decreasing Inertia Weight Based PID Controller for Ultrasonic Motor" International Journal of Innovation, Management and Technology, Vol. 3, No. 3, June 2012.

[32] M.Clerc, and J. Kennedy, “The particle swarm-explosion, stability and convergence in a multi dimensional complex space,” IEEE Transaction Evolutionary Computation, Vol. 6, 2002, pp.58–73.

[33] E.N., S. Hosseinian and B. Moradzadeh, 2010. Generation and reserve dispatch in a competitive market using constrained particle swarm optimization. Int. J. Electr. Power Energ. Syst. 32:79-86.