(Online) ISSN 2712-0562
*Original Research*

**Sustainable Engineering and Innovation**
Vol. 7, No. 2, 2025, pp.391-402
https://doi.org/10.37868/sei.v7i2.id610

# Using fruit fly and dragonfly optimization algorithms to estimate the Fama-MacBeth model

**Mariam Jumaah Mousa [1,2*], Munaf Yousif Hmood[2]**
[1] Department of Banking and Financial, Imam Alkadhim University College, Iraq
[2] Department of Statistics, College of Administration and Economics, University of Baghdad, Iraq

Corresponding author E-mail: maram.alamy@iku.edu.iq

Received Sep. 22, 2024
Revised Jan. 25, 2025
Accepted May 24, 2025
Online Sep. 18, 2025

**Abstract**

This research proposes the application of the dragonfly and fruit fly algorithms to enhance estimates generated by the Fama-MacBeth model and compares their performance in this context for the first time. To specifically improve the dragonfly algorithm's effectiveness, three parameter tuning approaches are investigated: manual parameter tuning (MPT), adaptive tuning by methodology (ATY), and a novel technique called adaptive tuning by performance (APT). Additionally, the study evaluates the estimation performance using kernel weighted regression (KWR) and explores how the dragonfly and fruit fly algorithms can be employed to enhance KWR. All methods are tested using data from the Iraq Stock Exchange, based on the Fama-French three-factor model. The results show that the dragonfly algorithm, particularly when using MPT and APT, demonstrates superior performance in improving the accuracy of Fama-MacBeth estimates and enhancing the effectiveness of the KWR approach.



*Keywords*: Fama-MacBeth model, Dragonfly algorithm, Fruit fly algorithm, Parameter optimization, Kernel weighted regression, Iraq Stock Exchange

## 1. Introduction

The Fama-MacBeth model is a stalwart of econometrics with a strong framework for cross-sectional regression analysis, primarily when used to apply to independent variables as surrogates [1]. It finds usage in identifying the most significant factors that affect the studied variable [2, 3], measurement of model fitness [4], and estimation of time-varying model parameters [5]. The model approach goes in two broad steps: initially, beta coefficients are estimated using time-series regressions, followed by gamma estimation using cross-sectional regressions after introducing the beta of the first step as independent variables. Though they are undoubtedly crucial, sophisticated model estimates are not flawless, and this is something that encourages us to seek improved estimation approaches. In this study, we suggested applying two artificial intelligence algorithms to achieve the optimal solution by approximating the parameter of the Fama-MacBeth regression model by applying the fruit fly algorithm and the dragonfly algorithm, and comparing the outcome with the kernel weights regression obtained in the study of [6].

The Fruit Fly Optimization Algorithm (FOA) is a highly efficient optimization technique inspired by the foraging behavior of fruit flies, which rely on their senses of smell and sight to locate optimal food sources. FOA has demonstrated its effectiveness in solving a wide range of optimization problems across various fields.

As a result, it has gained significant attention and has been successfully applied to numerous real-world problems, including financial modeling [7], operational optimization in chemical engineering [8], mathematical analysis in cloud computing [9], power load forecasting [10], web auction logistics services [11], PID controller tuning [12], the multidimensional knapsack problem [3], and many other scientific applications [13].

One prominent application of FOA is in radar target recognition, specifically in analyzing high-resolution range profiles (HRRP). To enhance HRRP performance, FOA is employed to optimize the model parameters of the Generalized Regression Neural Network (GRNN), a model widely used in science, engineering, and finance.

The dragonfly optimization algorithm (DOA), on the other hand, is inspired by the static and dynamic flight behaviors of dragonflies during hunting and migration. It mimics the swarm intelligence of dragonflies to explore and exploit the search space effectively. By leveraging these behaviors, the DOA is capable of finding optimal or near-optimal solutions to complex optimization problems. Its versatility continues to drive the expansion of its applications across various domains.

The dragonfly optimization (DO) approach has drawn a great deal of attention from several domains of application owing to its proven capability in addressing challenging optimization problems. Previous work has investigated a broad variety of applications, demonstrating the flexibility and universality of this algorithm. DA has also been employed as an example to identify pixels containing salient object information in images [14], leading to fast and efficient object segmentation. During parameter tuning, the authors in [15] utilized DA to select the optimal feature set among candidates and the optimal parameter values (penalty factor and kernel parameter) for KELM. In this case, DA showed its strength as a search strategy. Iteratively tuning DA parameters is feasible; DA has also been applied as an optimizer of support vector machine (SVM) parameters [16], with higher performance than PSOSVM and GASVM. The experiment also demonstrated that although the classification error rate is reduced by adding more solutions or generations, the computation time is longer. In several image processing applications, DA has demonstrated superior capabilities. It produced better results than previous methods when used to segment color fundus images [17]. According to [18], it has also been used in medical image watermarking approaches. In these cases, it made it easier to choose efficient pixels, resulting in greater correlation coefficient values compared to algorithms like PSO, GA, and random selection. In the field of forecasting, a DA-based artificial neural network was used to predict initial fuel demand in India [19].

Although earlier studies have demonstrated the capabilities of fruit fly and dragonfly optimization algorithms in addressing optimization problems, it is worth noting that most studies focus on applications such as image processing or the optimization of neural network performance. This highlights the urgent need to extend the application of these two algorithms to sophisticated financial models like the Fama-MacBeth model so that their parameter estimates are derived with higher accuracy and model likelihood estimation errors are reduced, thereby increasing the accuracy of assessing asset pricing models. The Fama-MacBeth model presents a special estimation problem because it becomes more complicated when employing time-varying beta coefficients obtained in the first step as explanatory variables in the second step, described in terms of cross-sectional regression.

This research aims to address this issue by developing improved estimators for this significant financial model. Beyond that, this paper is the first known attempt to make a direct comparison of the dragonfly and fruit fly algorithms' performance in estimating the Fama-MacBeth model. Moreover, the study seeks to enhance the efficiency of the KWR algorithm, one of the most widespread algorithms used in this area. To these ends, this study proposes the employment of the dragonfly and drosophila algorithms. For enhanced performance of the dragonfly algorithm itself, the study examined three approaches to parameter tuning: manual parameter tuning (MPT), adaptive tuning on methodology (ATY) [20], and an innovative new proposed adaptive parameter tuning approach based on performance. The performance of the aforementioned estimation methods will subsequently be compared with that of the kernel weighted regression (KWR). The research will be further extended to suggest the application of the dragonfly and fruit fly algorithms for optimizing the parameters of the KWR method itself. All these methods will be experimented upon on the Iraq Stock Exchange dataset using the Fama-French three-factor model.

## 2. Fama-Macbeth models (FM)

The Fama-Macbeth regression model will be in the following form:

$$Y_{i,t} = \bar{\gamma}_{0,t} + \bar{\gamma}_{1,t}B_{1,i,t} + \bar{\gamma}_{2,t}B_{2,i,t} + \cdots + \bar{\gamma}_{K,t}B_{K,i,t} + u_{i,t} \quad \ldots(1)$$

Where:

$Y_{i,t}$ shows the dependent variable with i =1, 2, 3… N variable

t = 1, 2, 3,.,T Time

$\beta_{1,i,t}$, $\beta_{2,i,t}$, $\beta_{k,i,t}$ are independent variables taken from step 1 of FM represented by a time series regression.

$\bar{\gamma}_{0,t}, \bar{\gamma}_{1,t}, \bar{\gamma}_{2,t}, \ldots, \bar{\gamma}_{K,t}$ denote the mean of the intercept $\gamma_{0,t}$ and $\gamma_{1,t}$, $\gamma_{2,t}$, … , $\gamma_{K,t}$ is the estimate of the effect of independent on dependent variables (in period t) for the K factors, and number observation T. $u_{i,}$ is the error term independent identical distribution (iid) with (mean 0, variance $\sigma^2$).

## 3. Methodology of estimation FM

In this section, we suggest introducing the fruit fly optimization algorithm and dragonfly optimization Algorithms to a Fama-Macbeth model (FMFOA), (FMDA), respectively, of time series regression to estimate $\beta_{k,i,t}$ (proxy variables ) and cross-sectional regression to estimate $\gamma_{K,t}$

The important steps of the fruit fly optimization algorithm are parameter and population location initialization

The main parameters of the FOA are the maximum iteration number (iter). This determines how many times the algorithm will iterate before stopping, the population size (flies) represents the number of fruit flies in the swarm, and the random flight distance range randValue. This defines the range within which fruit flies explore the search space randomly; each fly contains the number of parameters to be estimated.

The fruit fly swarm location ($\theta \quad intial$) is randomly initialized in the search space as follows.

$$\theta \quad intial = rand * (UB - LB) + LB \qquad i=1,2,\ldots N \qquad \ldots(2)$$

Where $\theta \quad intial$ in the first step of FM is $B_{k,i,t}intial$ for every time point t with k independent variables, and every dependent variable's i, rand is a matrix of order (population size P (number of petameter)) random function which returns a value from the uniform distribution on the interval [0, 1].

In the second step of FM, the fruit fly swarm location ($\theta \quad intial$ is $\gamma_{S,p}intial$ for every cross-sectional s. rand is a random matrix of order (NP (number of petameter)). the UB and LB are the upper and lower bounds of the fruit fly swarm location.

A crucial step in the fruit fly optimization process is osphresis search. All flies have keen senses of smell. As if utilizing its sense of smell to identify possible food paths, each fly in this stage creates new locations surrounding its current location at random. This stage gives the flock the opportunity to search a larger area, which raises the likelihood of discovering new solutions at random and, consequently, by preventing an early convergence of the flock's unpredictable flight patterns to less-than-ideal sites, it also maintains diversity. Furthermore, the likelihood of discovering the best one. Then the swarm location is determined by searching based on the osphresis search, as the equation below illustrates.

$$\text{theta\_swarm} = \theta intial + \text{randValue} \quad \ldots.(3)$$

Where theta_swarm to estimate beta is beta_swarm(i, :), and to estimate Gamm, the theta_swarm is Gamma swarm (s,: ).

The value of randValue falls between -1 and 1. By guaranteeing small, localized motions within the range [-1, 1], it preserves equilibrium between exploration (global search) and exploitation (local refining).

It is a quantitative measure of the quality of the solution represented by the fruit fly site. It is calculated using a fitness function, which depends on the nature of the problem, as the fitness function varies depending on the problem being solved. The algorithm reflects the quality of this site in the context of the specific problem.

In the context of the FM step, the smell concentration is calculated using the objective (fitness) function, which measures the quality of the proposed solution. This function is the root mean square error (RMSE). However, due to the odor concentration principle, the reciprocal of the RMSE is used instead.

$$Smell_i = 1/RMSE \qquad \ldots(4)$$

In the first step of FM, $Smell_i$ calculates for each fruit fly $B_{k,i,t} intial$, beta_swarm(i, :) (each fly represents the beta values of all independent variables), of the time series regression model with rolling regression.

In the second step of FM for $Smell_i$ calculate to $\gamma_{s,} intial$, Gamma swarm (s,: ), were used to conduct the cross-sectional analysis.

D) Determine the best location for vision search

This is the location that achieves the highest smell concentration and the highest value of the ($Smell_i$ ) to estimate beta.

[smellBest,best index]=max(smelli);with $B_{k,i,t}$

[best smell,best_index]=max(smelli);with beta_swarm(i, :)

Then, the current maximum smell concentration value (bestSmell) is compared with the value (smellBest). If bestSmell > smellBest, smellBest is updated with bestSmell, and the fruit fly swarm flies towards that location with the maximum smell concentration value by using vision.

```
if current_bestsmell > smellbest
smellBest = bestSmell
best_beta = beta_swarm(best index, :);
   end
```

And to estimate Gamma after introducing time-varying betas as independent variables in cross-sectional regression, as follows:

[SmellGAMMA Best,bestindex]=max(SmellGAMMA i);with $\gamma_{s,p} intial$ …(8)

[best SmellGAMMA,best_index]=max(SmellGAMMA i);with Gamma swarm (s,: ) …(9)

Then, the current maximum smell concentration value (best SmellGAMMA is compared with the value (SmellGAMMA Best. If best SmellGAMMA > SmellGAMMA Best, SmellGAMMA Best is updated with best SmellGAMMA, and the fruit fly swarm flies towards that location with the maximum smell concentration value by using vision.

```
if current best SmellGAMMA > SmellGAMMA Best
SmellGAMMA Best = best SmellGAMMA
     best_beta = Gamma swarm (best index, :);
   end
```

The osphresis searching phase and vision searching phase are repeated until the smell concentration is not superior to the previous iterative smell concentration anymore, or the iterative number reaches the maximum iterative number.

For the dragonfly optimization algorithms to a Fama-MacBeth regression model, the algorithm below illustrates the steps used to estimate a Fama-MacBeth model using the dragonfly optimization algorithm**.**

We start by initializing a swarm of dragonflies to estimate the beta coefficients of each of the available assets by specifying the number of dragonflies that will participate in the search and the beta values, then randomly initializing the starting positions of these dragonflies within the given solution space. Finally, to guide the movement of each dragonfly in the solution space, we assign it a random initial velocity or step vector where:

A – Number of dragonflies (n): Determine the number of possible solutions to explore to estimate the beta coefficients for each dependent variable in the first step (time series regression) and in the second step (cross-sectional regression) to estimate gamma coefficients.

B – Solution space is determined based on theoretical predictions or previous studies in the first step. Bounds of the beta coefficients [$\beta_{i,k,t}$]. For each dependent variable i from 1 to N, k variables, for each point time t=1,2…,T., and in the second step [$\gamma_{k,s}$] s=1…T number of cross-sectional.

C – Initial dragonfly locations ($P_j(0)$), j=1, …n

$$P_j(0)=[\theta_{0j}(0), \theta_{1j}(0),..., \theta_{kj}(0)]$$

Where ($P_j(0)$) in the first step to estimate beta is $P_{i,j\,t}(0)$, and $\theta_{0j}(0)$ is $\beta_{kijt}(0)$

In the second step to estimate Gamma, the initial dragonfly locations are $P_{j,s}(0)$, and $\theta_{kj}(0)$ is $\gamma_{k.j,s}(0)$.

D – Initial step vectors $V_{,j}(0)$

The step vector represents the magnitude and direction of the movement the dragonfly will make in the solution space in subsequent iterations. $\Delta\theta_{kj}(0)$ represents the step vector of the j-th dragonfly. For each j-th dragonfly in the swarm, a vector $\Delta\theta_{kj}(0)$ is created with the same dimensions as its position vector Xi. Its elements have relatively small random values around zero. These values determine the magnitude of the change in each element of the dragonfly's position $P_j(0)$. They are concerned with the direction of the movement, that is, the direction in which the dragonfly will start moving in the solution space, the $V_{,j}(0)$ is:

$$V_j(0)=[\Delta\theta_{0j}(0), \Delta\theta_{1j}(0),..., \Delta\theta_{kj}(0)]$$

Where ($V_j(0)$) in the first step to estimate beta is $V_{i,j\,t}(0)$, and $\Delta\theta_{0j}(0)$ is $\Delta\beta_{kijt}(0)$

And in the second step to estimate Gamma, the initial step vectors are $V_{j,s}(0)$ and $\Delta\theta_{kj}(0)$ is $\Delta\gamma_{k.j,s}(0)$.

To evaluate the quality of the solution represented by each dragonfly, the goal of the optimization process is to find the solution that gives the best value for the objective function (either a maximum or a minimum value, depending on the nature of the problem).

In the context of estimating the beta coefficients for a time series regression in the first step of the Fama-MacBeth model, the objective function measures how well the set of beta coefficients represented by each dragonfly explains the relationship between the dependent variable and the independent variables for each asset and each point in time. The objective function is represented by the root sum of square error (RMSE), measuring how well the estimated time series regression differs (based on current beta coefficients) from the true dependent variable, with the goal of minimizing RMSE.

In the context of estimating the gamma coefficients for a cross-section regression in the second step of the Fama-MacBeth model, the objective function measures how well the set of gamma coefficients represented by each dragonfly explains the relationship between the dependent variable and the independent variables, represented by the time-varying betas for each cross-section. The objective function is represented by the root sum of squared errors, measuring how well the estimated cross-section regression differs (based on the best parameters of the beta and current gamma coefficients) from the true dependent variable for each cross-section, to minimize RMSE.

The objective function value is calculated for each dragonfly to evaluate solutions and guide the search process. The objective function values are used to identify the best solutions that have been identified. Finding it so far (Food Pi), the food source in the algorithm, and also identifying the worst solution (the enemy in the algorithm (Enemy Pi)). This information guides the dragonflies' movement in subsequent iterations toward promising regions in the solution space and avoiding unfavorable regions. Updating the best solutions: In each iteration, the objective function values of the current solutions are compared with the best solutions found previously. If the dragonfly finds a better solution, the best solution is updated.

For E – Update, the food source (Food Pi) and the enemy (Enemy Pi) are updated at each iteration to ensure that the best and worst outcomes change as the dragonflies move through the solution space. A new dragonfly might find a solution that is better than the current "food source" or worse than the current "enemy." Therefore, these locations must be updated at each iteration to ensure that the "food source" always represents the best solution found up to that point, and that the "enemy" always represents the worst solution. To guide movement, the location of the "food source" is used to attract other dragonflies toward it (foraging behavior), and the location of the "enemy" is used to repel other dragonflies (enemy avoidance behavior). Updating these locations ensures that movement is based on the latest information about promising and unpromising regions in the solution space.

For F – Update the behavior parameters, w: inertia weight, s: separation weight, a: orientation weight, c: cohesion weight, f: food attraction weight, e: enemy repulsion weight. Determine how dragonflies move and interact. The influence of these parameters' changes during the search between the exploration and exploitation phases. In the FM, the behavior parameters may be different.

For G – Update neighboring radius, to adjust the range of influence dragonflies have on each other, i.e., a dynamic mechanism in the dragonfly algorithm during the search process, aiming to achieve an effective balance between exploring the solution space and exploiting promising regions.

For H – Update velocity vector, to update the speed: If the dragonfly has neighbors, we use the following equation:

$$V_{,j}(t+1)=w\, V_{,j}(t)+s\, S+a\, A+c\, C+f\, F+e\, E \qquad \ldots(17)$$

If the dragonfly has no neighbors:

$$P_j(t+1)=P_{j,}(t)+rand\cdot L1 \qquad \ldots(19)$$

The dragonfly moves randomly to explore new regions in the solution space. Rand is a random number between 0 and 1 (or a vector of random numbers in each dimension), and L1 is the length of the random step.

Where:

S: The separation to static collision avoidance of the individuals from other individuals in the neighborhood, and is calculated as:

$$S=\sum_{l=1}^{n}\left(P_j-P_l\right)*e^{\|P_j-P_l\|} \qquad \ldots(12)$$

Where $P_j$ is the position of the current individual, $P_l$ shows the position l-th neighboring individual.

A (Alignment): The velocity matching of individuals to other individuals in the neighborhood is calculated as:

$$A=\frac{\sum_{l=1}^{n}V_l}{n} \qquad \ldots(13)$$

Where $V_l$ shows the velocity of the l-th neighbouring individual.

$C(P_j)$ (Cohesion): the tendency of individuals towards the center of the mass of the neighborhood

$$C=\frac{\sum_{l=1}^{n}P_l}{n}-P_j \qquad \ldots(14)$$

$F(P_j)$: Attraction to food calculated as:

$$F = Food\ P_t - P_j \qquad \ldots(15)$$

Here $P_j$ is the position of the current individual, and $Food\ P_i$ shows the position of the food source.

$EP_j$ Distraction outwards an enemy is calculated as follows:

$$E = \text{Enemy}P_t + P_j \qquad \ldots(16)$$

where $P_j$ is the position of the current individual, and Enemy$P_t$ shows the position of the enemy.

I – Updating the position vector.

$$P_{,j}(t+1)=P_j(t)+V_{,j}(t+1) \qquad \ldots(18)$$

J – Check and correct the bounds are employed to obtain beta time varying estimates: $\hat{\beta}_{i,t} = Food\ P_{i,t}$ , , $\hat{\gamma}_S = Food\ P_S$ based on estimated beta as independent variable.

To improve the performance of the dragonfly algorithm (DA) in estimating the parameters of the Fama-MacBeth model, this research explores and compares three different strategies for tuning the algorithm's basic behavioral parameters: manual tuning, adaptive performance-based tuning, and adaptive tuning according to the methodology of [20].

1. Manual parameter tuning (MPT)

In this approach, there were fixed values assigned for the parameters s, a, c, f, e, and w. These were based on the initial experiments and experience about how the algorithm was acting in relation to the problem to be solved. Then, the best set that provides the most optimal performance is chosen based on some parameters, e.g., solution precision and convergence rate.

2. Adaptive tuning

An adaptive parameter tuning mechanism was introduced based on [20] (YTA). In this introduction, the value of the enemy repulsion coefficient (e) was computed dynamically according to the following equation

$$e = 0.1 - i\left(\frac{0.1}{\left(\frac{I}{2}\right)}\right) \qquad \ldots(19)$$

$$w = 0.9 - i\left(\frac{0.9 - 0.4}{I}\right) \qquad \ldots(20)$$

Where:

i is the current iteration and

I is the number of iterations, where s, a, and c are three different random numbers between 0 and 2e, f is a random number between 0 and 2, and w is calculated in Equation 20.

This paper proposes an adaptive tuning (APT) approach for the dragonfly algorithm's parameters. This method is based on evaluating the algorithm's efficiency during the search process. This modification seeks to achieve three main goals: improving the algorithm's flexibility to adapt to the nature of the problem, accelerating the convergence process toward the optimal solution, and improving the quality of the final solutions. The parameter values (w, s, a, c, f, e) are continuously updated as the number of iterations progresses. Instead of relying on fixed magnitudes, these magnitudes are dynamically adjusted at each iteration (iter) using the swarm's position or its search progress, using the following formulas:

$$P = (\max iteriter) * (\min p - \max\ p) + \max\ p \qquad \ldots(21)$$

Where:

p stands for a value for a parameter to be updated.

iter is the current iteration number in the optimization process.

max iter is the maximum number of iterations specified to terminate the optimization process.

min p is the minimum value that parameter p can reach at the end of the optimization process.

max p is the maximum value that parameter p starts with at the beginning of the optimization process.

## 4. Results and discussion

This research evaluates asset returns on the Iraq Stock Exchange using the Fama-MacBeth and Fama-French three-factor models. The model utilizes the market return index, book-to-market value, and firm size to assess asset risk. The model was applied to 22 listed companies using quarterly data from the first quarter of 2010 to the first quarter of 2024. Missing values were estimated using the KNN method, depending on the number of neighbors.

To estimate the parameters of the Fama-MacBeth model, the fruit fly (FOA) and dragonfly (DA) algorithms were applied. The maximum number of iterations was set at 1,000, and the swarm size for both algorithms was set to 50 individuals. Given the nature and volatility of financial data, the initial positions of individuals in both algorithms were randomly initialized within the range [-10, 10].

For the dragonfly algorithm, the optimization process required fine-tuning the basic behavioural parameters following the methodology described in [20], which is (ATY). Additionally, two tuning strategies were explored: adaptive tuning (APT) and manual tuning (MPT). Experiments showed that manual parameter tuning provided the best results in the context of this study. The optimal values achieved for manual tuning were: s = 0.01, a = 0.1, c = 0.1, f = 0.5, and e = 0.1. The inertia weight (w) was linearly decreasing from 0.9 to 0.2 over the iterations to achieve a balance between exploration and exploitation in the search space. The proposed methods, DA with three methods to select parameters of DA and FOA (swarm size =50, iter=1000, bound=rand value).

In the first step of the Fama-MacBeth approach, we used time-series regression with a rolling window regression approach to estimate time-varying beta for each dependent variable. And in the Second step, we represented cross-sectional regression to estimate gamma based on time-varying beta as independent variables from the first step of FM We find the RMSE for each cross-section as shown in Table 1.

Table 1. RMSE of the Fama-Macbeth model

| Date | FMDA(MPT) | FMDA (ATY) | FMDA(APT) | FM-FOA | KWR | FMDA(APT) | DAKWR(MPT) | FOAKWR |
|---|---|---|---|---|---|---|---|---|
| 12/01/2014 | 0.18500 | 0.37928 | 0.18586 | 0.19653 | 0.20141 | 0.18655 | 0.18554 | 0.19347 |
| 03/01/2015 | 0.07391 | 0.39187 | 0.07817 | 0.10450 | 0.12819 | 0.07488 | 0.07335 | 0.09055 |
| 06/01/2015 | 0.21450 | 0.28158 | 0.21635 | 0.22215 | 0.25252 | 0.21518 | 0.21392 | 0.23427 |
| 09/01/2015 | 0.12012 | 0.26568 | 0.12332 | 0.14649 | 0.12285 | 0.12147 | 0.12017 | 0.12180 |
| 12/01/2015 | 0.16004 | 0.24009 | 0.16003 | 0.16651 | 0.17713 | 0.16100 | 0.16039 | 0.15802 |
| 03/01/2016 | 0.12119 | 0.32506 | 0.12172 | 0.13841 | 0.16687 | 0.12016 | 0.11989 | 0.13969 |
| 06/01/2016 | 0.11791 | 0.43175 | 0.11649 | 0.13328 | 0.18788 | 0.12118 | 0.11775 | 0.12343 |
| 09/01/2016 | 0.13468 | 0.26774 | 0.13680 | 0.15501 | 0.14509 | 0.13561 | 0.13414 | 0.13784 |
| 12/01/2016 | 0.12560 | 0.30263 | 0.12482 | 0.14176 | 0.18530 | 0.13086 | 0.12271 | 0.12699 |
| 03/01/2017 | 0.13175 | 0.38456 | 0.13451 | 0.16160 | 0.15056 | 0.13338 | 0.13205 | 0.14089 |
| 06/01/2017 | 0.10348 | 0.37520 | 0.10674 | 0.11324 | 0.15369 | 0.10957 | 0.10421 | 0.11429 |
| 09/01/2017 | 0.22685 | 0.34338 | 0.22691 | 0.23179 | 0.24716 | 0.22716 | 0.22693 | 0.22988 |

| Date | FMDA(MPT) | FMDA (ATY) | FMDA(APT) | FM-FOA | KWR | FMDA(APT) | DAKWR(MPT) | FOAKWR |
|---|---|---|---|---|---|---|---|---|
| 12/01/2017 | 0.11783 | 0.27169 | 0.11757 | 0.14790 | 0.11945 | 0.11821 | 0.11702 | 0.11822 |
| 03/01/2018 | 0.16296 | 0.38514 | 0.16272 | 0.16956 | 0.26504 | 0.16494 | 0.16161 | 0.16216 |
| 06/01/2018 | 0.15882 | 0.28196 | 0.16102 | 0.17094 | 0.17014 | 0.16077 | 0.15910 | 0.16203 |
| 09/01/2018 | 0.13762 | 0.17841 | 0.13729 | 0.14761 | 0.14880 | 0.13911 | 0.13749 | 0.13960 |
| 12/01/2018 | 0.13614 | 0.39372 | 0.13618 | 0.14478 | 0.13946 | 0.13762 | 0.13619 | 0.13799 |
| 03/01/2019 | 0.19803 | 0.32817 | 0.19892 | 0.20699 | 0.22658 | 0.19992 | 0.19815 | 0.20224 |
| 06/01/2019 | 0.14497 | 0.40304 | 0.14587 | 0.15539 | 0.15457 | 0.14602 | 0.14522 | 0.14685 |
| 09/01/2019 | 0.18386 | 0.17915 | 0.18447 | 0.20275 | 0.18368 | 0.18392 | 0.18337 | 0.18783 |
| 12/01/2019 | 0.09313 | 0.20769 | 0.09572 | 0.09655 | 0.14102 | 0.09843 | 0.09321 | 0.12007 |
| 03/01/2020 | 0.11423 | 0.22612 | 0.11550 | 0.12167 | 0.15547 | 0.11596 | 0.11431 | 0.12351 |
| 06/01/2020 | 0.09181 | 0.31148 | 0.09220 | 0.09729 | 0.09506 | 0.09267 | 0.09191 | 0.09268 |
| 09/01/2020 | 0.15072 | 0.25987 | 0.15019 | 0.17113 | 0.23428 | 0.15194 | 0.14972 | 0.15860 |
| 12/01/2020 | 0.12118 | 0.25708 | 0.12241 | 0.13966 | 0.13043 | 0.12343 | 0.12141 | 0.12395 |
| 03/01/2021 | 0.10565 | 0.34082 | 0.11012 | 0.10988 | 0.48627 | 0.10923 | 0.10720 | 0.11525 |
| 06/01/2021 | 0.23294 | 0.33334 | 0.23359 | 0.23338 | 0.23921 | 0.23278 | 0.23293 | 0.23279 |
| 09/01/2021 | 0.11249 | 0.33390 | 0.11403 | 0.13387 | 0.11303 | 0.11430 | 0.11267 | 0.11407 |
| 12/01/2021 | 0.23686 | 0.48998 | 0.24561 | 0.24948 | 0.48571 | 0.24363 | 0.23650 | 0.26034 |
| 03/01/2022 | 0.19596 | 0.43952 | 0.19930 | 0.19819 | 0.23872 | 0.19442 | 0.19732 | 0.21134 |
| 06/01/2022 | 0.20894 | 0.32270 | 0.21179 | 0.21596 | 0.22135 | 0.21019 | 0.20892 | 0.20877 |
| 09/01/2022 | 0.28284 | 0.38978 | 0.28078 | 0.28633 | 0.33219 | 0.28476 | 0.28269 | 0.29212 |
| 12/01/2022 | 0.14009 | 0.41177 | 0.14333 | 0.15241 | 0.23017 | 0.14245 | 0.13999 | 0.16705 |
| 03/01/2023 | 0.24120 | 0.33741 | 0.24394 | 0.25074 | 0.27573 | 0.24043 | 0.24005 | 0.24559 |
| 06/01/2023 | 0.15324 | 0.54264 | 0.15940 | 0.17737 | 2.51627 | 0.15236 | 0.15418 | 0.17191 |
| 09/01/2023 | 0.17886 | 0.38206 | 0.17918 | 0.19904 | 0.27565 | 0.18448 | 0.18181 | 0.20502 |
| 12/01/2023 | 0.15612 | 0.41927 | 0.15775 | 0.17410 | 0.15655 | 0.15592 | 0.15583 | 0.15565 |
| 03/01/2024 | 0.17195 | 0.43517 | 0.17654 | 0.18821 | 0.19365 | 0.17507 | 0.17188 | 0.18242 |
| Average RMSE | 0.155323 | 0.33818 | 0.15808 | 0.16980 | 0.26440 | 0.15816 | 0.15636 | 0.16445 |

Table 1 compares the performance of eight different data interpretation methods, based on the root mean square error (RMSE) measure. The results clearly show that the DA (MPT) method performed best across all studied assets and for all cross-sections, recording the lowest RMSE values compared to the other methods. This indicates that DA (MPT) was the most capable of providing accurate and reliable interpretations of the data. However, DA (ATY) was weak and possessed the highest RMSE values. That is to say that this algorithm was the least capable of reading the data, and the prediction made by it was far from the actual values. DA (APT) performed immensely well, and its performance was nearest to that of DA (MPT). That means that DA (APT) is a viable substitute that can come up with good interpretations of the data. The fruit fly algorithm (FOA) outperformed the KWR and DA (ATY) algorithms in data estimation with low values of RMSE. Moreover, Tables 1 and 2 indicate how the results of the KWR algorithm were enhanced upon being optimized through the dragonfly and fruit fly algorithms. This enhanced the KWR estimate by yielding lower RMSE values compared to using the KWR values in isolation, thus affirming the optimization's ability to increase the accuracy of the estimates.

Figure 1 shows the development of the proposed DA (MPT)'s objective function over iterations when it is used for estimating the coefficients of Fama-MacBeth regression models. An effort is made to comprehend the behavior of the algorithm and the convergence towards the optimal solution.

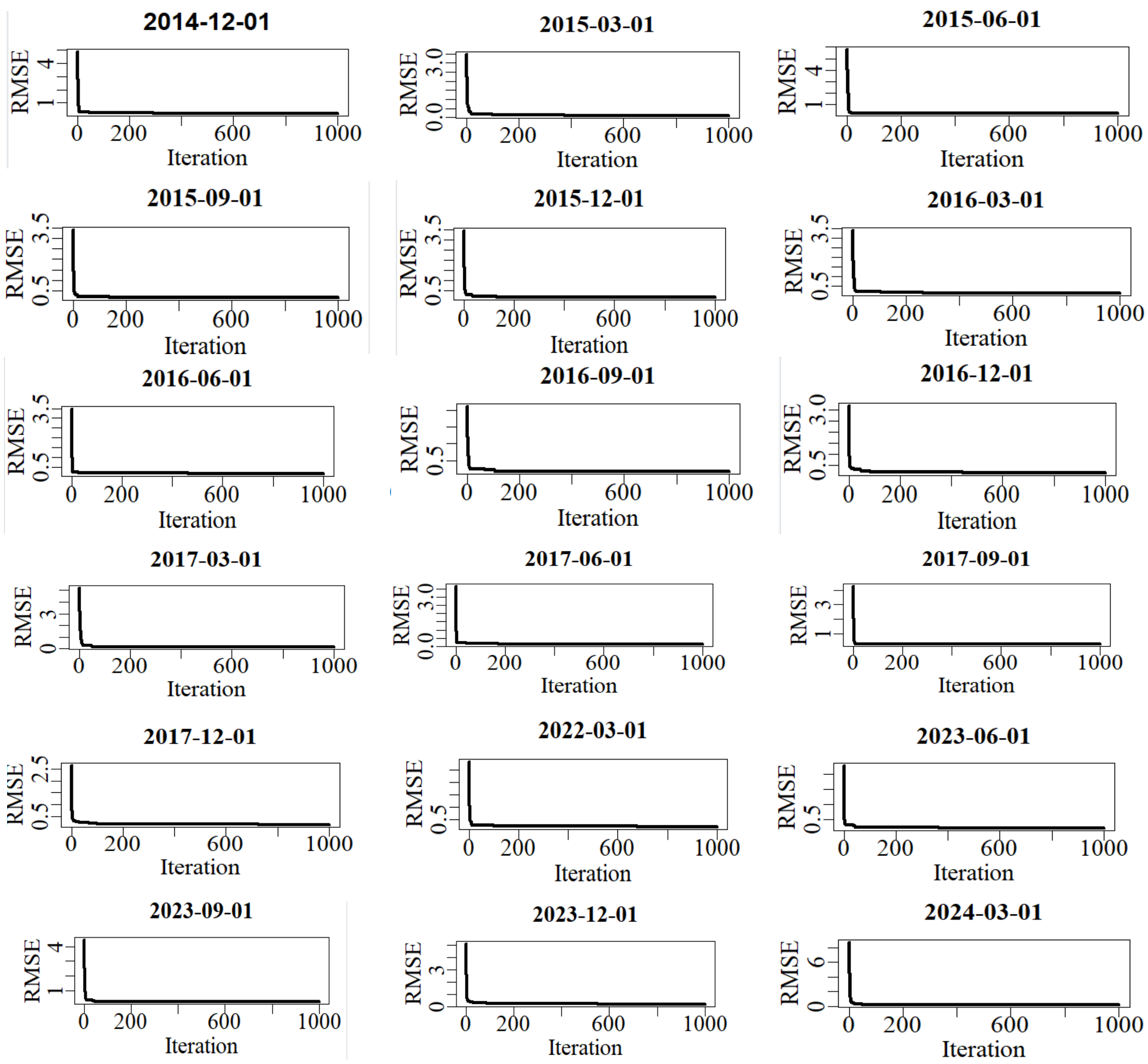


Figure 1. Evolution of the objective function of the DA (MPT) for the Fama-MacBeth model across iterations

Figure 1 illustrates the convergence path of the dragonfly algorithm toward the optimal solution for parameter estimation in the Fama-MacBeth regression model across all given cross-sections. In each subplot, it is evident that the value of the objective function starts at a high level during the initial iterations and progressively decreases as the algorithm explores the solution space. This behavior clearly demonstrates the dragonfly algorithm's capability to efficiently navigate the search space and converge toward regions that minimize the model's error.

The steady decline in the objective function value across iterations highlights the effectiveness of the algorithm's optimization process in identifying parameter sets that best fit the data for each cross-section. Furthermore, this approach can be extended and enhanced using linear and non-linear techniques [21], as well as through integration with fractal-based methods [22, 23].

## 5. Conclusion

The findings of this research, whose purpose was to enhance the Fama-MacBeth model estimates by the dragonfly and fruit fly algorithms, showed a distinct superiority of the dragonfly algorithm compared to the fruit fly algorithm in approximation estimation. Apart from this, the research also proved that parameter tuning of the dragonfly algorithm via adaptive tuning, especially the suggested performance-based tuning (APT) mechanism, was equal in efficiency with manual tuning (MPT) in generating results that were very close to one another. Finally, the outcome confirmed the positive impact of both dragonfly and fruit fly algorithms on enhancing kernel weighted regression (KWR) method estimates, and the dragonfly algorithm (APT, MPT) surpassed it in this regard as well. These findings demonstrate the capability of using intelligent optimization algorithms, the dragonfly algorithm in this instance, with suitable tuning systems for the parameters in order to achieve improved and more reliable estimates for sophisticated financial models within developing markets.

**Declaration of competing interest**

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

**Funding information**

The authors declare that they have received no funding from any financial organization to conduct this research.

**References**

[1] E. Ferreira, J. Gil-Bazo, and S. Orbe, “Conditional beta pricing models: A nonparametric approach,” J. Bank. Finance, vol. 35, no. 12, pp. 3362–3382, 2011. https://doi.org/10.1016/j.jbankfin.2011.05.016.

[2] J. Green, J. R. M. Hand, and X. F. Zhang, “The characteristics that provide independent information about monthly stock returns,” Rev. Financ. Stud., vol. 30, no. 12, pp. 4389–4436, 2017. https://doi.org/10.1093/rfs/hhx019.

[3] L. Wang, X. L. Zheng, and S. Y. Wang, “A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem,” Knowl.-Based Syst., vol. 48, pp. 17–23, 2013. https://doi.org/10.1016/j.knosys.2013.03.009.

[4] A. Ang and D. Kristensen, “Testing conditional factor models,” J. Financ. Econ., vol. 106, no. 1, pp. 132–156, 2012. https://doi.org/ 10.1016/j.jfineco.2012.04.008.

[5] T. Škrinjarić, “Rolling regression CAPM on Zagreb stock exchange: Can investors profit from it?,” Econ. Rev. J. Econ. Bus., vol. 16, no. 2, pp. 7–22, 2018. [Online]. Available: https://hrcak.srce.hr/203758

[6] R. T. Baillie, F. Calonaci, and G. Kapetanios, “Hierarchical time-varying estimation of asset pricing models,” J. Risk Financ. Manag., vol. 15, no. 1, p. 14, 2022. https://doi.org/10.3390/jrfm15010014.

[7] W. T. Pan, “A new fruit fly optimization algorithm: taking the financial distress model as an example,” Knowl.-Based Syst., vol. 26, pp. 69–74, 2012. https://doi.org/10.1016/j.knosys.2011.07.007.

[8] J. Niu, W. Zhong, Y. Liang, N. Luo, and F. Qian, “Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization,” Knowl.-Based Syst., vol. 88, pp. 253–263, 2015. https://doi.org/10.1016/j.knosys.2015.07.027.

[9] L. Wu, C. Zuo, and H. Zhang, “A cloud model based fruit fly optimization algorithm,” Knowl.-Based Syst., vol. 89, pp. 603–617, 2015. https://doi.org/10.1016/j.knosys.2015.07.006.

[10] H. Z. Li, S. Guo, C. J. Li, and J. Q. Sun, “A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm,” Knowl.-Based Syst., vol. 37, pp. 378–387, 2013. https://doi.org/10.1016/j.knosys.2012.09.018.

[11] S.-M. Lin, "Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network," Neural Comput. Appl., vol. 22, no. 3–4, pp. 783–791, 2013. https://doi.org/10.1007/s00521-012-0791-2.

[12] J. Han, P. Wang, and X. Yang, "Tuning of PID controller based on fruit fly optimization algorithm," in Proc. IEEE Int. Conf. Mechatronics Autom., Chengdu, China, Aug. 2012, pp. 409–413. https://doi.org/10.1109/ICMA.2012.6282878.

[13] M. Liu and Z. H. Sun, "Application of the fruit fly optimization algorithm to an optimized neural network model in radar target recognition," Comput. Opt., vol. 45, no. 2, pp. 296–300, 2021. https://doi.org/10.18287/2412-6179-CO-789.

[14] D. Połap and M. Woźniak, "Detection of important features from images using heuristic approach," in Inf. Softw. Technol., ICIST 2017, Druskininkai, Lithuania, Oct. 2017, pp. 432–441. https://doi.org/10.1007/978-3-319-67642-5_37.

[15] J. Wu et al., "A novel ship classification approach for high resolution SAR images based on the BDA-KELM classification model," Int. J. Remote Sens., vol. 38, no. 23, pp. 6457–6476, 2017. https://doi.org/10.1080/01431161.2017.1346401.

[16] A. Tharwat, T. Gabel, and A. E. Hassanien, "Parameter optimization of support vector machine using dragonfly algorithm," in Adv. Intell. Syst. Inform., Springer, 2018, pp. 309–319. https://doi.org/10.1007/978-3-319-70139-4_30.

[17] S. Rakoth and J. Sasikala, "Multilevel segmentation of fundus images using dragonfly optimization," Int. J. Comput. Appl., vol. 164, no. 4, pp. 28–32, 2017. https://doi.org/10.5120/ijca2017914040.

[18] B. Hemamalini and V. Nagarajan, "Wavelet transform and pixel strength-based robust watermarking using dragonfly optimization," Multimed. Tools Appl., vol. 79, no. 13, pp. 8727–8746, 2020. https://doi.org/10.1007/s11042-018-6096-0.

[19] J. Kumaran and J. Sasikala, "Dragonfly optimization-based ANN model for forecasting India's primary fuels' demand," Int. J. Comput. Appl., vol. 164, no. 7, pp. 18–22, 2017. https://doi.org/10.5120/ijca2017914042.

[20] M. Yasen, N. Al-Madi, and N. Obeid, "Optimizing neural networks using dragonfly algorithm for medical prediction," in Proc. 8th Int. Conf. Comput. Sci. Inf. Technol. (CSIT), Amman, Jordan, Jul. 2018, pp. 71–76. https://doi.org/10.1109/CSIT.2018.8486227.

[21] W. Xiaoxi, "Do Anomalies Still Exist? Linear and Nonlinear Methods to Select Characteristics," Ph.D. dissertation, Univ. of Hong Kong, 2017. [Online]. Available: https://hub.hku.hk/handle/10722/250847.

[22] Y. S. Mezaal, H. T. Eyyuboglu, and J. K. Ali, "New dual band dual-mode microstrip patch bandpass filter designs based on Sierpinski fractal geometry," in 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT), 2013.https://doi.org/10.1109/ACCT.2013.55.

[23] Y. S. Mezaal, H. T. Eyyuboglu, and J. K. Ali, "Wide Bandpass and Narrow Bandstop Microstrip Filters based on Hilbert fractal geometry: design and simulation results," PLoS One, vol. 9, no. 12, p. e115412, 2014. https://doi.org/ 10.1371/journal.pone.0115412.