This paper is a replication study of the work conducted by Fangzheng Cheng, Tijun Fan, Dandan Fan, and Shanling Li.

Prediction of Oil Price Turning Points with LPPL and MPGA

B. Dufour J. Mourgues T. Charbonnier A. Remiat

January 18th, 2025

Crude oil is crucial to the global economy, with price volatility causing major economic and political disruptions. Accurately predicting price turning points is essential for strategic reserves and investment.

This paper presents an enhanced LPPL model, optimized via a multi-population genetic algorithm (MPGA), for improved forecasting accuracy.

Keywords of the study

- Financial bubbles detection
- Non-linear models estimation
- Heuristic & Genetic algorithms

Context of LPPL Models

- LPPL models aim to describe speculative bubbles in financial markets.
- **Key Assumption** : A bubble exhibits super-exponential growth with log-periodic oscillations as it approaches a critical point.
- **Motivation** : To capture the interplay between rational traders and noise traders in destabilizing the market.

Historical Use

- Empirical validation on commodities and equity indices.
- Interesting extensions on currencies and cryptocurrencies

Traders interplay : The Ising model

Each trader's position ($s_i = +1$ for long, $s_i = -1$ for short) is determined by :

$$s_i = \text{sign}\left(\sum_{j \in N(i)} K s_j + \sigma \epsilon_i + G\right),$$

where :

- $N(i)$: Trader i's network.
- $K > 0$: Coupling strength (influence of peers).
- $\sigma > 0$: Strength of idiosyncratic factors.
- $\epsilon_i \sim N(0, 1)$: Independent random shock.
- G : Global influence.

Agent-Based Foundations of LPPL Models

LPPL models are derived from agent-based interactions between :

- **Noise traders :** Exhibit herding behavior, amplifying trends.
- **Rational traders :** Attempt to exploit mispricings but may fail to stabilize the market.

Conditional Crash Probability

The conditional probability of a crash, $h(t)$, given that the bubble has not yet burst, represents the likelihood of a sudden collective sell-off :

$$h(t) = B(t_c - t)^{-b}, \quad b \in (0, 1).$$

This describes the increasing risk of market makers failing to absorb shocks as t approaches t_c.

Log-Periodic Power Law Equation

General LPPL Form

$$p(t) = A + B(T_c - t)^{\alpha} + C(T_c - t)^{\alpha}\cos(\omega \ln(T_c - t) + \phi)$$

- $p(t)$: Asset price or market indicator.
- t_c : Critical time (crash or turning point).
- A, B, C, ω, ϕ : Other parameters to be estimated.
- α : Critical exponent $(0 < \alpha < 1)$.

Interpretation

- Feedback loops create instability, while oscillations arise from trader interactions.
- Explains market crashes as endogenous events, not purely external shocks.

Derivation of linear parameters

- Linear parameters can be directly derived using a least squares approach for given nonlinear parameters :
- Definitions :

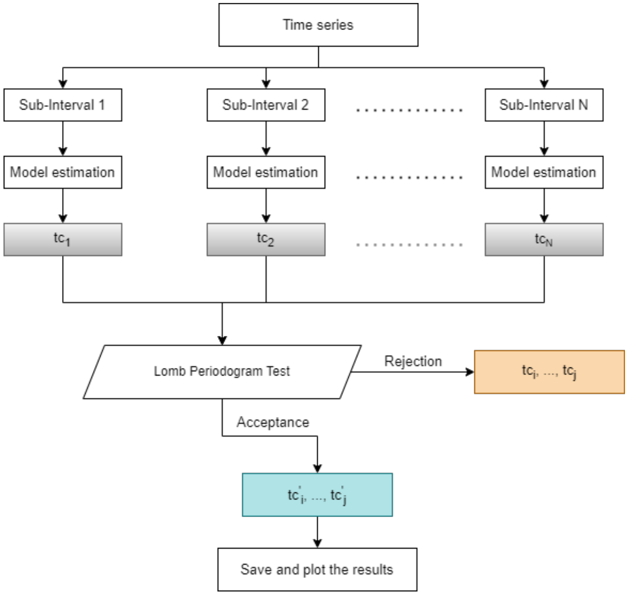
$$f_j = (T_c - t_j)^\alpha \quad \text{and} \quad g_j = (T_c - t_j)^\alpha \cos\left(\omega \ln(T_c - t_j) + \phi\right)$$

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \left(\mathrm{V}^T \mathrm{V}\right)^{-1} \mathrm{V}^T \mathrm{Y}$$

where :

$$\mathrm{V} = \begin{bmatrix} 1 & f_1 & g_1 \\ 1 & f_2 & g_2 \\ \vdots & \vdots & \vdots \\ 1 & f_J & g_J \end{bmatrix}, \quad \mathrm{Y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(J) \end{bmatrix}$$

Turning points prediction in practice



Subinterval Selection Strategy

- **Step 1 : Define the step size (δ) for subinterval starts :**
  - Compute δ as the maximum of :

$$(time_{end} - time_{start}) \times 0.75/(3 \text{ weeks})$$

and 3 weeks.

- **Step 2 : Loop over subinterval end times ($subinterval_{end}$) :**
  - Start from $time_{end}$, decrease by 1 week at each step.
  - Continue until $time_{end} - 6$ weeks.
- **Step 3 : Loop over subinterval start times ($subinterval_{start}$) :**
  - Start from $time_{start}$, increase by δ at each step.
  - Stop at $time_{end} - \frac{(time_{end} - time_{start})}{4}$.
- **Step 4 : Extract subintervals :**
  - Define each subinterval as $[subinterval_{start}, subinterval_{end}]$.
  - Extract these subintervals from the main sample $[time_{start}, time_{end}]$.

Why Heuristic Algorithms for LPPL ?

Non-Linear Nature of LPPL

- The LPPL model is highly **non-linear**, with parameters like ω, t_c, and m interacting in complex ways.
- Standard estimation methods (e.g., least squares, likelihood) fail to estimate the model.

Heuristic Algorithms

- **Definition :** Algorithms inspired by natural processes or optimization heuristics to explore large and complex solution spaces.
- Designed to :
  - Avoid local minima.
  - Handle high-dimensional, non-linear models effectively.
- Examples : Genetic algorithms, simulated annealing, particle swarm optimization.

Key Concepts :

- Inspired by **natural evolution** and genetic selection.
- Operates on a population of candidate solutions, evolving them through :
  - **Selection :** Fittest individuals are chosen for reproduction.
  - **Crossover :** Genetic material is combined to create offspring.
  - **Mutation :** Small random changes introduce diversity.

Why GAs for LPPL ?

- Effectively searches non-linear, multi-dimensional parameter spaces.
- Balances **exploration** (diverse solutions) and **exploitation** (refining good solutions).

Multi-Population Genetic Algorithm (MPGA)

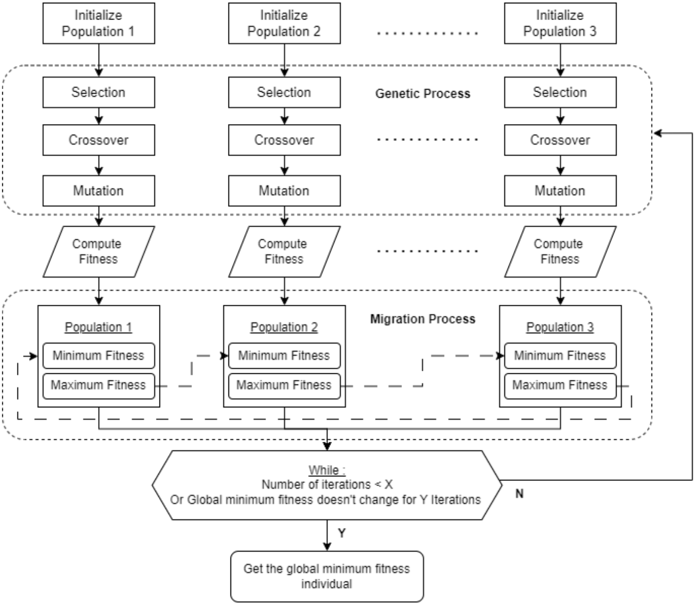
Principle of MPGA

- MPGA extends standard genetic algorithms by introducing **multiple sub-populations**.
- Each sub-population evolves independently, avoiding premature convergence to local minima.
- Periodic **migration of individuals** between populations introduces diversity and improves global search.

Advantages for LPPL Estimation

- **Avoids local minima** : Independent sub-populations explore different areas of the parameter space.
- **Improved convergence** : Migration ensures diversity and prevents stagnation.
- **Effective for LPPL** : Handles the complex, multi-modal structure of the LPPL model.

MPGA Process



Particle Swarm Optimization (PSO)

Principle of PSO

- PSO is inspired by **social behavior of swarms** (e.g., birds).
- It optimizes a function by iteratively moving a population of particles (candidate solutions) through the search space.
- Each particle adjusts its position based on :
  - **Personal Best (p_i) :** Best solution it has found so far.
  - **Global Best (g) :** Best solution found in the entire swarm.

PSO Algorithm Steps

1. Initialize particles with random positions and velocities.
2. Evaluate the fitness of each particle.
3. Update velocities and positions :

$$v_i(t+1) = wv_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Simulated Annealing (SA)

Principle of SA

- Simulated Annealing (SA) is inspired by the **annealing process** in metallurgy : Material is heated and slowly cooled to remove defects and reach a stable structure.
- SA mimics this by exploring the search space and accepting worse solutions to escape local minima.

SA Algorithm Steps

1. Start with an initial solution and a high "temperature" T.
2. Generate a new candidate by making a small random change.
3. Evaluate the fitness of the new solution :
    - If better, accept it, else accept it with probability :

$$P = \exp\left(-\frac{\Delta E}{T}\right),$$

where ΔE is the fitness difference.

Lomb Periodogram Analysis in LPPL Models

Purpose of the Analysis

- The Lomb periodogram is used to validate the turning points (t_c) predicted by the LPPL model.
- It checks if the frequency $\frac{\omega}{2\pi}$ derived from the LPPL fits matches the frequency of the periodic oscillations near the critical time t_c.

Why the Lomb Periodogram ?

- Handles **non-uniform time series**, common in financial data.
- Provides an **objective method** for identifying periodic components.
- Effective for detecting periodic oscillations in noisy environments.

Lomb Periodogram Method

Power Spectral Density

The power spectral density $P(f)$ is computed for the frequency series using the formula :

$$P(f) = \frac{1}{2\sigma^2}\left[\frac{\left(\sum_{j=1}^{J}(x_j-\bar{x})\cos\left(2\pi f(t_j-\tau)\right)\right)^2}{\sum_{j=1}^{J}\cos^2\left(2\pi f(t_j-\tau)\right)} + \frac{\left(\sum_{j=1}^{J}(x_j-\bar{x})\sin\left(2\pi f(t_j-\tau)\right)\right)^2}{\sum_{j=1}^{J}\sin^2\left(2\pi f(t_j-\tau)\right)}\right]$$

Where :

- $x_j = y_j - A - B(t_c - t_j)^a$
- $\bar{x}$ is the mean of x_j
- σ^2 is the variance of x_j.

Frequency Calculation in Lomb Periodogram

Offset time

The time offset t is calculated as :

$$\tau = \frac{1}{4\pi f} \arctan \left(\sum_{j=1}^{J} \sin(4\pi f t_j) \Big/ \sum_{j=1}^{J} \cos(4\pi f t_j) \right)$$

- With $tj = ln(t_c - t)^a$

Key Process

- Invalid frequencies in $P(f)$ are removed (e.g., values below a threshold or the most probable frequency).
- If no valid values remain, the predicted turning points are **rejected**.
- Otherwise, the frequency with the **maximum valid value** is used.

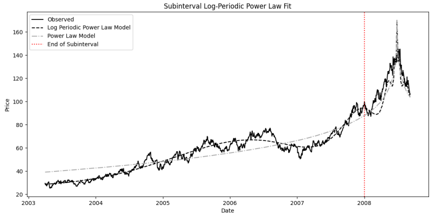
Validation of Turning Points with Lomb Periodogram

Validation Criteria

- The frequency from the Lomb periodogram is compared to $\frac{\omega}{2\pi}$ from MPGA.
- A turning point is valid if the difference between frequencies is less than 0.3.

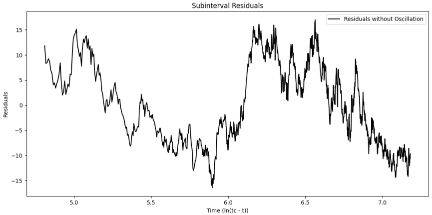
Conclusion

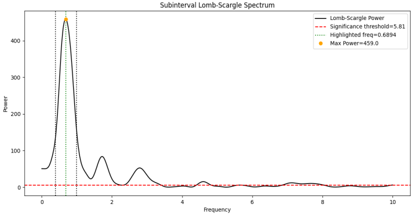
- Only turning points passing the Lomb periodogram test are **recorded as valid**.
- Ensures robustness in predicting critical times (t_c) in LPPL analysis.



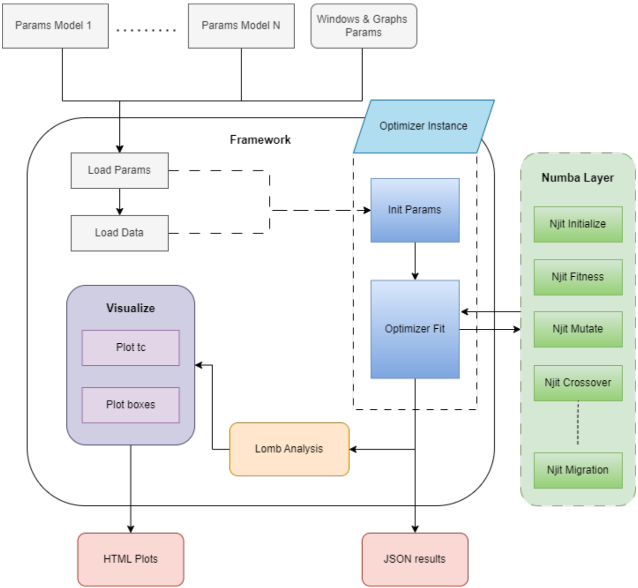
$$y(t) = \underbrace{A + B(T_c - t)^{\alpha}}_{\textbf{Power Law}} + \underbrace{C(T_c - t)^{\alpha}\cos(\omega \ln(T_c - t) + \phi)}_{\textbf{Log-Periodic}}$$

$$\textbf{Target frequency} = \frac{\omega}{2\pi}$$





Framework Organization



Real Turning Points

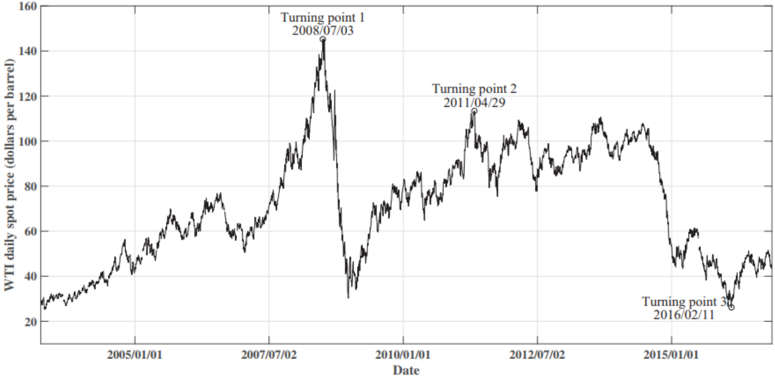


Figure – The WTI spot price between April 1, 2003 and November 14, 2016.

MPGA predicted results daily data

Figure – Prediction result of daily data of April 1, 2003 to January 2, 2008

MPGA predicted results daily data

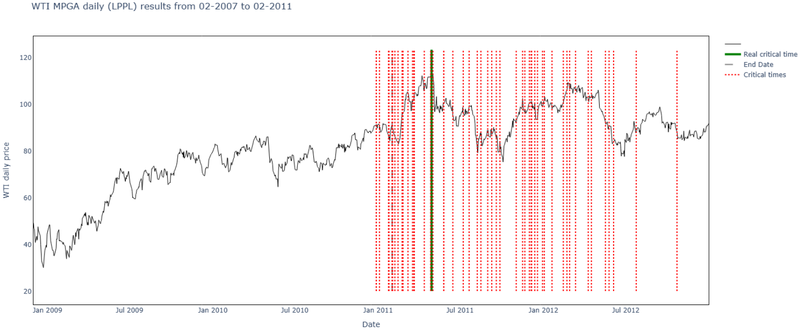


Figure – Prediction result of daily data of February 1, 2007 to February 1, 2011

MPGA predicted results daily data

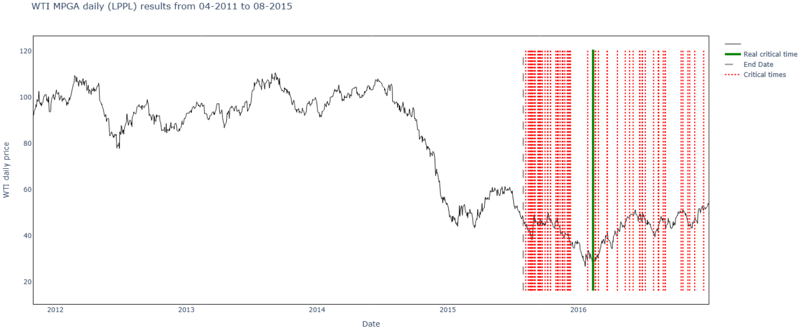


Figure – Prediction result of daily data of April 29, 2011 to August 1, 2015

MPGA predicted results weekly data

Figure – Prediction result of weekly data of April 1, 2003 to January 2, 2008

MPGA predicted results weekly data

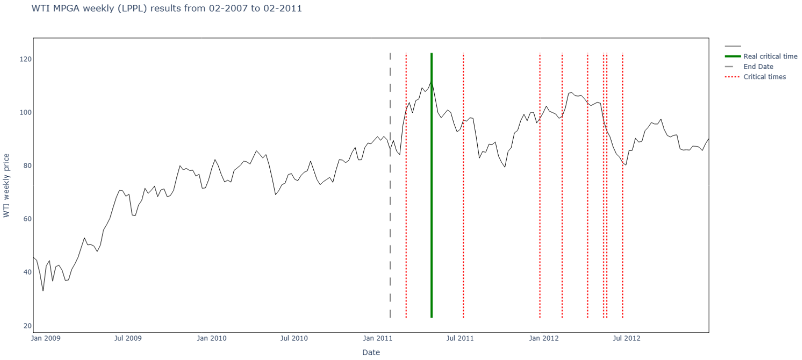


Figure – Prediction result of weekly data of February 1, 2007 to February 1, 2011

MPGA predicted results weekly data

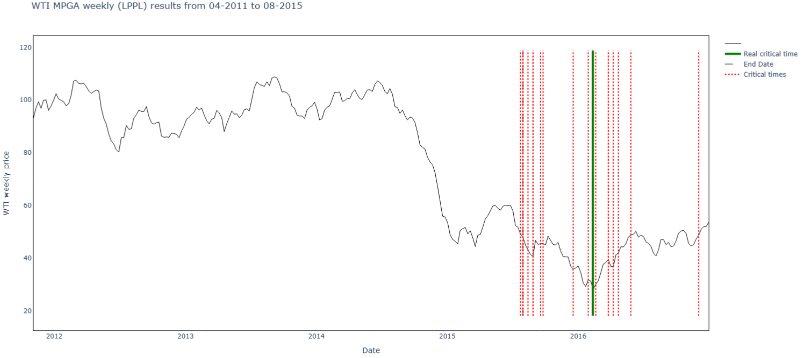
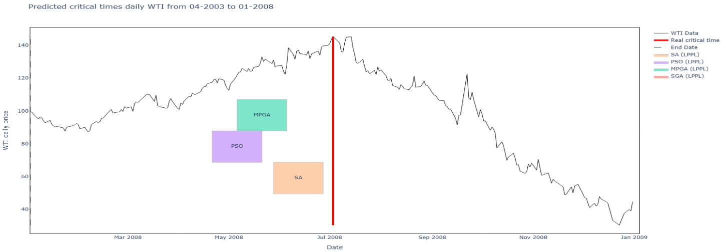


Figure – Prediction result of weekly data of April 29, 2011 to August 1, 2015

Comparison of Predicted Results (daily)



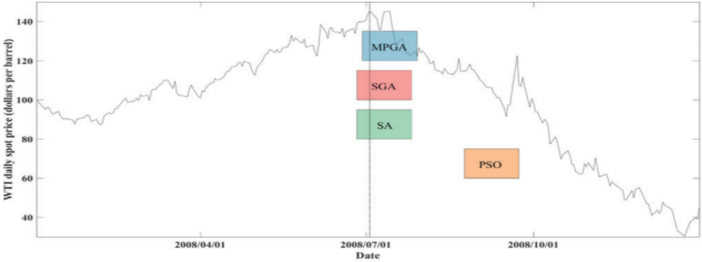
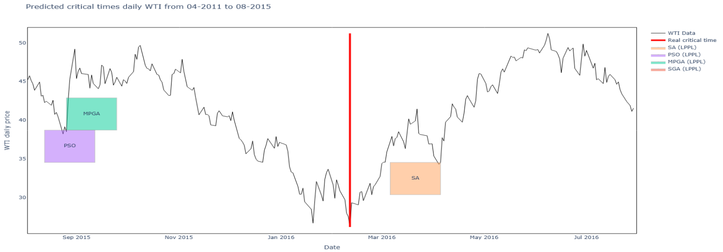


Figure – Comparison of daily data of April 1, 2003 to January 2, 2008.

Comparison of Predicted Results (daily)

Figure – Comparison of daily data of February 1, 2007 to February 1, 2011.

Comparison of Predicted Results (daily)



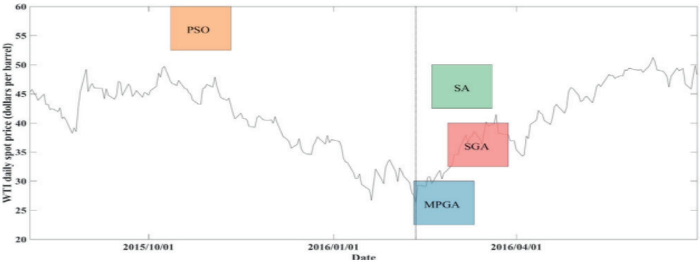


Figure – Comparison of daily data of April 29, 2011 to August 1, 2015.

Comparison of Predicted Results (weekly)

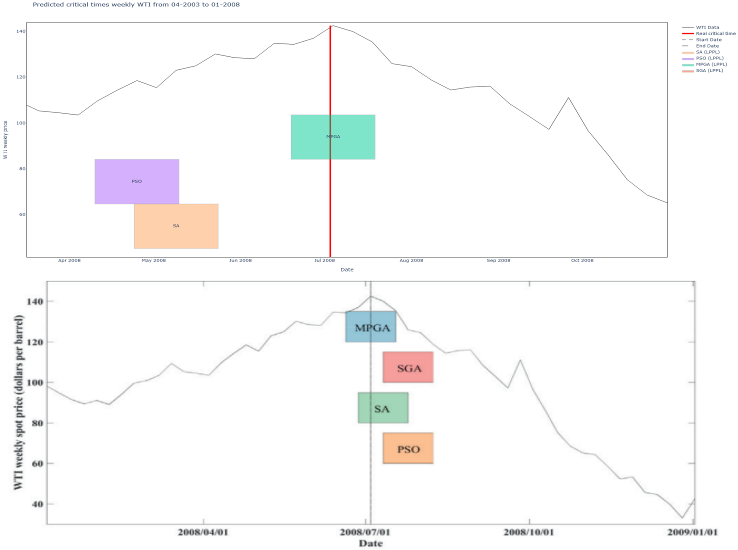


Figure – Comparison of weekly data of April 1, 2003 to January 2, 2008.

Comparison of Predicted Results (weekly)

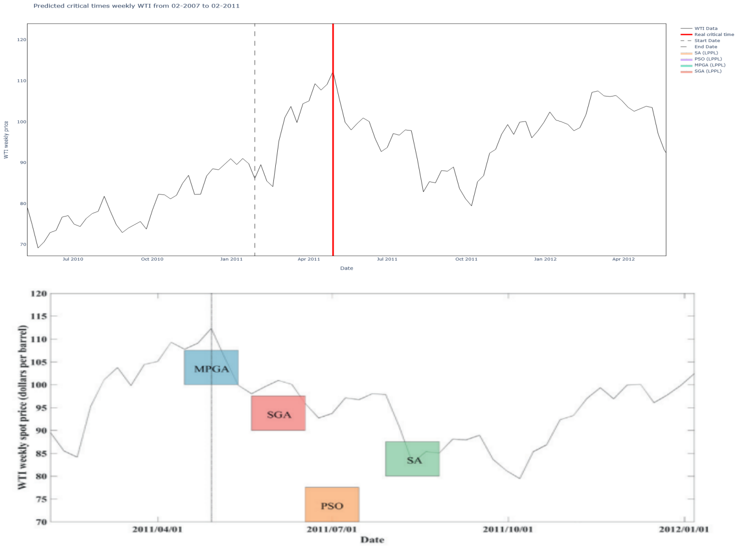


Figure – Comparison of weekly data of February 1, 2007 to February 1, 2011.

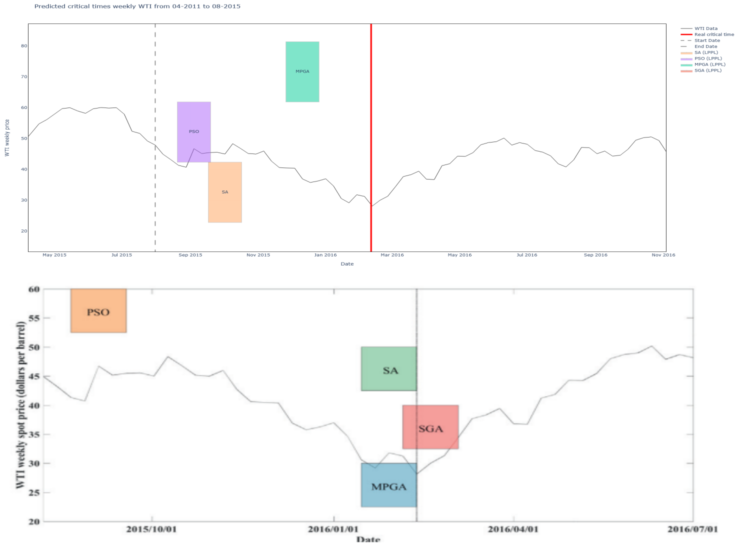


Figure – Comparison of weekly data of April 29, 2011 to August 1, 2015.

Table – Comparison of prediction results of different heuristic algorithms.

Optimization Method	**Turning Point 1**	**Turning Point 2**	**Turning Point 3**
Historical Turning Point	**2008/07/03**	**2011/04/29**	**2016/02/11**
30 days optimized by MPGA	False (2008/05/06–2008/06/05)	False (2019/11/04–2019/12/04)	False (2015/08/26–2015/09/25)
4 weeks optimized by MPGA	True (2008/06/19–2008/07/19)	False (2012/07/26–2012/08/25)	False (2015/11/26–2015/12/26)
30 days optimized by SGA	False (2010/06/13–2010/07/13)	False (2019/07/11–2019/08/10)	False (2018/07/02–2018/08/01)
4 weeks optimized by SGA	False (2010/05/20–2010/06/19)	False (2013/12/19–2014/01/18)	False (2019/11/14–2019/12/14)
30 days optimized by SA	False (2008/05/25–2008/06/25)	False (2019/11/14–2019/12/14)	False (2016/02/22–2016/03/22)
4 weeks optimized by SA	False (2008/04/25–2008/05/25)	False (2016/06/02–2016/07/02)	False (2015/09/13-2015/10/12)
30 days optimized by PSO	False (2008/04/21–2008/05/21)	False (2021/01/11-2021/02/10)	False (2015/08/13-2015/09/12)
4 weeks optimized by PSO	False (2008/04/10–2008/05/10)	False (2013/11/28–2013/12/28)	False (2015/08/20–2015/09/19)

USO (United State Oil Fund) Analysis

Figure – Daily Price of USO (red) and WTI (blue) from April 10, 2006 to November 14, 2016

USO Predicted Turning Points

Figure – Comparison of daily (upper graph) and weekly data (lower graph) of April 29, 2011 to August 1, 2015. USO

USO Predicted Turning Points

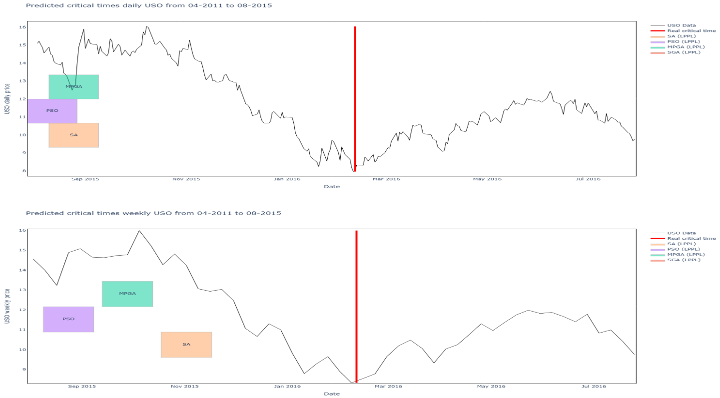


Figure – Comparison of daily (upper graph) and weekly data (lower graph) of April 29, 2011 to August 1, 2015. USO

Conclusion on the paper replication

- Significance of selected historical turning points
- Estimation algorithms hard to calibrate (many extra-parameters or method choices)
- Validation of turning points subject to methodological choices

Extensions : Firefly Algorithm

Overview

- Inspired by firefly **swarm behavior**, ideal for optimization.
- Effective for **multi-objective, nonlinear problems**.
- Utilizes **global communication** to balance exploration and exploitation.

Key Steps of the Algorithm

At each iteration, fireflies are updated based on :

- **Attraction :** Fireflies move toward brighter ones based on their light intensity.
- **Light Intensity :** Determined by the objective function (RSS for LPPL).
- **Distance Effect :** Attraction decreases with distance due to light absorption.

Extensions : Tabu Search Algorithm

Overview

- Tabu Search is a **metaheuristic algorithm** designed to solve combinatorial and nonlinear optimization problems.
- It explores the solution space by allowing moves that do not always improve the objective function at every step.
- Incorporates **memory structures** (tabu list) to avoid revisiting previously explored solutions.

Main Steps of the Algorithm

- **Initialization :** Start with initial solution and empty tabu list.
- **Move Selection :** Choose the best neighbor, subject to the tabu list constraints.
- **Update :** Accept the new solution if it's better.
- **Tabu List Update :** Add the current solution to the tabu list to prevent revisiting.

Extensions : Results new algorithms

Figure – Comparison of daily (upper) and weekly (lower) data of April 2003 to January 2008 on all models.

Extensions : Results new algorithms

Figure – Comparison of daily (upper) and weekly (lower) data of April 2011 to August 2015 on all models.

Main Idea

- Reduce interdependence between **phase** (ϕ) and **angular log-frequency** (ω).
- Decrease the number of **nonlinear parameters** in the LPPL equation.

Reformulation Using New Parameters

- LPPL (classic model) :

$$p(t) = A + B(t_c - t)^\alpha + C(t_c - t)^\alpha \cos(\omega \ln(t_c - t) + \phi)$$

- LPPLS (simplified model) :

$$p(t) = A + B(t_c - t)^\alpha + C_1(t_c - t)^\alpha \cos(\omega \ln(t_c - t)) + C_2(t_c - t)^\alpha \sin(\omega \ln(t_c - t))$$

Nelder-Mead Algorithm

Why Nelder-Mead for LPPL ?

- Suitable for **nonlinear, multidimensional** optimization problems.
- Works well without needing gradient information, ideal for complex LPPL functions.
- Efficient in lower dimensions, (optimizing t_c, α, ω in LPPLS).

Steps of the Algorithm

At each iteration, the simplex is modified using :

- **Reflection** : Reflects the worst point across the centroid.
- **Expansion** : Tries to expand along the reflection direction.
- **Contraction** : Shrinks the simplex towards the best point if no improvement.
- **Reduction** : Reduces all points towards the best if the simplex becomes too small.

Extensions : Results with LPPLS

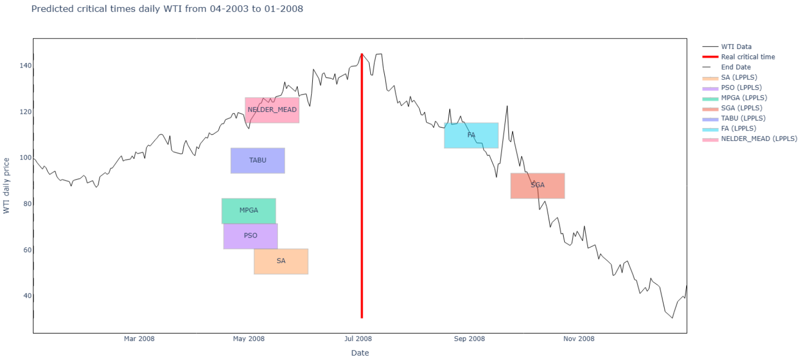


Figure – Comparison of daily data of April 1, 2003 to January 2, 2008.

Extensions : Results with LPPLS

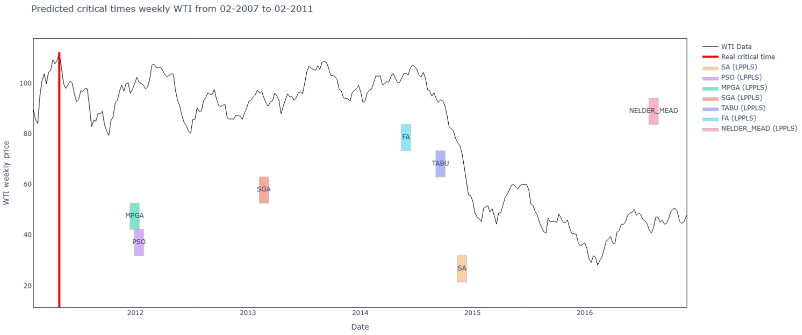


Figure – Comparison of weekly data of February 1, 2007 to February 1, 2011.

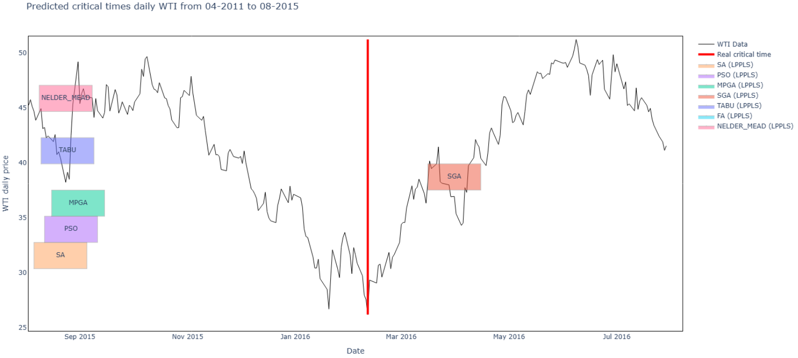


Figure – Comparison of daily data of April 29, 2011 to August 1, 2015.

Extensions : Turning point prediction on the SP500

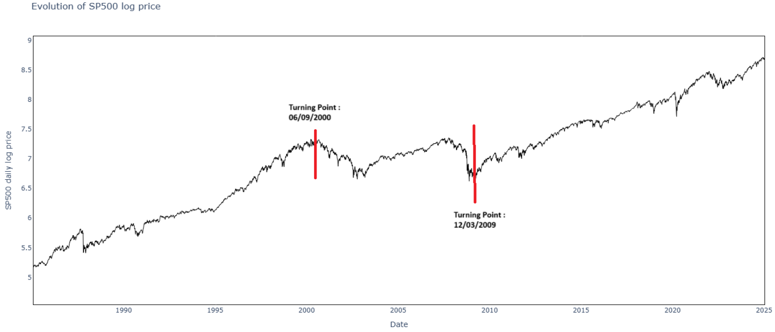


Figure – SP500 daily log prices

Extensions : Turning point prediction on the SP500

Figure – Comparison of daily (upper) and weekly (lower) data using LPPLS from march 1995 to march 2000.

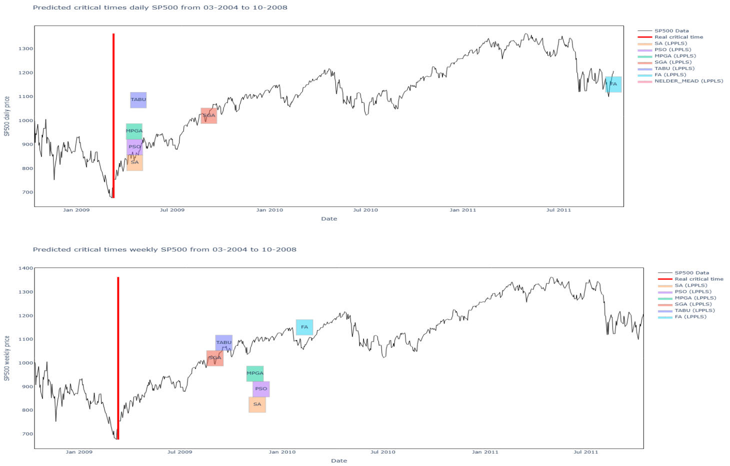


Figure – Comparison of daily (upper) and weekly (lower) data using LPPLS from march 2004 to october 2009.

Extensions : Turning point prediction on the BTC

Figure – BTC daily log prices.

Extensions : Turning point prediction on the BTC

Figure – Prediction result of LPPL daily data of January 2013 to November 2017

Extensions : Turning point prediction on the BTC

Figure – Prediction result of LPPL daily data of Mars 2018 to September 2021

Extensions : Turning point prediction on the BTC

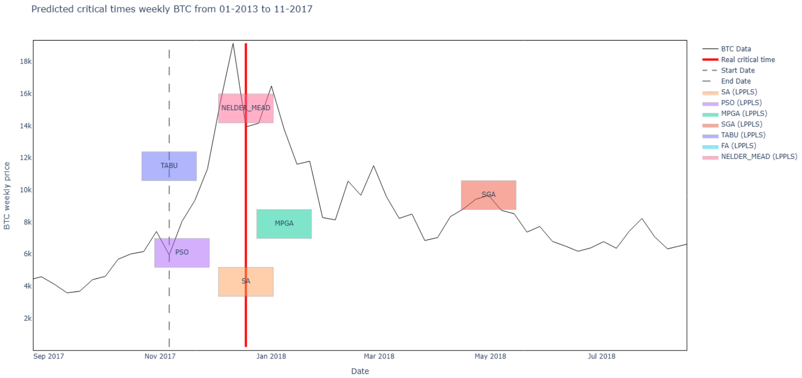


Figure – Prediction result of LPPLS weekly data of January 2013 to November 2017

Extensions : Turning point prediction on the BTC

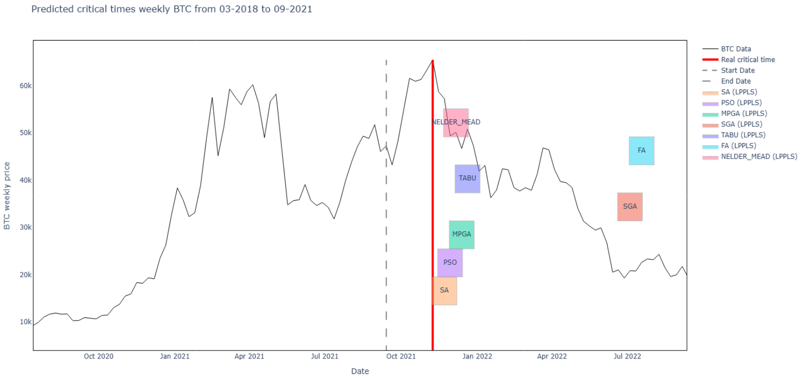


Figure – Prediction result of LPPLS weekly data of Mars 2018 to September 2021

Extensions : Bubble on BTC ?

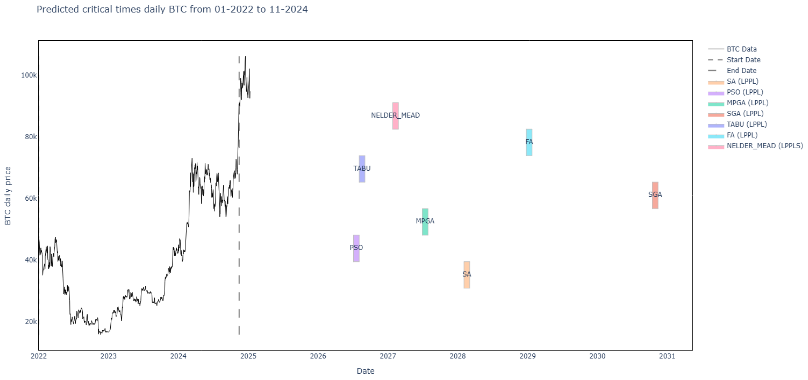


Figure – Next Prediction of turning point on BTC from january 2022

Extensions : Trading Strategies on BTC

- Uses LPPS model to identify significant turning points.
- Positioned long structurally.
- **Short Position :** Enter when a significant turning point is predicted within 30 days.
- **Exit Short :** Close short position after 30 days or when market conditions change.
- **This strategy helps avoid periods of intense downturns and capitalize on positive market movements.**

Extensions : Trading Strategies on BTC

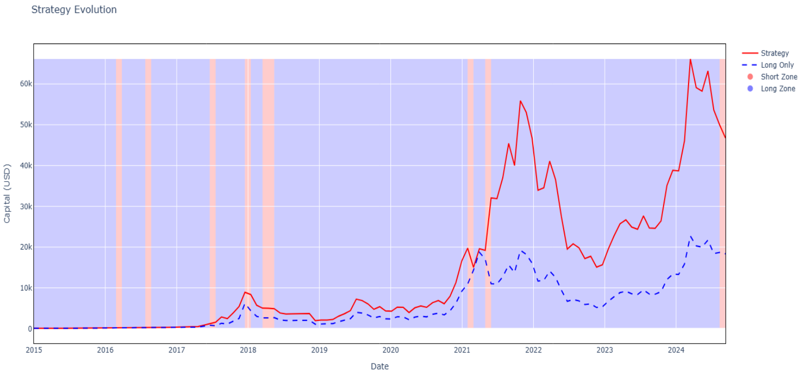


Figure – Trading Strategy results from January 2015 to September 2024

Conclusion

- MPGA consistently demonstrates superior performance, confirming its status as the most effective algorithm in this context.
- Notable and promising results were achieved with the combination of LPPLS and the Nelder-Mead optimization method, highlighting its potential for further exploration.
- The approach proves to be versatile and applicable across various asset classes, including cryptocurrencies and equities

Appendix : Turning point prediction on the SSE

Figure – SSE daily log prices.

Extensions : Turning point prediction on the SSE

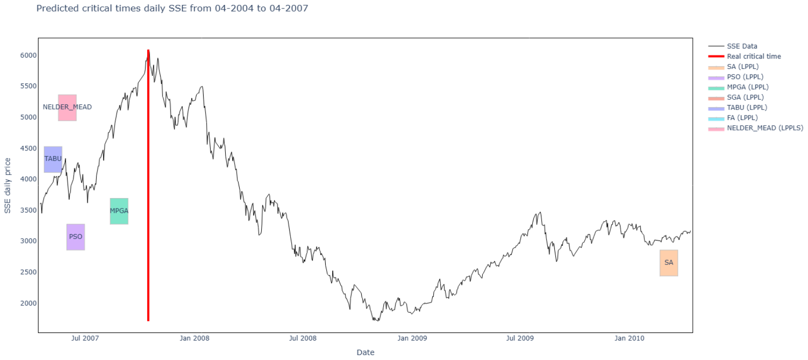


Figure – Comparison of daily data using LPPL from april 2004 to april 2007.