Part I : Trading strategy

For the sourcing of the data, we are going to use the FX data uploaded on KEATS.

First of all, let's **break down the format of the data provided on KEATS :**

We have two tick data files (*DAT_ASCII_EURUSD_T_201912 & DAT_ASCII_EURGBP_T_201606*) representing tick data for the EUR/GBP and the EUR/USD currency pair.

The **timestamp** is such as :

- YYYY: Year
- MM: Month
- DD: Day
- HH: Hour
- MM: Minute
- SS: Second
- sss: Millisecond

After the timestamp, there are **three comma-separated values**:

- Bid price (the price at which buyers are willing to buy)
- Ask price (the price at which sellers are willing to sell)
- Volume (the trading volume at that tick)

Example : "20191201 170003296,1.102290,1.102470,0" represents a tick at December 1st, 2019, 17:00:03.296, with bid price 1.102290, ask price 1.102470, and volume 0.

We also have two other files for each tick data file (*DAT_ASCII_EURUSD_T_201912 & DAT_ASCII_EURGBP_T_201606*), listing the gaps found between timestamps in the tick data. They basically indicate irregularities or missing data points in the tick data, which is for us an helpful information for preprocessing and handling missing data appropriately in order to implement our trading strategy.

Pseudo code of the trading strategy

Explanations

1 - Initialization phase :

- We load the tick data from the given CSV files representing EUR/USD and EUR/GBP currency pairs
- We fetch the additional files containing information about gaps in the data for EUR/USD and EUR/GBP

- We extract timestamp and price details from each row of our loaded data for both currency pairs
- We set up initial values for strategy parameters like the lookback period, the entry and exit thresholds, and the position size

2 - Data dissection with intrinsic time framework :

- We compute the intrinsic time for each tick based on its timestamp, by converting timestamps into a continuous time format
- We check for any gaps in the data and manage them appropriately to ensure data continuity
- We divide the tick data into intervals based on intrinsic time, to facilitate analysis within discrete time segments

3 - Strategy execution :

- We iterate through each time interval and we implement the trading strategy within each interval (by considering initial strategy parameters to make buy/sell decisions)

4 - Strategy performance assessment:

- We use tools to visualize the results of our strategy
- We evaluate the strategy's performance using metrics such as profitability, drawdown, and Sharpe ratio across both EUR/USD and EUR/GBP pairs.
- We conduct sensitivity analysis on strategy parameters to evaluate its robustness across various parameter values
- We visualize strategy results and analyze its behavior under diverse market conditions for both currency pairs

Contextualization and core concepts

In the context of high-frequency trading (HFT), the **intrinsic time framework** is a way to measure time that is independent of the clock time. Indeed, instead of using conventional time units like seconds or minutes, intrinsic time represents the time elapsed since the start of trading activity.

The **lookback period** is a parameter used in technical analysis and trading strategies : it defines the number of historical data points or time periods to consider when making trading decisions. In the context of our code, the lookback period determines the historical window over which price changes are analyzed to generate trading signals. Using a longer lookback period may capture more significant price trends but can also introduce more lag in the signals.

Entry and exit thresholds are criteria used to trigger trading signals in a strategy. They define the minimum price change required to initiate a trade (entry) or close an existing position (exit). In our code, the entry threshold is the minimum absolute price change required to trigger a buy or sell signal. If the absolute price change exceeds this threshold, a signal is generated to enter a position. Similarly, the exit threshold is used to close positions when the price moves in the opposite direction by a certain amount.

In [2]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import re

def parse_gap_data(file_path):
    gaps = []
    with open(file_path, 'r') as file:
        lines = file.readlines()
        for line in lines:
            match = re.search(r'Gap of (\d+)s found between (\d+) and (\d+)', line)
            if match:
                gap_info = {
                    'duration': int(match.group(1)),
                    'start_time': match.group(2),
                    'end_time': match.group(3)
                }
                gaps.append(gap_info)
    return gaps

def handle_gaps(data, gap_data):
    # Defining the logic to handle data gaps by using the parsed gap information
    for gap in gap_data:
        start_time = pd.to_datetime(gap['start_time'], format='%Y%m%d%H%M%S')
        end_time = pd.to_datetime(gap['end_time'], format='%Y%m%d%H%M%S')
        # Fill or remove data within the gap period
        data.loc[(data['Timestamp'] >= start_time) & (data['Timestamp'] <= end_time

# Loading EUR/USD tick data from CSV file
eurusd_data = pd.read_csv(r"C:\Users\rapha\OneDrive\Bureau\UK\MSc Comp. Fin\S2\High
eurgbp_data = pd.read_csv(r"C:\Users\rapha\OneDrive\Bureau\UK\MSc Comp. Fin\S2\High

# Converting timestamp column to datetime format
eurusd_data['Timestamp'] = pd.to_datetime(eurusd_data['Timestamp'], format='%Y%m%d
eurgbp_data['Timestamp'] = pd.to_datetime(eurgbp_data['Timestamp'], format='%Y%m%d

# Calculation of the intrinsic time (eg. milliseconds since the start of trading)
eurusd_data['Intrinsic_Time'] = (eurusd_data['Timestamp'] - eurusd_data['Timestamp'
eurgbp_data['Intrinsic_Time'] = (eurgbp_data['Timestamp'] - eurgbp_data['Timestamp'

# Loading of the gap data
gap_eurusd = parse_gap_data(r"C:\Users\rapha\OneDrive\Bureau\UK\MSc Comp. Fin\S2\Hi
gap_eurgbp = parse_gap_data(r"C:\Users\rapha\OneDrive\Bureau\UK\MSc Comp. Fin\S2\Hi

# Handling data gaps
handle_gaps(eurusd_data, gap_eurusd)
handle_gaps(eurgbp_data, gap_eurgbp)
```

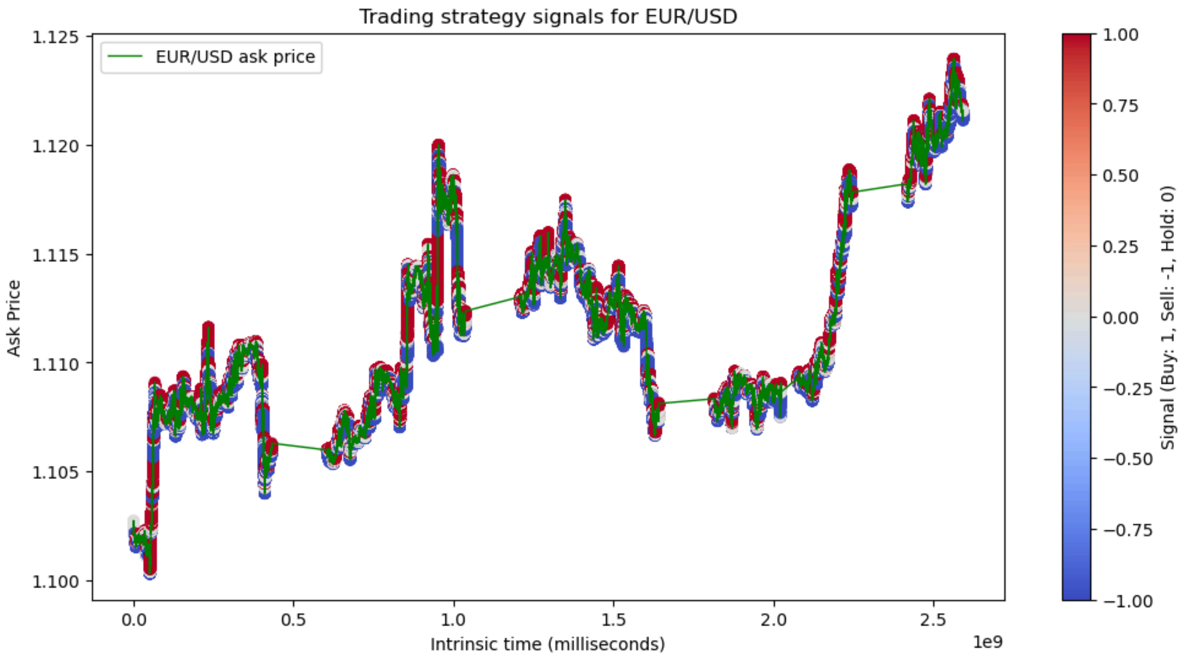
In [4]:
```python
# Application of the trading strategy
def apply_strategy(data, lookback_period, entry_threshold):
    signals = []
    positions = []
    for i in range(len(data)):
        if i < lookback_period:
            signals.append(0)  # No signal during the initial lookback period
            positions.append(0)  # No position during the initial lookback period
        else:
            price_change = data['Ask'].iloc[i] - data['Ask'].iloc[i - lookback_peri
            if abs(price_change) > entry_threshold:
                if price_change > 0:
```

```
                    signals.append(1)  # Buy signal
                else:
                    signals.append(-1)  # Sell signal
            else:
                signals.append(0)  # No signal

            if len(positions) > 0:
                if signals[-1] == 1 and positions[-1] <= 0:
                    positions.append(1)  # Buy
                elif signals[-1] == -1 and positions[-1] >= 0:
                    positions.append(-1)  # Sell
                else:
                    positions.append(0)  # Hold
    return signals, positions

# Define strategy parameters
lookback_period = 1000
entry_threshold = 0.0001

eurusd_signals, eurusd_positions = apply_strategy(eurusd_data, lookback_period, ent
eurgbp_signals, eurgbp_positions = apply_strategy(eurgbp_data, lookback_period, ent
```

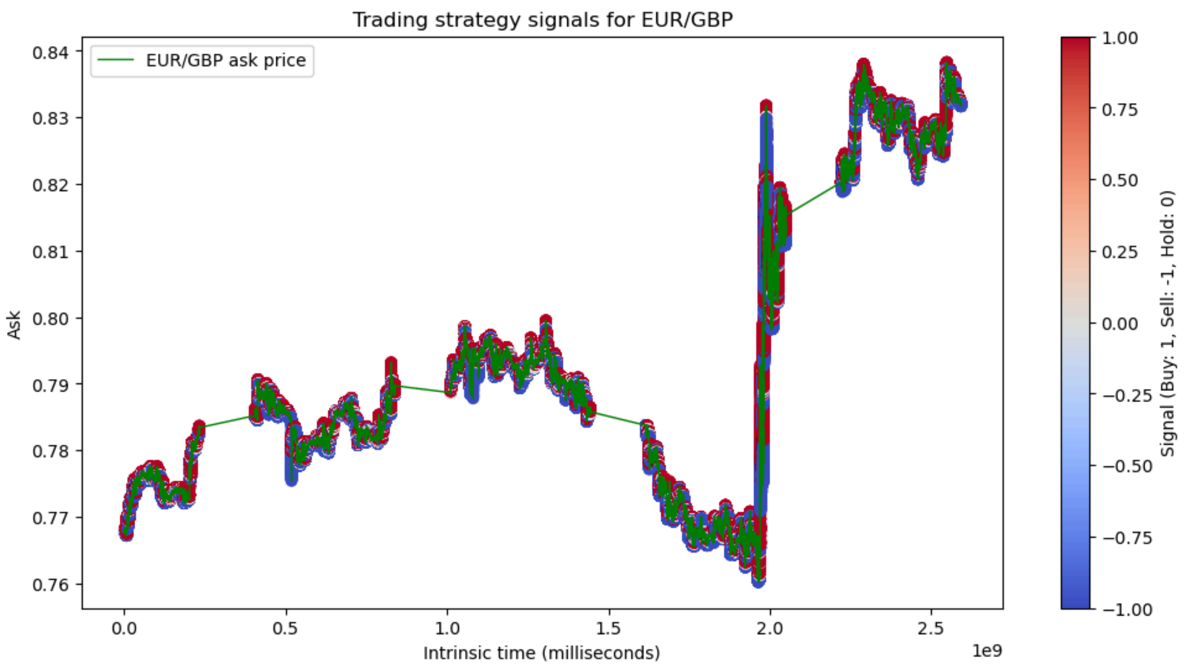
Part II : Testing of the strategy

As we explained previously, we are using the two tick data files (EUR/USD and EUR/GBP) from KEATS to backtest our trading strategy. We already loaded the given data, defined the functions and the overall code for our strategy, and we just have to run the strategy on our two datasets.

In [7]:
```
# EUR/USD signals and position visualization
plt.figure(figsize=(12, 6))
plt.plot(eurusd_data['Intrinsic_Time'], eurusd_data['Ask'], label='EUR/USD ask pric
plt.scatter(eurusd_data['Intrinsic_Time'], eurusd_data['Ask'], marker='o', c=eurusd
plt.xlabel('Intrinsic time (milliseconds)')
plt.ylabel('Ask Price')
plt.title('Trading strategy signals for EUR/USD')
plt.legend()
plt.colorbar(label='Signal (Buy: 1, Sell: -1, Hold: 0)')
plt.show()
```



In [6]:
```
# EUR/GBP signals and position visualization
plt.figure(figsize=(12, 6))
plt.plot(eurgbp_data['Intrinsic_Time'], eurgbp_data['Ask'], label='EUR/GBP ask pric
plt.scatter(eurgbp_data['Intrinsic_Time'], eurgbp_data['Ask'], marker='o', c=eurgbp
plt.xlabel('Intrinsic time (milliseconds)')
plt.ylabel('Ask')
plt.title('Trading strategy signals for EUR/GBP')
plt.legend()
plt.colorbar(label='Signal (Buy: 1, Sell: -1, Hold: 0)')
plt.show()
```



Discussion about the empirical results and suggestions

In order to discuss in an accurate way the empirical results of our strategy and make valid suggestions, we first need to find ways of assessing the efficiency of our strategy thanks to several metrics.

In our case, we're going to use three metrics : Profit and Loss, Sharpe ratio and Maximum drawdown.

- **Profit and Loss (P&L)** : measures the overall profitability of the strategy by calculating the *total profit or loss generated by executing the strategy over the backtested period*

In [4]:
```
import numpy as np

# Function to calculate PnL
def calculate_pnl(data, positions):
    pnl = pd.Series(0.0, index=data.index)
    for i in range(1, len(data)):
        pnl.iloc[i] = pnl.iloc[i - 1] + positions[i - 1] * (data['Ask'].iloc[i] - d
    return pnl

eurusd_pnl = calculate_pnl(eurusd_data, eurusd_positions)
eurgbp_pnl = calculate_pnl(eurgbp_data, eurgbp_positions)
```

- **Sharpe ratio** : measures the risk-adjusted return of the strategy by indicating the *excess return generated per unit of risk*

It it such as :

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

With :

R_p the expected portfolio return

R_f the risk free rate

σ_p the standard deviation of the portfolio

In [ ]:
```
# Function to calculate Sharpe Ratio
def calculate_sharpe_ratio(pnl):
    daily_returns = pnl.pct_change().dropna()
    sharpe_ratio = np.sqrt(252) * (daily_returns.mean() / daily_returns.std())
    return sharpe_ratio

sharpe_ratio_eurusd = calculate_sharpe_ratio(eurusd_pnl)
sharpe_ratio_eurgbp = calculate_sharpe_ratio(eurgbp_pnl)
```

- **Maximum drawdown** : measures the largest peak-to-trough decline in the equity curve of the strategy, it is the *maximum loss incurred by the strategy during the backtested period*

It it such as :

$$Maximum\ Drawdown = \max\left(\frac{P_i - P_j}{P_i}\right)$$

<center

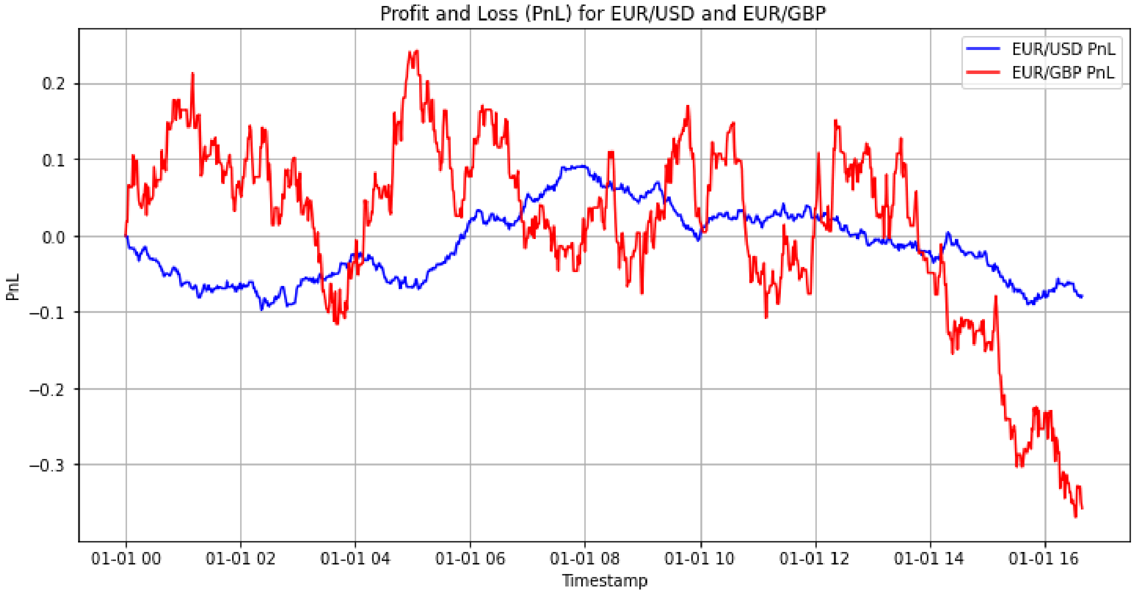
With :

P_i the trough (lowest value) of the equity curve following the peak

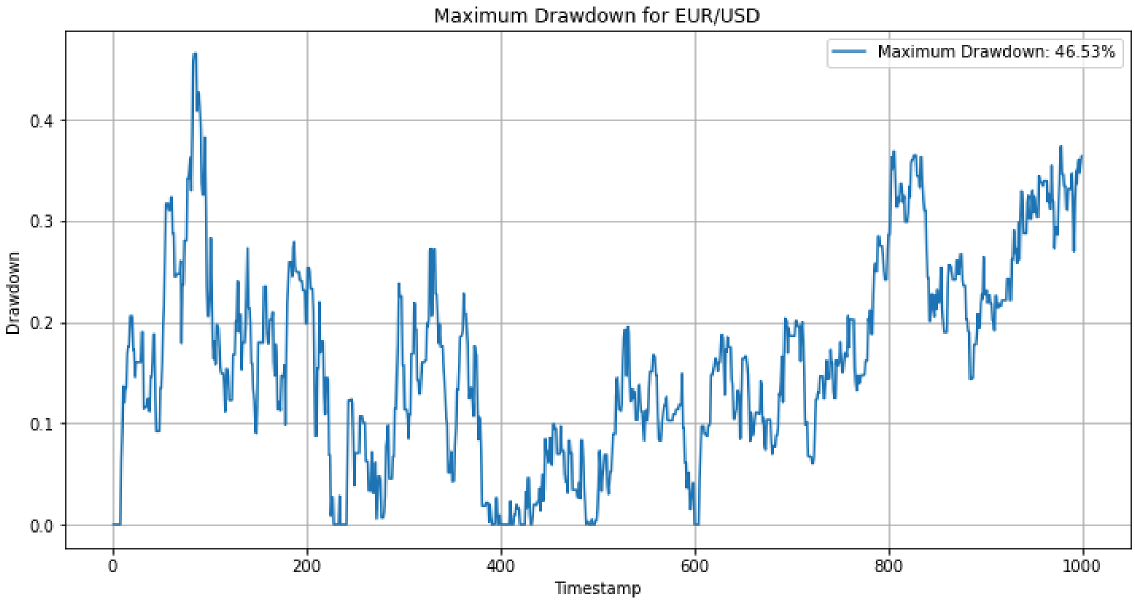
P_j the peak (highest value) of the equity curve up to a certain point

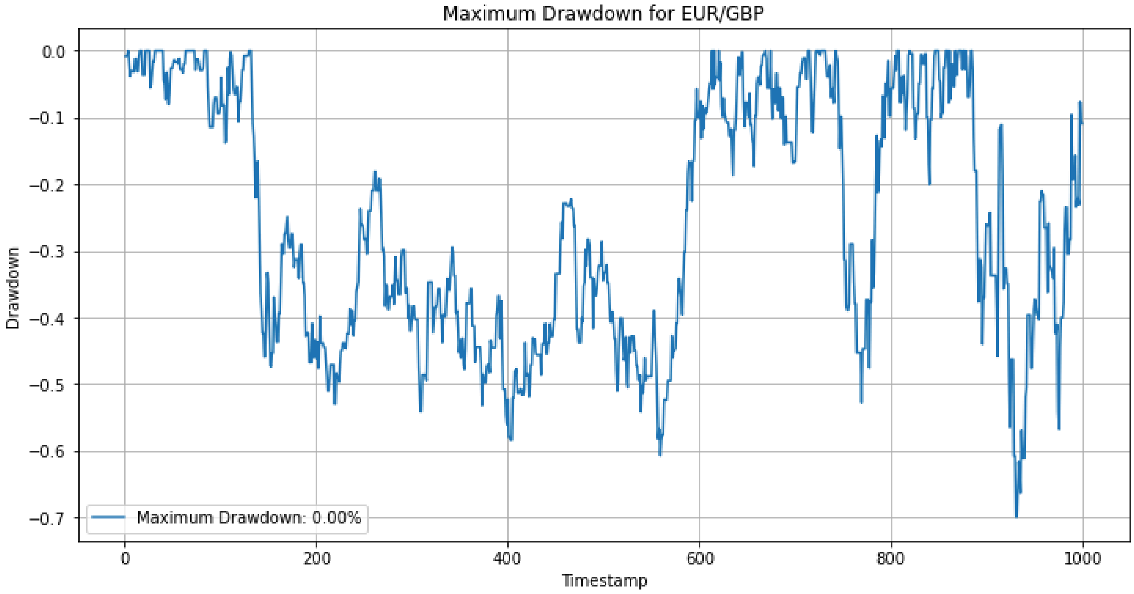
In [ ]:
```
# Function to calculate Maximum Drawdown
def calculate_max_drawdown(pnl):
    peak = pnl.iloc[0]
    max_drawdown = 0
    for i in range(1, len(pnl)):
        if pnl.iloc[i] > peak:
            peak = pnl.iloc[i]
        else:
            drawdown = (peak - pnl.iloc[i]) / peak
            if drawdown > max_drawdown:
                max_drawdown = drawdown
    return max_drawdown

max_drawdown_eurusd = calculate_max_drawdown(eurusd_pnl)
max_drawdown_eurgbp = calculate_max_drawdown(eurgbp_pnl)
```

In [ ]:
```
print("Sharpe ratio for EUR/USD:", sharpe_ratio_eurusd)
print("Sharpe ratio for EUR/GBP:", sharpe_ratio_eurgbp)
print("Maximum drawdown for EUR/USD:", max_drawdown_eurusd)
print("Maximum drawdown for EUR/GBP:", max_drawdown_eurgbp)

# Plot PnL for EUR/USD and EUR/GBP
plt.figure(figsize=(12, 6))
plt.plot(eurusd_data['Timestamp'], eurusd_pnl, color='blue', label='EUR/USD PnL')
plt.plot(eurgbp_data['Timestamp'], eurgbp_pnl, color='red', label='EUR/GBP PnL')
plt.xlabel('Timestamp')
plt.ylabel('PnL')
plt.title('Profit and Loss (PnL) for EUR/USD and EUR/GBP')
plt.legend()
plt.grid(True)
plt.show()


# Plot maximum drawdown for EUR/USD and EUR/GBP
plt.figure(figsize=(10, 5))
plt.bar(['EUR/USD', 'EUR/GBP'], [max_drawdown_eurusd, max_drawdown_eurgbp], color=[
plt.title('Maximum drawdown')
plt.xlabel('Currency pair')
plt.ylabel('Maximum drawdown')
plt.ylim([0, max(max_drawdown_eurusd, max_drawdown_eurgbp) + 0.1])
plt.grid(True)
plt.show()
```




```
Sharpe ratio for EUR/USD: 0.33645921468111706
Sharpe ratio for EUR/GBP: -0.3533631499638172
Maximum drawdown for EUR/USD: 0.4653396137601056
Maximum drawdown for EUR/GBP: 0
```





Based on the provided data and metrics, let's analyze the efficiency of our trading strategy:

1. The fact that the **PnL starts positive** indicates that the strategy may have performed well initially, generating profits. However, the subsequent decline in PnL suggests that the strategy may not have been able to sustain its profitability over time. We could explain it with various factors : changing market conditions, our strategy's inability to adapt to different market regimes...

2. For EUR/USD, our Sharpe ratio of 0.33 indicates a positive risk-adjusted return, suggesting that the strategy has generated returns that outweigh the associated risks. However, we can still consider a Sharpe ratio of 0.33 relatively low, and this indicates we can improve in terms of risk management and return generation.

3. For EUR/GBP, our Sharpe ratio of (-0.35) suggests that the strategy has failed to generate positive returns relative to the level of risk taken. The strategy may not be suitable or effective for trading the EUR/GBP currency pair, and adjustments or alternative strategies may be necessary.

4. A MDD of 0.46 for EUR/USD and 0 for EUR/GBP indicates that the strategy experienced significant losses relative to its previous highs. The strategy may be prone to large drawdowns and could benefit from improved risk management techniques to limit losses during adverse market conditions.

Overall Efficiency and Improvement Opportunities

While the initial positive PnL and Sharpe ratio for EUR/USD may suggest some effectiveness of the intrinsic time trading strategy, the subsequent decline in PnL and significant drawdowns raise concerns about its long-term viability. Furthermore, the negative Sharpe Ratio for EUR/GBP indicates that the strategy may not be suitable for trading that currency pair. Therefore, significant improvements and adjustments are necessary to enhance the strategy's efficiency and effectiveness.

Certainly, here's a discussion on the limitations of the model and the nature of the metrics involved:

Limitations of the Model

The results given by our strategy may showcase several limitations.

The intrinsic time trading strategy relies on simplistic assumptions about market behavior and price movements. It may not capture the complexity of real-world financial markets, where various factors such as macroeconomic indicators, geopolitical events, and market sentiment can influence prices. Also, the model may lack adaptability to changing market conditions, because financial markets are dynamic and evolve over time, requiring strategies that can adjust to different regimes and environments. In this case, the intrinsic time strategy's fixed rules may fail to adapt to such changes effectively.There's also a risk of overfitting the model to historical data, which may not generalize well to future market conditions. The strategy's performance may also be sensitive to the quality and granularity of the data used, leading to potential biases or inaccuracies in the assessment of results.

About the FX market and international economics

The FX market has its own characteristics that we have to take into account when analyzing the results of a trading strategy. Each currency pair has its own unique profile and characteristics, influenced by factors such as liquidity, volatility, trading volume, and underlying economic fundamental :

- Major currency pairs, such as EUR/USD, GBP/USD, and USD/JPY, typically exhibit higher liquidity and lower spreads compared to exotic currency pairs.
- Cross currency pairs, such as EUR/GBP, EUR/JPY, and GBP/JPY, may have different volatility profiles and sensitivity to macroeconomic factors due to the absence of the US dollar.

Indeed, a one-size-fits-all approach to trading strategies may not be optimal, as different currency pairs may respond differently to the same set of trading rules, and strategies that perform well on one currency pair may not necessarily translate to success on another pair due to variations in market conditions and price behavior.

Useful sources and bibliography

- Bridging the Gap: Decoding the Intrinsic Nature of Time in Market Data ; James B. Glattfelder, Anton Golub (https://arxiv.org/abs/2204.02682)
- High-Frequency Trading ; Peter Gomber, Björn Arndt, Marco Lutat, Tim Uhle (https://www.deutsche-boerse.com/resource/blob/69642/6bbb6205e6651101288c2a0bfc668c45/data/high-frequency-trading_en.pdf)
- https://www.investopedia.com/articles/07/sharpe_ratio.asp
- https://www.wallstreetmojo.com/maximum-drawdown/

- https://link.springer.com/chapter/10.1057/9781137396495_1
- https://www.fxstreet.com/