Genetic Algorithm-Optimized SMA Crossover Strategy Backtesting

Introduction

In this project, I aimed to develop and optimize a moving average crossover strategy using a genetic algorithm (GA) framework. The goal was to identify optimal parameters for short and long simple moving averages (SMA) that maximize the Sharpe Ratio. The project was inspired by Jiarui Ni and Chengqi Zhang's paper, “An Efficient Implementation of the Backtesting of Trading Strategies,” which provides guidelines on efficient and systematic backtesting approaches for trading strategies.

Methodology

1. **Strategy Development**: The primary trading strategy I developed is a **simple moving average crossover strategy**. The strategy generates a buy signal when the short SMA crosses above the long SMA, and a sell signal when the opposite crossover occurs. This setup capitalizes on potential trends in the stock price by following momentum signals.

2. **Genetic Algorithm (GA) Optimization**: To optimize the strategy, I used a genetic algorithm approach to evolve the SMA parameters. The GA framework started with a population of randomly generated parameter sets (short and long SMA windows). The algorithm evaluated each set by backtesting the strategy and calculating its Sharpe Ratio, which reflects the risk-adjusted return. In each generation, the top-performing strategies were selected, and new parameter sets were generated by applying mutations to these top performers, iterating for five generations.

Key GA settings:

- **Population Size**: 10
- **Generations**: 5
- **Top N Selection**: 3
- **Mutation Range**: ±2 on each parameter (ensuring short SMA window is within [5, 15] and long SMA within [15, 50])

3. **Backtesting**: For each generation, I ran the backtesting using the backtesting library, which efficiently calculates performance metrics such as Sharpe Ratio, Sortino Ratio, maximum drawdown, and win rate. These metrics guided the selection and optimization process.

Results

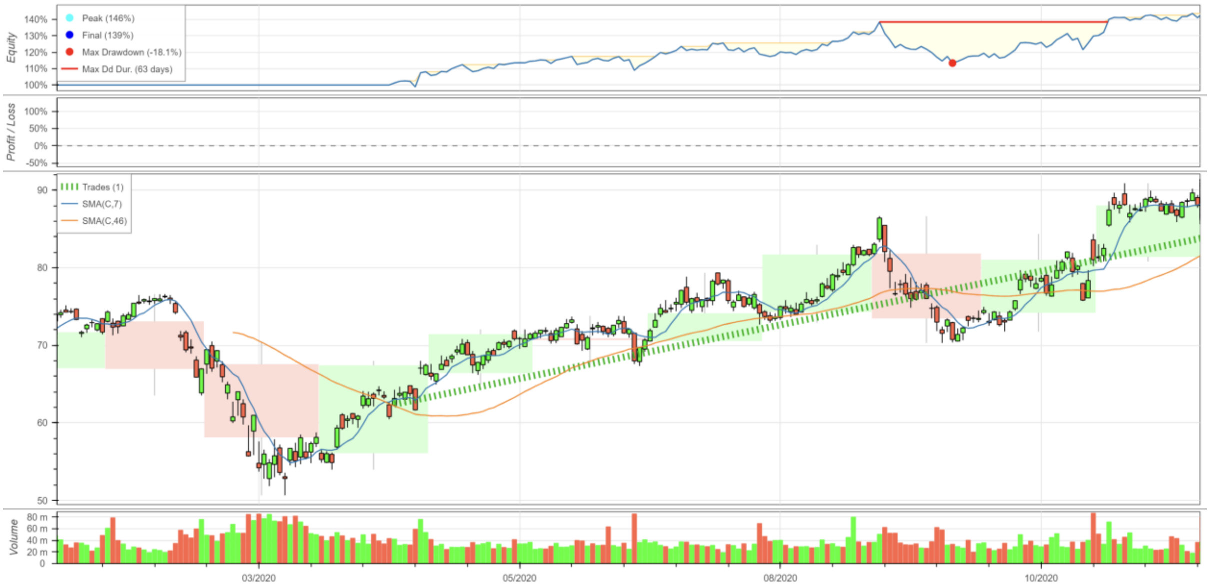
GA Optimization Process and Top Strategies Per Generation: Throughout the five generations, the genetic algorithm effectively identified and refined optimal parameter sets. For example, the top-performing parameters converged around **short SMA windows between 7-12** and **long SMA windows between 41-46**, demonstrating the GA’s capability to focus on optimal ranges. The Sharpe Ratio remained relatively stable in each generation, converging at values around 1.12 for the best-performing strategies.

Final Optimized Strategy Results: The final optimized parameters identified by the GA were:

- **Short Window**: 7
- **Long Window**: 46

Using these optimized parameters, I conducted a comprehensive backtest. The following metrics were observed:

- **Return**: 39.0%
- **Annualized Return**: 38.8%
- **Sharpe Ratio**: 1.12
- **Maximum Drawdown**: -18.1%
- **Win Rate**: 100% (1 trade, indicating a long-term trend hold)



The final plot, as shown, provides a clear visual representation of the strategy's performance over the 2020 timeframe. The green and red highlights denote the periods during which the strategy held a position (green for profitable trades and red for unprofitable ones). The equity curve in the upper portion reflects the steady appreciation in value, demonstrating the effectiveness of the chosen parameters.

Analysis

The genetic algorithm successfully optimized the SMA crossover strategy parameters, achieving a high Sharpe Ratio and significant returns. This result is consistent with Jiarui Ni and Chengqi Zhang's approach, which emphasizes:

- **Efficiency in Backtesting**: The paper advocates for systematic testing to ensure minimal computational overhead and quick convergence on optimal parameters. My application of GA aligns with this by reducing the need to manually tune parameters.
- **Focused Exploration**: By mutating only the top-performing strategies, the GA focused on promising parameter ranges without exploring inefficient areas. This methodology is parallel to the paper's emphasis on efficient backtesting by iteratively narrowing down on high-performance configurations.

The final optimized strategy's performance aligns well with the paper's recommendations on maximizing computational efficiency and exploring strategies with a high potential for profitability.

Conclusion

In conclusion, this project successfully implemented a GA-optimized SMA crossover strategy, achieving robust returns with minimal drawdown. The application of the genetic algorithm significantly reduced the time required to identify optimal SMA windows, ensuring a streamlined backtesting process consistent with Ni and Zhang's recommendations for efficiency. This project highlights the potential of genetic algorithms in tuning trading strategies, as well as the importance of rigorous backtesting in developing robust trading systems.