# A Deep Dive into Portfolio Optimisation

## Under the Hood

SCR Quantitative Research Team

Surrey Capital Research, University of Surrey

January 2025

# The Big Question

If you had **£100,000** to invest across stocks, bonds, gold, and commodities...

**How would you split it?**

- Split equally (1/18 per asset)?
- Use complex mathematical optimization?
- Something else?

**We're testing this empirically.**

# The Research Question

## Core Question

Can sophisticated portfolio optimisation models beat the naive 1/N strategy?

**Why this matters:**

- **For investors:** Should you pay for complex portfolio management?
- **For students:** Does classroom theory work in practice?
- **For quant enthusiasts:** Rigorous empirical comparison

*This is a live debate in quantitative finance — theory vs. practice.*

# Our Investment Universe

| Asset Class | Examples | Count |
|---|---|---|
| UK Equities | HSBC, BP, Shell, Tesco, AstraZeneca, etc... | 15 |
| UK Government Bonds | iShares Core UK Gilts ETF | 1 |
| Precious Metals | Invesco Physical Gold ETC | 1 |
| Commodities | WisdomTree Commodities ETF | 1 |
| **Total** | **Multi-Asset Portfolio** | **18** |

**Data period:** 2015–2025 (10 years)

- Includes COVID crash, 2022 selloff, Brexit volatility
- Daily prices, monthly rebalancing

# The Four Strategies We're Testing

1. **Equal Weight (1/N)** — Naive baseline
2. **Mean-Variance Optimisation (MVO)** — Markowitz (1952)
3. **Black-Litterman** — Bayesian equilibrium + views
4. **Risk Parity** — Equal risk contribution

**All tested on identical data with identical methodology**

# Strategy 1: Equal Weight (1/N)

### The Benchmark

Simplest possible: split money equally across all 18 assets

**How it works:**

- £5,556 to each asset (on £100k capital)
- Monthly rebalancing to maintain 1/18 weights
- No forecasting, no optimization

### Why it's tough to beat

DeMiguel et al. (2009): Equal weight often beats sophisticated models due to estimation error in expected returns and covariances.

# Strategy 2: Mean-Variance Optimisation

## Markowitz (1952) — Foundation of Modern Portfolio Theory

Maximize return for given risk (or minimize risk for given return)

**Optimization:**

$$\max_{w} \frac{w^T \mu - r_f}{\sqrt{w^T \Sigma w}} \quad \text{s.t.} \quad \sum w_i = 1,\ w_i \geq 0$$

**The challenge:**

- Very sensitive to estimation errors in $\mu$ and $\Sigma$
- Small input changes $\rightarrow$ big portfolio changes
- Can produce extreme, unstable weights

# Strategy 3: Black-Litterman

## Combining Market Equilibrium with Investor Views

Overcomes MVO instability by anchoring to market equilibrium

**How it works:**

1. Start with equilibrium returns (from market cap weights)
2. Express views: "UK banks will outperform by 2%"
3. Blend equilibrium + views using Bayesian updating
4. Optimize portfolio with posterior returns

**Key advantage:** More stable than pure MVO — doesn't require perfect forecasts

# Strategy 4: Risk Parity

## Equal Risk Contribution, Not Equal Capital

Philosophy: diversify risk, not just money

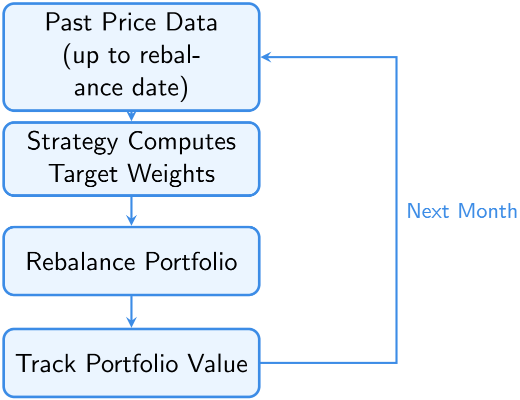
Each asset contributes equally to total portfolio risk:

$$w_i \times \frac{\partial \sigma_p}{\partial w_i} = \frac{\sigma_p}{N}$$

**In practice:**

- Bonds (low vol) get *more* capital
- Equities (high vol) get *less* capital
- Popular with institutional investors (e.g., Bridgewater All Weather)

# How We're Testing: Backtesting Engine




**Critical: No Look-Ahead Bias**

Strategies only see **past data**. This prevents unrealistically good backtest results.

# Code: Preventing Look-Ahead Bias

```
for current_date in dates:
    if current_date in rebalance_dates:
        # CRITICAL: Only past data
        past_prices = self.prices.loc[:current_date]

        # Strategy uses ONLY past data
        target_weights = self.strategy.get_target_weights(
            decision_date=current_date,
            past_prices=past_prices  # No future!
        )

        # Execute rebalancing
        # ...

    # Track portfolio value
    equity_curve[current_date] = portfolio_value
```

# What We're Measuring

| Metric | What It Measures |
|---|---|
| Total Return | Cumulative gain/loss over 10 years |
| CAGR | Annualised return |
| Volatility | Annualised standard deviation |
| Sharpe Ratio | Excess return per unit of risk |
| Max Drawdown | Worst peak-to-trough loss |
| Sortino Ratio | Return per unit of downside risk |
| Turnover | Trading frequency (costs) |

**Goal:** Understand risk-adjusted performance, not just returns

# What Insights We're Looking For

1. **Risk-Adjusted Performance**
   - Which model has the best Sharpe ratio?
   - Is complexity worth the cost?
2. **Crisis Performance**
   - COVID crash (March 2020)
   - 2022 equity-bond selloff
   - Brexit volatility (2016)
3. **Practical Considerations**
   - Turnover costs
   - Implementation difficulty
   - Robustness to parameter choices

# The Strategy Architecture

## BaseStrategy Interface

All models implement this clean abstraction:

```python
class BaseStrategy(ABC):
    @abstractmethod
    def get_target_weights(
        self,
        decision_date: pd.Timestamp,
        past_prices: pd.DataFrame,  # ONLY past
        current_positions: pd.Series,
        cash: float,
    ) -> pd.Series:
        """Return target weights (sum = 1.0)"""
        raise NotImplementedError
```

**Why?** Fair comparison, extensibility, enforces no look-ahead bias

# Example: Equal Weight Strategy

```python
class EqualWeightStrategy(BaseStrategy):
    def __init__(self, tickers: list[str]):
        self.tickers = tickers

    def get_target_weights(self, decision_date,
                           past_prices,
                           current_positions, cash):
        # Just return 1/N for all assets
        n = len(self.tickers)
        weights = pd.Series(1.0 / n,
                            index=self.tickers)
        return weights
```

**Simple, but surprisingly hard to beat!**

# Where We Are Now

**Completed:**

- Data pipeline (18 assets, 2015-2025)
- Backtesting engine
- Equal weight baseline
- Black-Litterman framework

→ **In Progress:**

- MVO implementation
- Black-Litterman views
- Risk Parity solver
- Performance analysis

## Coming Soon

- Full backtest results across all 4 strategies
- Regime analysis (bull/bear/crisis)
- Comprehensive research report

# What This Means

## Bridging the gap between theory and practice

- **For investors:** Does sophisticated management add value?
- **For students:** Testing what you learn in lectures
- **For quants:** Rigorous comparison with clean methodology

*Academic finance says "here's the optimal solution"*
*We're asking: "does it actually work?"*

# Want to Follow Along?

## GitHub Repository

`github.com/AP-Capital-Research/deep-dive-into-portfolio-optimisation`

## Full Research Report

Expected publication: [Month] 2025

**Questions or feedback?**

Contact us at [email/contact]

# Thank You

We're excited to share our findings
with the quant finance community

*Stay tuned for the full results*