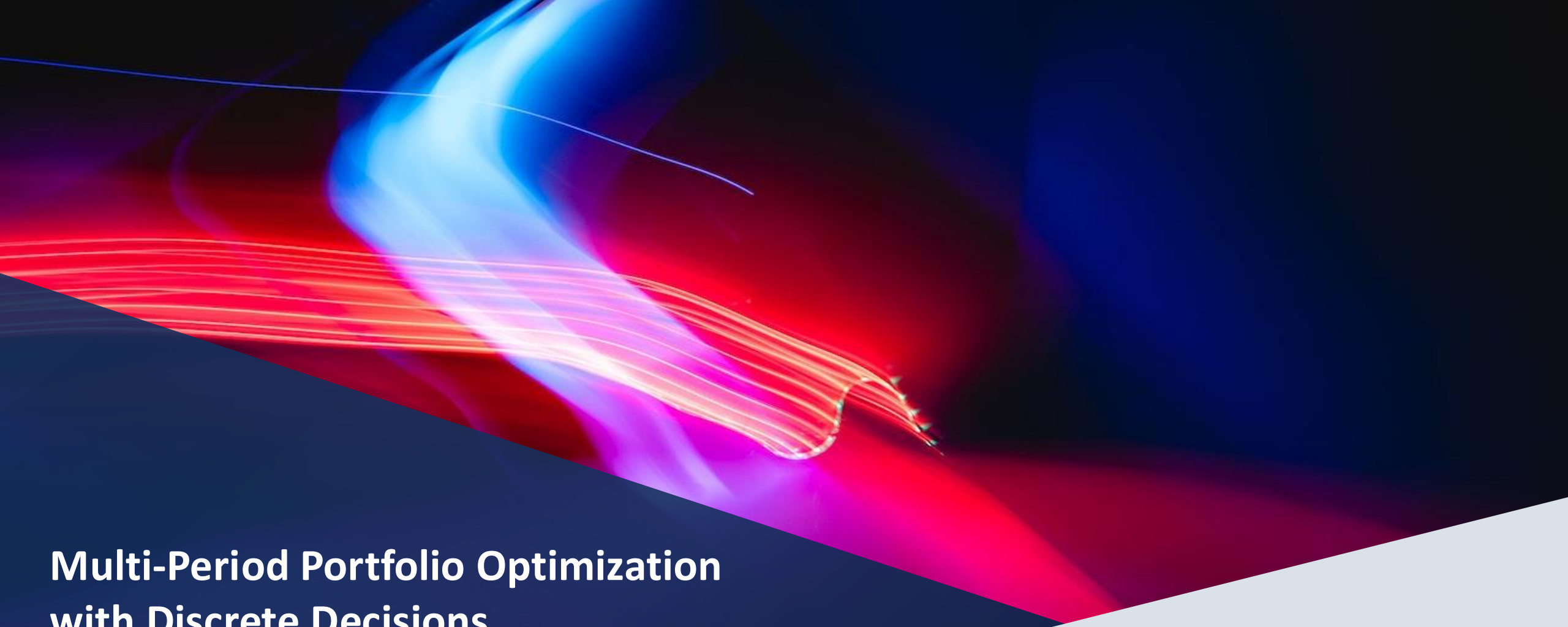


Multi-Period Portfolio Optimization with Discrete Decisions

Dr. Silke Horn
Senior Optimization Engineer
horn@gurobi.com

GUROBI OPTIMIZATION

Better decisions start here

The Gurobi Difference

With Gurobi, you can identify the optimal way to achieve your objectives, in seconds.

The World's Most-Trusted Brands Run on Gurobi

Serving

80%
Of the World's Leading Enterprises

1,200+
Global Customers

40+
Industries

"Every day, we depend on the Gurobi Optimizer to deliver optimal systematic fixed-income portfolios for our private and institutional investors."

Mathieu van Roon, Portfolio Manager, Robeco

VIEW CASE STUDIES

You’re in good company

ASSET MANAGEMENT

- Portfolio Optimization
  - Collateral Allocation
  - Portfolio Replication
  - Bond Management
  - Hedging Strategies
  - Debt Management
- Credit Swap Management
- Trade Settlement
- Asset-Liability Management
- Payment Netting
- Systemic Risk Management

WINCOR NIXDORF

BANK OF AMERICA

ROBECO

BNY MELLON

citigroup

arute solutions

swissQuant

Mean Variance Portfolio Optimization

… with discrete constraints

… over multiple periods

GUROBI OPTIMIZATION

Mean-Variance PO

- Return is a random variable
- Using estimates of first and second moment:
  - Compute optimal compromise between estimated return and risk
  - Enforce full investment
- Classical continuous optimisation problem
- What if we have **discrete constraints** on the investment?

GUROBI OPTIMIZATION

What's a discrete decision?

- Trading constraints:
  - Minimum trade/holding size
  - Limited turnover
  - Lot selection for tax optimization
- Transaction costs:
  - Market slippage
  - Fixed-charge transaction costs
- Portfolio constraints:
  - Cardinality
  - Diversification by risk classes
  - Diversification by market sectors
- ...

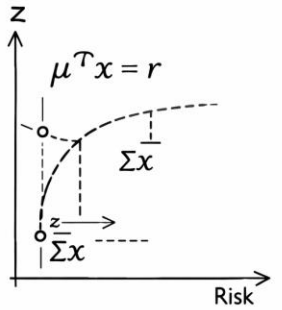
The classical Markowitz model

$$\begin{aligned} \max \quad & \mu^T x - \frac{\gamma}{2} x^T \Sigma x \\ \text{s.t.} \quad & \sum x_i = 1 \\ & x \geq 0 \end{aligned}$$

```
with gp.Model as m:
    x = m.addMVar(n)
    m.addConstr(x.sum() == 1)



    m.setObjective(mu @ x - gamma/2.0 * x @ sigma @ x)
    m.optimize()
```

- $x \in \mathrm{R}^n$ is the optimization variable, indicating the relative proportion of investment into each asset
- $\mu \in \mathrm{R}^n$ is an estimator of the return
- $\Sigma \in \mathrm{R}^{n,n}$ is an estimator for the covariance
- $0 \leq \gamma \in \mathrm{R}$ is the risk-aversion parameter
- Nonnegativity of x ensures long-only

- $\mu \in \mathrm{R}^n$: NumPy 1D ndarray
- $\Sigma \in \mathrm{R}^{n,n}$: NumPy 2D ndarray

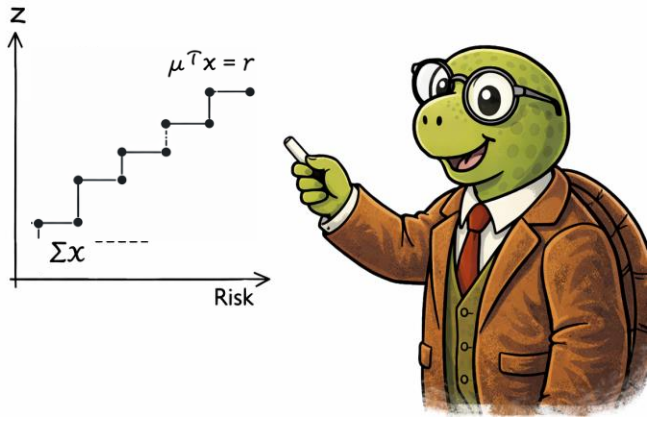


Adding a cardinality constraint

$$\begin{aligned} \max \quad & \mu^T x - \frac{\gamma}{2} x^T \Sigma x \\ \text{s.t.} \quad & \sum x_i = 1 \\ & x \leq b \\ & \sum b_i \leq k \\ & b \in \{0,1\} \\ & x \geq 0 \end{aligned}$$

```python
with gp.Model as m:
    x = m.addMVar(n)
    m.addConstr(x.sum() == 1)
    b = m.addMVar(n, vtype=gp.GRB.BINARY)
    m.addConstr(x <= b)
    m.addConstr(x.sum() <= k)
    m.setObjective(mu @ x - gamma/2.0 * x @ sigma @ x)
    m.optimize()
```

- $b \in \mathrm{R}^n$ is a **binary** optimization variable, deciding whether we're "allowed" to trade
- If $b_i = 0$ then x_i is forced to zero
- At most k entries of b can take value 1



Meet Gurobi Finance

GUROBI OPTIMIZATION

Gurobi Finance

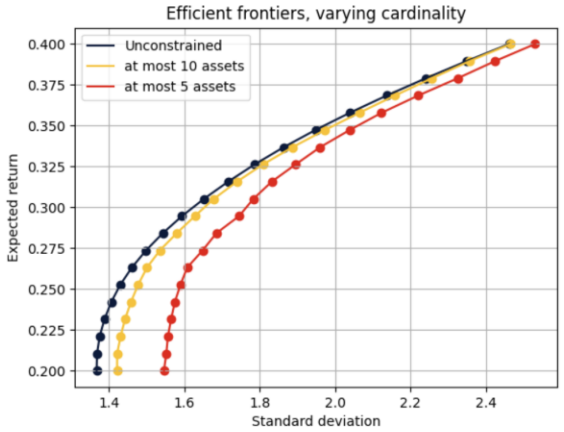
Search

CONTENTS:

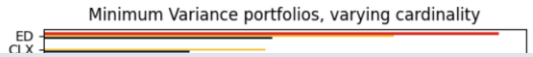
Portfolio Optimization with Gurobi

Portfolio optimization is a tool for optimizing the expected return, risk, or other measure for a portfolio of investments. Quantitative analysts and portfolio managers use portfolio optimization to support their investment making decisions. With Gurobi's mixed-integer programming (MIP) technology, it is possible to incorporate discrete decisions in the portfolio selection. Common examples are cardinality constraints on the number of allocations, or the consideration of transaction costs.

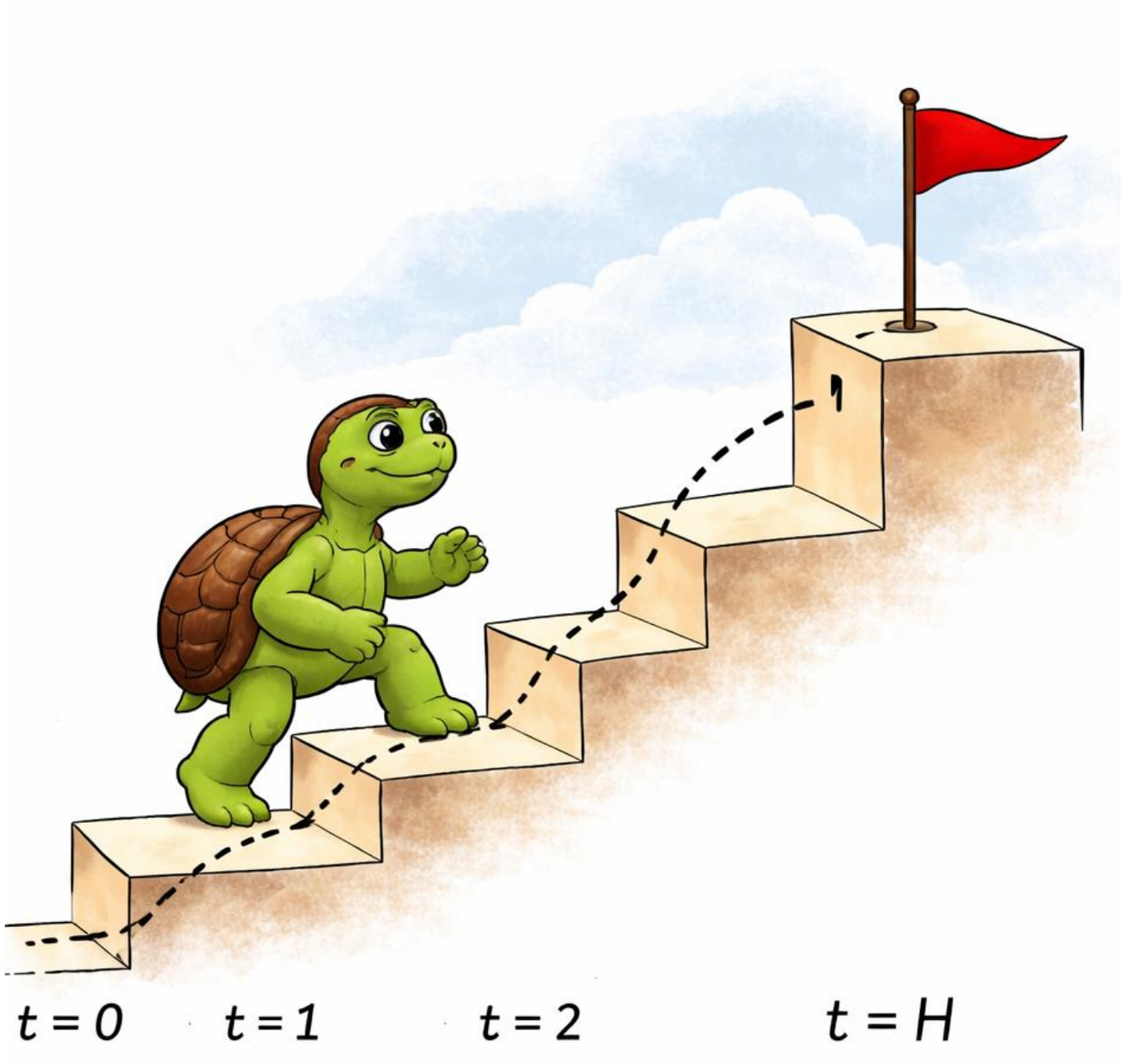
As a simple demonstration, we compute the efficient frontier for a set of 463 S&P 500 stocks (1) without constraints on the number of transactions, (2) with the restriction of trading at most 10 assets, and (3) trading only five assets.



Using only five assets, that is, effectively forbidding diversification, the estimated risk increases substantially, whereas restricting the investment to comprise at most ten assets, we see that the efficient frontier moves only slightly away from the unconstrained frontier. Hence, there is limited benefit of using more than ten assets from an efficiency point of view. Finally, we compare the minimum variance portfolios for the three scenarios above:



https://gurobi-finance.readthedocs.io



GUROBI OPTIMIZATION

Portfolio decisions are sequential

- The target keeps moving (markets and signals evolve)
- Decisions today → tomorrow's constraints
- Trading is gradual (costs, turnover)
- Optimize across time, not step-by-step

From single-period to multi-period

$$\max \quad \mu^T x \quad - \frac{\gamma}{2} x^T \Sigma x$$

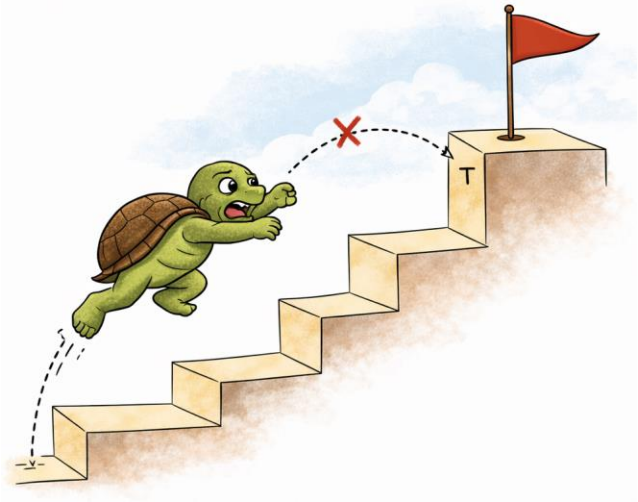
$$\text{s.t.} \quad \sum_i x_i \quad = 1$$

$$x \geq 0$$

```
with gp.Model as m:
    x = m.addMVar(n)


    m.addConstr(x.sum() == 1)



    m.setObjective(mu @ x - gamma/2.0 * x @ sigma @ x)
    m.optimize()
```

- $x \in \mathrm{R}^n$: optimization variable indicating the proportion of investment into asset i

- $\mu \in \mathrm{R}^n$: estimator of the return
- $\Sigma \in \mathrm{R}^{n,n}$: estimator of the covariance



From single-period to multi-period

$$\max \quad \sum_t \mu_t^T x_{.t} - \frac{\gamma}{2} x_{.t}^T \Sigma x_{.t}$$

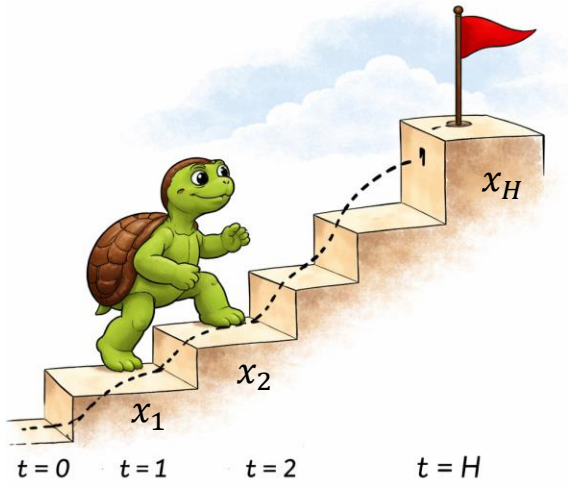
$$\text{s.t.} \quad \sum_i x_{i,t} = 1$$

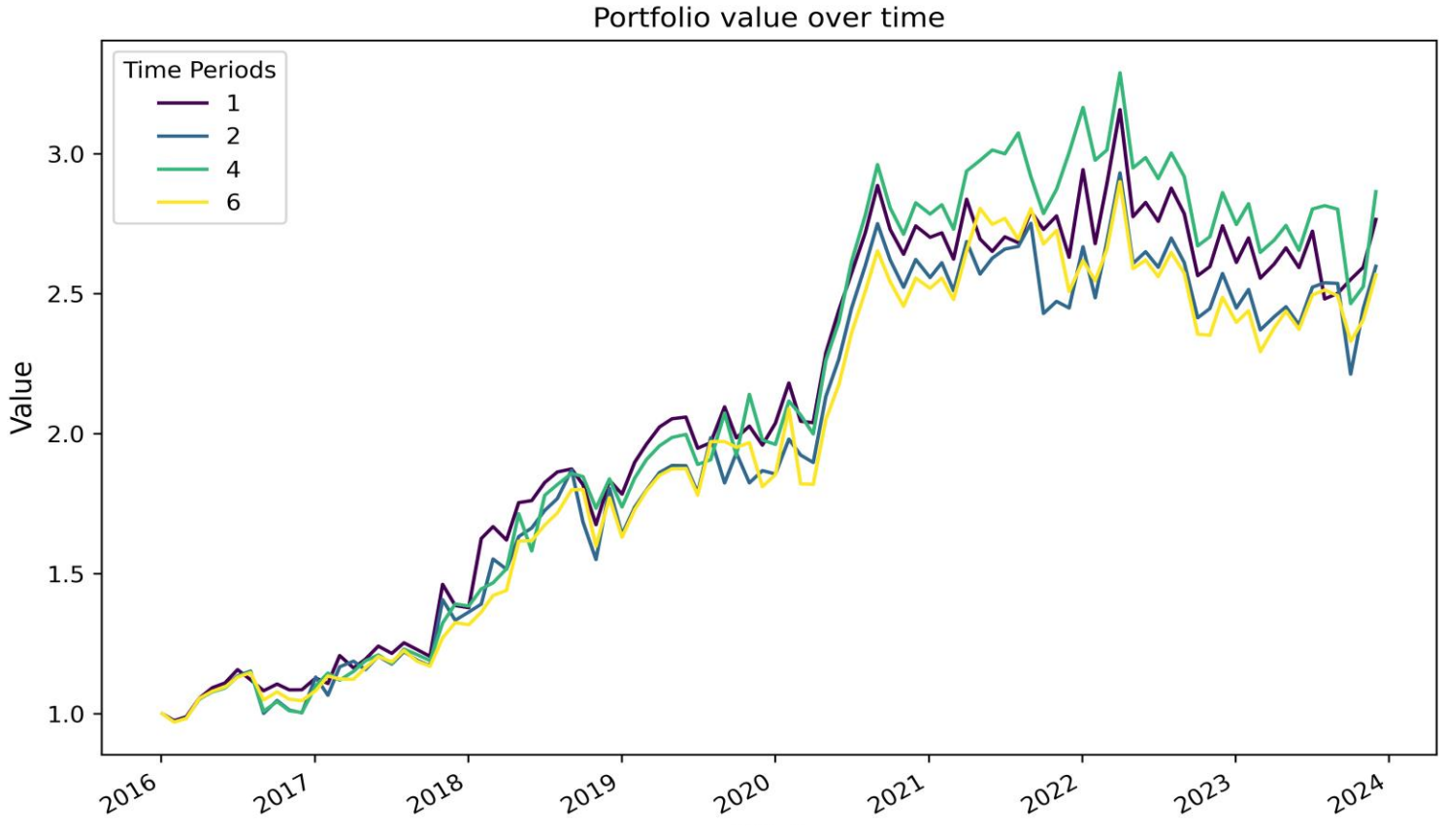
$$x_{i,t} = x_{i,t-1} + x_{i,t}^{\text{buy}} - x_{i,t}^{\text{sell}}$$

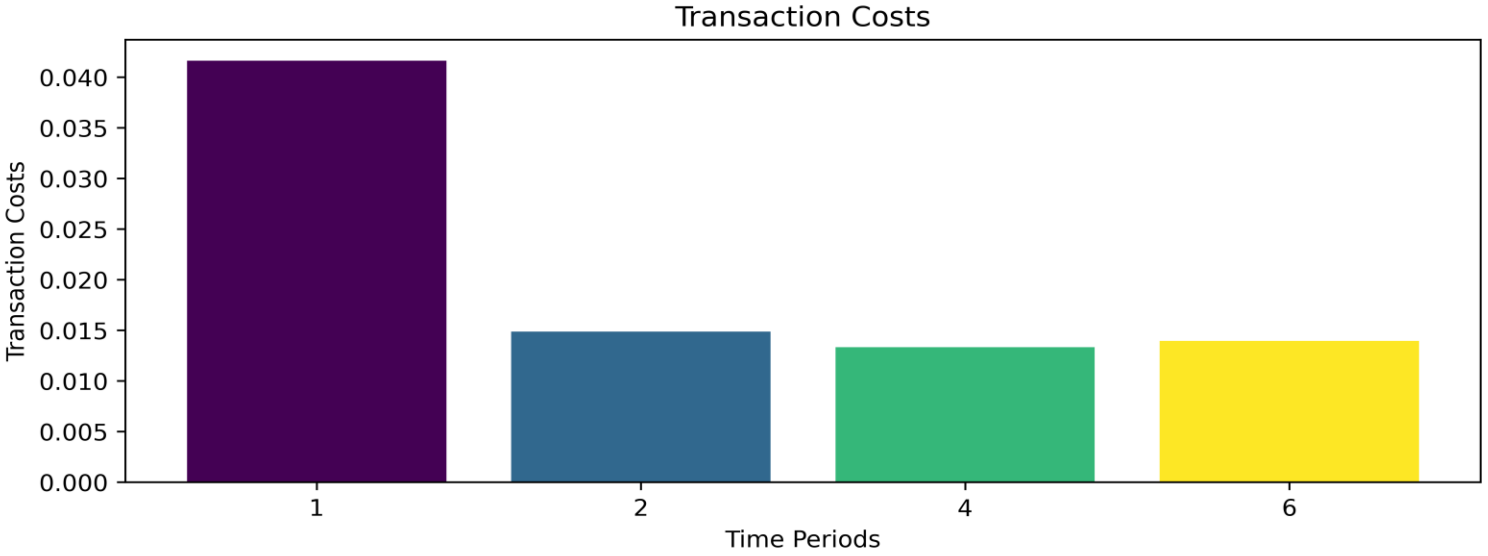
$$x \geq 0$$

```
with gp.Model as m:
    x = m.addMVar((n, H+1))
    x_buy = m.addMVar((n, H+1))
    x_sell = m.addMVar((n, H+1))
    m.addConstr(x.sum(axis=0) == 1)
    for t in range(H+1):
        m.addConstr(x[:,t] == x[:,t-1] + x_buy[:,t] - x_sell[:,t])
    m.addConstr(x[:, 0] == x0)
    m.setObjective((mu @ x - gamma/2.0 * x.T @ sigma @ x).sum())
    m.optimize()
```

- $x \in \mathrm{R}^{n,H+1}$: optimization variable indicating the proportion of investment into asset i in time period t
- $x^{\text{buy}}, x^{\text{sell}} \in \mathrm{R}^{n,H+1}$: optimization variables indicating the proportion of buys/sells in asset i in time period t
- $\mu_t \in \mathrm{R}^n$: estimator of the return in time period t
- $\Sigma \in \mathrm{R}^{n,n}$: estimator of the covariance



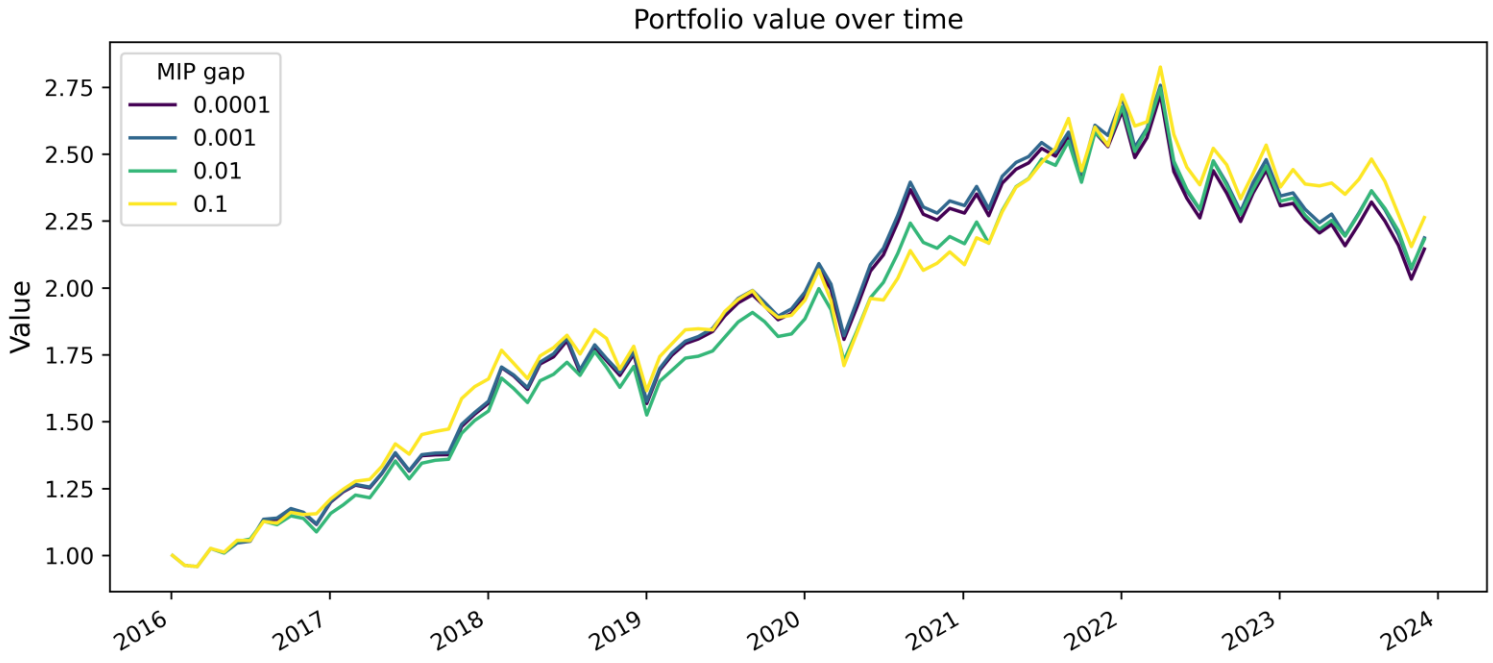


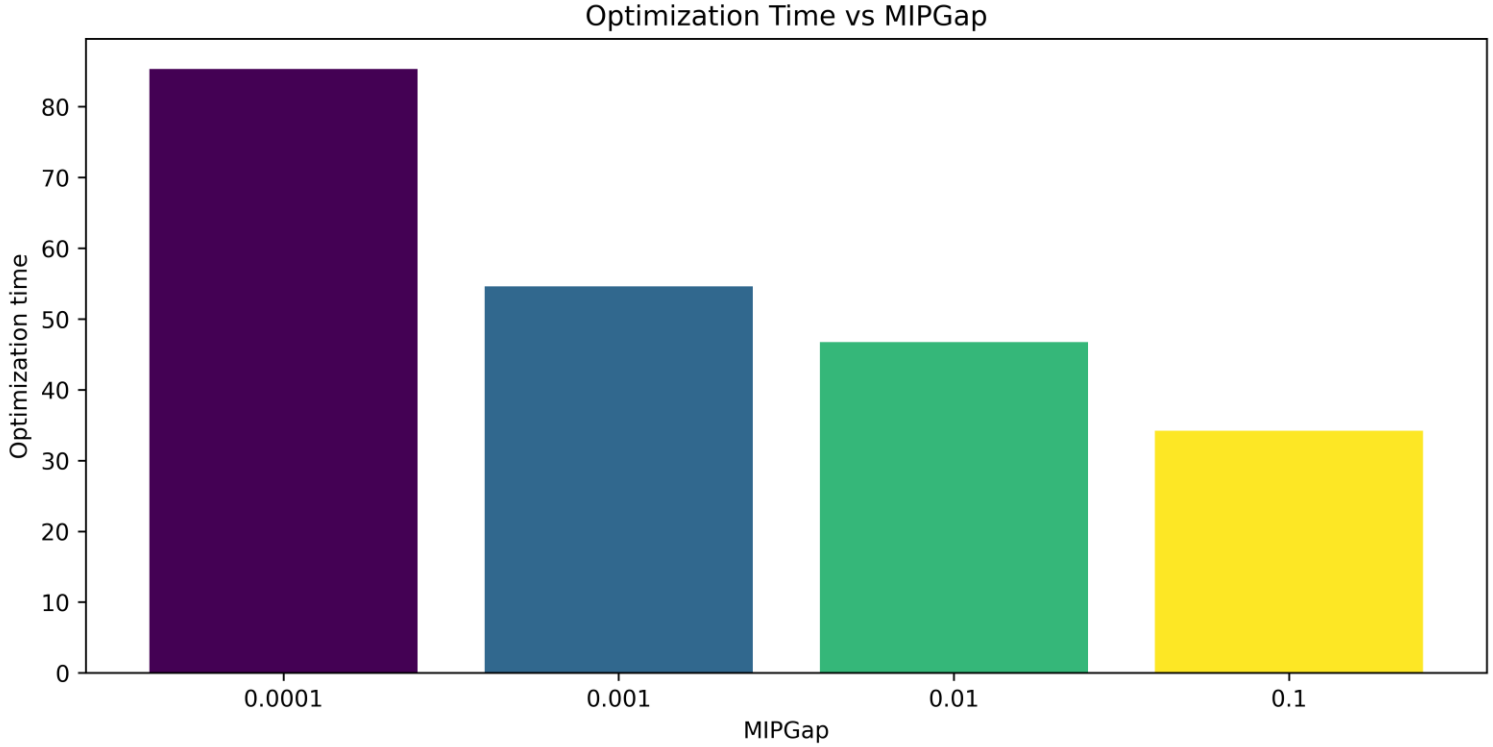


GUROBI OPTIMIZATION

Multi-period portfolio backtesting

- Our example:
  - Stocks from the S&P 500
  - Monthly rebalancing
  - Limited turnover
  - Transaction costs (slippage, …)
  - Minimum position/trade size
  - Time span 10 years
  - 10 factors (via PCA)
- Rolling horizon: In each step, execute first period, then recompute

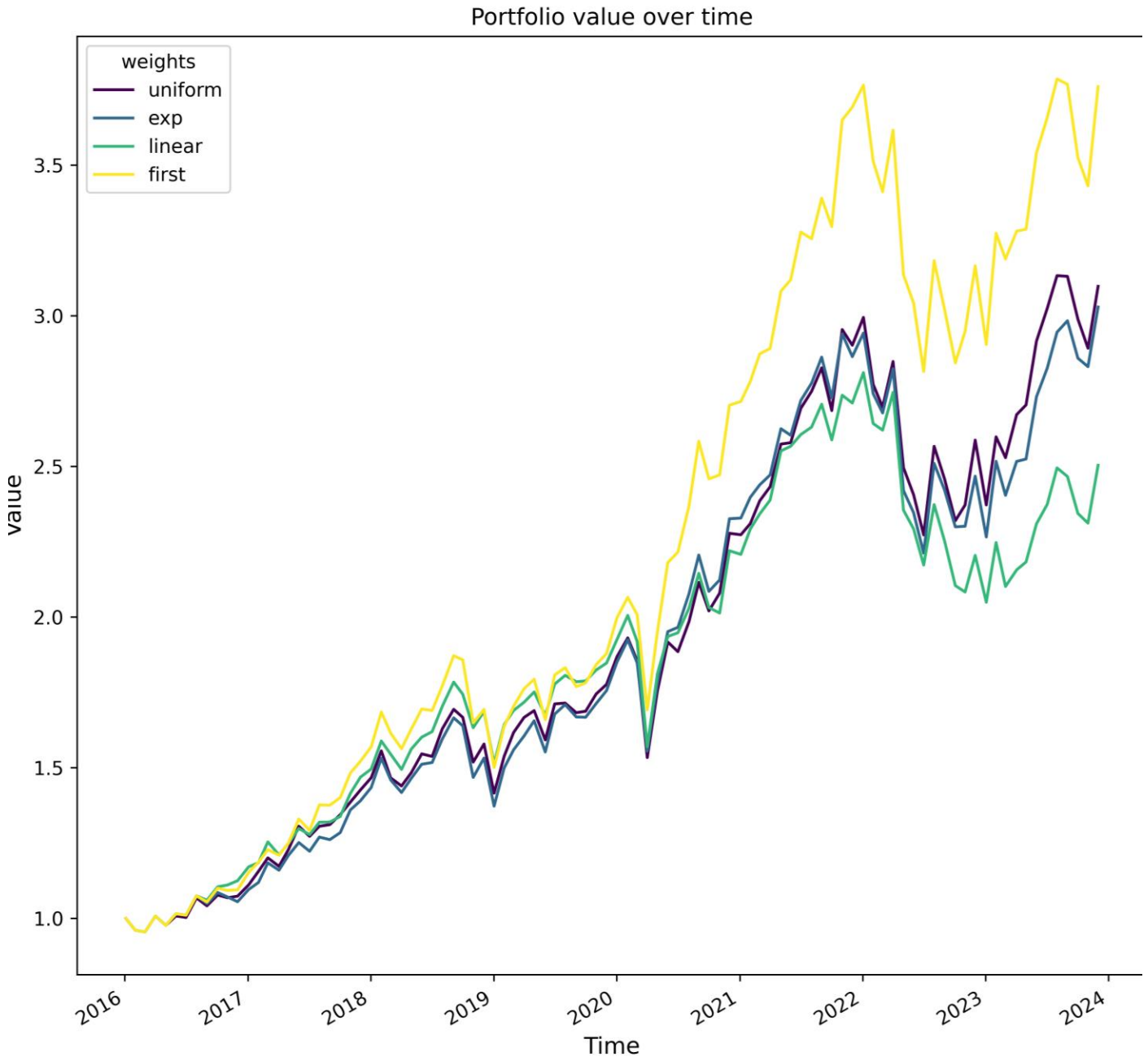




GUROBI OPTIMIZATION

Scaling multi-period optimization

- Backtesting requires solving many models
- Our example:
  - 2 time periods
  - Rolling horizon
- The default MIP gap may be overly strict
- Relaxing to 1%: about 50% speedup in our test with very similar portfolio development
- Tradeoff: slight inaccuracy, more throughput
- Parameter tuning can improve performance



GUROBI OPTIMIZATION

Multi-period modeling in practice

- Objective weighting
  - Discount future periods
  - *When:* signals decay / uncertainty increases
- Rolling horizon
  - Solve over multiple periods
  - Execute first step only
  - *When:* backtesting / live trading
- Target portfolio transition
  - Plan trades over multiple periods
  - Execute gradually to control costs
  - *When*: large reallocations / illiquid markets
- Solver tuning
  - Adjust MIP gap
  - Warm-start from previous solution (shifting forward)

GUROBI OPTIMIZATION

Takeaways

- Multi-period ≠ single-period repeated
- Backtesting = many solves → performance matters
- Modeling determines tractability
- Gurobi enables solving these models at scale
- Convenient and intuitive Python API
  (but you can use C, C#, C++, Java, Matlab, R, too!)

GUROBI OPTIMIZATION

Thank You

For more information: gurobi.com

See us at the Gurobi booth!

Dr. Silke Horn
Senior Optimization Engineer
horn@gurobi.com

