Yield Optimization Mathematical Framework

YieldSolver

February 13, 2025

1 YieldSolver as a part of YieldQuant

YieldQuant is an AI assistant that helps you optimize your yield. It uses a mathematical framework called YieldSolver to optimize your yield.

1.1 Problem Overview

YieldSolver is an optimization framework that aims to allocate capital among various liquidity and staking pools to maximize yield while considering risk constraints. The allocation problem is formulated as a constrained optimization problem using convex programming.

1.2 Notation

- N : Number of tokens.
- M : Number of liquidity and staking pools.
- $\mathbf{c} \in \mathbb{R}^N$: Vector of available capital for each token.
- $\mathbf{W} \in \mathbb{R}^{M \times N}$: Allocation matrix where W_{ij} represents the capital allocated to token j in pool i.
- $\mathbf{A} \in \mathbb{R}^{M \times N}$: APY matrix, where A_{ij} represents the adjusted annual percentage yield for token j in pool i.
- $\mathbf{R} \in \mathbb{R}^M$: Risk score vector for all pools.
- $\gamma \in [0, 1]$: User-defined risk tolerance parameter.
- $\mathbf{w} \in \mathbb{R}^M$: Risk-adjusted weight vector.

1.3 Capital Constraints

The total capital allocated across pools must equal the available capital for each token:

$$\sum_{i=1}^{M} W_{ij} = c_j, \quad \forall j \in 1, \ldots, N \tag{1}$$

Additionally, allocations must be non-negative:

$$W_{ij} \geq 0, \quad \forall i, j. \tag{2}$$

1.4 Liquidity Pool Constraints

Each liquidity pool consists of two tokens with equal allocation:

$$W_{ik} = W_{il}, \quad \forall i \text{ where tokens } k \text{ and } l \text{ are paired.} \tag{3}$$

For all other tokens not involved in a particular pool:

$$W_{ij} = 0, \quad \forall j \notin k, l. \tag{4}$$

1.5 Risk Score Calculation

The risk score R_i for each pool i is calculated based on trading volume (V_i), total value locked (TVL, T_i), and volatility (σ_i). The inputs are winsorized at a limit λ, then log-transformed and normalized:

$$V_i' = \log(1 + V_i), \quad T_i' = \log(1 + T_i)\ R_i \quad = w_1(1 - \frac{V_i'}{Q_{95}(\mathbf{V'})}) + w_2(1 - \frac{T_i'}{Q_{95}(\mathbf{T'})}) + w_3\frac{\sigma_i}{Q_{95}(\sigma)} \tag{5}$$

where $Q_{95}(\cdot)$ denotes the 95th percentile normalization, and weights w_1, w_2, w_3 sum to 1.

1.6 Historical APY Adjustment

To account for fluctuations in historical APY, we apply a smoothing mechanism:

$$A_{ij}^{\text{adj}} = \alpha A_{ij}^{t} + (1 - \alpha)A_{ij}^{t-1}, \tag{6}$$

where A_{ij}^{adj} is the adjusted APY for token j in pool i, A_{ij}^{t} is the most recent observed APY, A_{ij}^{t-1} is the previous period APY, and $\alpha \in [0, 1]$ is a smoothing parameter that determines the weight given to recent observations.

Additionally, extreme values are winsorized to prevent excessive influence from outliers:

$$A_{ij}^{\text{final}} = \min(\max(A_{ij}^{\text{adj}}, Q_5(A)), Q_{95}(A)), \tag{7}$$

where $Q_5(A)$ and $Q_{95}(A)$ are the 5th and 95th percentiles of the historical APY distribution.

1.7 Risk Weighting Function

The risk-adjusted allocation weight is determined using a softmax transformation of the inverse hyperbolic tangent function:

$$w_i = \frac{\exp(\operatorname{atanh}((1 - \gamma)(1 - 2R_i))/T)}{\sum_{j=1}^{M} \exp(\operatorname{atanh}((1 - \gamma)(1 - 2R_j))/T)} \tag{8}$$

where T is a temperature parameter.

1.8 Objective Function

We maximize the risk-adjusted expected yield:

$$\max_{\mathbf{W}} \sum_{i=1}^{M} w_i \sum_{j=1}^{N} \frac{T_{ij}}{A_{ij} + T_{ij}} W_{ij} \tag{9}$$

subject to the capital constraints, liquidity pool constraints, and non-negativity constraints.