The DeFi Ecosystem Game: Proof-Via-Simulations Mathematical Algorithms

May 20, 2024

1 Introduction

The following Algorithms are constructed as a mathematical equivalent to the Portfolio Manager (PM), as coded in gon_pm.py. Please also note that they mimic the 'State Machine Diagrams' of the original paper, 'The DeFi Ecosystem Game: Proof-Via-Simulations'. Once this paper is officially published, it will be added to the codebase. This report is structured as follows: Chapter 2 provide a description for all notations used in the Algorithms presented in Chapter 3.

Finally, please note that the Algorithms of Chapter 3 sometimes refer to the appendix in the paper, which is currently not published and for now denoted as ??.

2 Mathematical Notations

This chapter provides a brief overview to the mathematical notations used in this report, as per below:

- d, R_S, N
  Denote a day in the simulation, the return of the Strategy used in the simulation, and the total number of simulation days, respectively.
- TVL_S, PnL_S, TVL_{AMM}
  Refer to the Total Value Locked (in USDT, not including GON rebates) on Strategy S, the Profit and Loss of given Strategy S, and the Total Value Locked on the UniswapV2 Auto Market Maker Contract, respectively.
- pm or $PM(i), \sum_{pm=1}^{\#PM} M_p$
  The ith Portfolio Manager interacting with the GONLabs game engine, and the sum of the monetary (USDT, not including GON rebates) value of each PM, deployed in Strategy S, respectively.
- λ_{fees}
  The constant performance fee $PM(i)$ for participating in PnL_S, where fees are charged only if $PnL_S(d) > 0$.
- $P_{\frac{GON}{USDT}}$
  The price of GON in USDT. When denoted in an algorithm, if would be updated as a function of an Action by $PM(i)$, so denoted without the 'i' notation for simplicity.
- $S(PM(i))$
  The total holdings of Portfolio Manager i in Strategy S. Note that these are the total proceedings in both USDT and GON terms and will be allocated to the approriate $PM(i)$ USDT or GON accounts.
- $\beta_{<param>}$
  Denote any local parameters used for the Portfolio Manager, e.g. β_{break} denotes a trade break the Portfolio Manager might take during its investment cycle. The description of any $\beta < param >$ is directly described as a comment in the respective Algorithm.
- $+=\leftarrow$
  This notation used to denote that the variable to the right is added to the variable to the left of that notation.
- $Algorithm(PM)$
  Used to denote in Algorithm 1 only and refer to the fact that this where each PM seperate Algorithm would be evaluated. i.e. if there are 5 SIMPLE PMs and 3 LP Provider, their respective Algorithms state machine cycles will be evaluated where $UserActions \leftarrow Algorithm(PM)$ is denoted.
- x, y, k
  The variables that holds the UniswapV2 constant AMM product rule: $k \leftarrow x \times y$.
- $< variable > [a, b]$ or $< variable > [a, a+1, ...b]$
  Denote that the $< variable >$ on the left side will pick a random number out of the range a to b, inclusive. note that both notations shown here are equivalent and used intangibly.

3 Algorithms

The following set of algorithms are presented in friendly form, as per below.

Algorithm 1 The GONLabs game engine

for $(d, R_S(d))$ **in** N **do**
  $TVL_S(d-1) \leftarrow \sum_{pm=1}^{\#PM} M_p$
  $TVL_S(d) \leftarrow TVL_S(d-1) \times (1 + R_S(d))$ // Update Strategy TVL
  $PnL_S(d) \leftarrow TVL_S(d) - TVL_S(d-1)$
  if $PnL_S(d) > 0$ **then**
    $\Delta USDT \leftarrow PnL_S(d) \times \lambda_{fees}$ // Compute Perf. fee
    $TVL_S(d) \leftarrow TVL_S(d) - \Delta USDT$ // Take fees from Strat. TVL
    $P_{\frac{USDT}{GON}}, TVL_{AMM} \leftarrow alg : UniswapV2TokenSwapUSDTforGON(\Delta USDT)$ // Buy & Burn
  end if
  $S(PM(i)) \mathrel{+}= \Delta GON_{rewards}(d) \leftarrow \sum_{pm=1}^{\#PM} \lambda(d)_{pm_i}$ // see Appendix ??
  $UserActions \leftarrow Algorithm(PM)$
end for

Algorithm 2 PM: Buy & Hold

On Start:
  $USDT_{pm(i)} \in [500, 501..., 1000]$ // Deposit USDT to PM[i] account
  $S(PM(i)) \leftarrow USDT_{pm(i)}$ // Deploy PM[i] USDT to Strategy
for d in N **do**
  No action
end for

Algorithm 3 PM: Liquidity Provider (LP)

Settings:
  $USDT_{LP} \leftarrow y + USDT_{support}$ // Deposit USDT to LP account
  $\beta_{support} \leftarrow 10\%$ // Pct. of $USDT$ of holdings to stabilise price with
  $\beta_{stblTarget} \leftarrow 5\%$ // Stability target
  $\beta_{break} \leftarrow 3$ // Num. of Days since last intervention
On Start:
  $x \leftarrow \frac{1}{2} GON_{LP}$ // 50% will go to AMM
  $GON_{LP} \leftarrow \frac{1}{2} x$ // 50% left with LP
  $TVL_{AMM} \leftarrow k = x \times y$ // LP Initialise Uniswap V2 AMM pool ratio
  $LastTradedDay \leftarrow 0$ // Counter for checking β_{break}
for d in N **do**
  if $\left(\frac{P_{\frac{GON}{USDT}}(d)}{P_{\frac{GON}{USDT}}(d-3)} - 1 \right) \times 100 \leq \beta_{stblTarget}$ **AND** $LastTradedDay > \beta_{break}$ **then**
    $\Delta USDT \leftarrow \beta_{support} \times USDT_{support}$
    $P_{\frac{GON}{USDT}}, TVL_{AMM} \leftarrow alg : UniswapV2TokenSwapUSDTforGON(\Delta USDT_{support})$
    $LastTradedDay \leftarrow 0$
  else
    $LastTradedDay \leftarrow LastTradedDay + 1$
  end if
end for

Algorithm 4 PM: SIMPLE

Settings:

$\beta_{yield} \in [20\%, 70\%]$ // Yield target

$\beta_{break} \in [15d, 20d]$ // Num. of Days since traded

β_{PMs} // Num. of SIMPLE PMs, see Appendix **??**

On Start:

$USDT_{PM(i)} \in [200, 201, ..., 400]$ // Deposit USDT to PM[i] account

$GON_{PM(i)} \leftarrow 0$ // No GON tokens to start

$S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

$LastTradedDay \leftarrow 0$ // Counter for checking β_{break}

for d in N **do**

  for i in β_{PMs} **do** // for each PM[i] of Investor type SIMPLE

    $PnL_{PM(i)}(d) \leftarrow S(Pnl_{PM(i)}, d)$ // compute **Unrealised** PnL on Strategy

    if $PnL_{PM(i)}(d) \geq \beta_{yield}$ **then**

      $USDT_{PM(i)}, GON_{PM(i)} \leftarrow S(PM(i))$ // Take profits

      $USDT_{PM(i)} \leftarrow alg : UniswapV2TokenSwapGONforUSDT(\Delta GON_{PM(i)})$ // Sell GON rewards

      $LastTradedDay \leftarrow 0$

    else if $LastTradedDay \geq \beta_{break}$ **then**

      $USDT_{PM(i)} \in [200, 201, ..., 400]$ // Deposit USDT to PM[i] account

      $S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

      $LastTradedDay \leftarrow 0$

    else

      $LastTradedDay \leftarrow LastTradedDay + 1$

    end if

  end for

end for

Algorithm 5 PM: SIMPLE++

Settings:

$\beta_{yield} \in [20\%, 70\%]$ // Yield target

$\beta_{break} \in [15d, 20d]$ // Num. of Days since traded

β_{PMs} // Num. of SIMPLE++ PMs, see Appendix **??**

$\beta_{APY} \leftarrow 60d$ // Strategy APY protection

On Start:

$USDT_{PM(i)} \in [200, 201, ..., 400]$ // Deposit USDT to PM[i] account

$GON_{PM(i)} \leftarrow 0$ // No GON tokens to start

$S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

$LastTradedDay \leftarrow 0$ // Counter for checking β_{break}

for d in N **do**

  for i in β_{PMs} **do** // for each PM[i] of Investor type SIMPLE++

    $PnL_{PM(i)}(d) \leftarrow S(Pnl_{PM(i)}, d)$ // compute **Unrealised** PnL on Strategy

    $APY(\beta_{APY}) \leftarrow \left(\frac{R(S(d))}{R(S(d-\beta_{APY}))} - 1\right)$ // compute APY on last β_{APY} days

    if $PnL_{PM(i)}(d) \geq \beta_{yield}$ **OR** $APY(\beta_{APY}) \times 100 < 0$ **then**

      $USDT_{PM(i)}, GON_{PM(i)} \leftarrow S(PM(i))$ // Take profits

      $USDT_{PM(i)} \leftarrow alg : UniswapV2TokenSwapGONforUSDT(\Delta GON_{PM(i)})$ // Sell GON rewards

      $LastTradedDay \leftarrow 0$

    else if $LastTradedDay \geq \beta_{break}$ **AND** $APY(\beta_{APY}) \geq 0$ **then**

      $USDT_{PM(i)} \in [200, 201, ..., 400]$ // Deposit USDT to PM[i] account

      $S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

      $LastTradedDay \leftarrow 0$

    else

      $LastTradedDay \leftarrow LastTradedDay + 1$

    end if

  end for

end for

Algorithm 6 PM: Conservative

Settings:

β_{PMs} // Num. of Conservative PMs, see Appendix **??**

β_{gas} // Gas fee for trading, see Table **??**

$\beta_{APY} \leftarrow 7d$ // Strategy APY protection

On Start:

$USDT_{PM(i)} \in 2000$ // Deposit USDT to PM[i] account

$GON_{PM(i)} \leftarrow 0$ // No GON tokens to start

$S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

for d in N **do**

  for i in β_{PMs} **do** // for each PM[i] of Investor type Conservative

    $PnL_{PM(i)}(d) \leftarrow S(Pnl_{PM(i)}, d)$ // compute **Unrealised** PnL on Strategy

    $APY(\beta_{APY}) \leftarrow \left(\frac{R(S(d))}{R(S(d-\beta_{APY}))} - 1\right)$ // compute APY on last β_{APY} days

    if $PnL_{PM(i)}(d) \geq 5 \times \beta_{gas}$ **OR** $APY(\beta_{APY}) \times 100 < 0$ **then**

      $USDT_{PM(i)}, GON_{PM(i)} \leftarrow S(PM(i))$ // Take profits

      $USDT_{PM(i)} \leftarrow alg : UniswapV2TokenSwapGONforUSDT(\Delta GON_{PM(i)})$ // Sell GON rewards

    else if $APY(\beta_{APY}) \geq 0$ **then**

      $USDT_{PM(i)} \in [200, 201, ..., 400]$ // Deposit USDT to PM[i] account

      $S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

    end if

  end for

end for

Algorithm 7 Uniswap V2 Token Swap GON for USDT

Input: ΔGON // GON to swap for USDT

Output: New: $P_{\frac{GON}{USDT}}$, TVL_{AMM}, Return: $\Delta USDT$

$k \leftarrow x \times y$ // Compute the constant product

$\Delta USDT \leftarrow y - \frac{k}{x+\Delta GON}$ // Calculate USDT received

$x_{\text{new}} \leftarrow x + \Delta GON$ // Update GON reserve

$y_{\text{new}} \leftarrow y - \Delta USDT$ // Update USDT reserve

$P_{\frac{GON}{USDT}} \leftarrow \frac{y_{\text{new}}}{x_{\text{new}}}$ // Calculate new price

$TVL_{AMM} \leftarrow x_{\text{new}} \times P_{\frac{GON}{USDT}} + y_{\text{new}}$ // Calculate new TVL

Algorithm 8 Uniswap V2 Token Swap USDT for GON

Input: $\Delta USDT$ // USDT to swap for GON

Output: New: $P_{\frac{USDT}{GON}}$, TVL_{AMM}, Return: ΔGON

$k \leftarrow x \times y$ // Compute the constant product

$\Delta GON \leftarrow x - \frac{k}{y+\Delta USDT}$ // Calculate GON received

$y_{\text{new}} \leftarrow y + \Delta USDT$ // Update USDT reserve

$x_{\text{new}} \leftarrow x - \Delta GON$ // Update GON reserve

$P_{\frac{GON}{USDT}} \leftarrow \frac{x_{\text{new}}}{y_{\text{new}}}$ // Calculate new price

$TVL_{AMM} \leftarrow y_{\text{new}} \times P_{\frac{GON}{USDT}} + x_{\text{new}}$ // Calculate new TVL

Algorithm 9 PM: Sophisticated

Settings:

$\beta_{yield} \in [30\%, 31\%...50\%]$ // Yield target

$\beta_{R_{GONRtn}} \leftarrow 30d$ // $\frac{GON}{USDT}$ return monitoring over last 30 days

β_{PMs} // Num. of Sophisticated PMs, see Appendix **??**

$\beta_{APY} \leftarrow 60d$ // Strategy APY protection

On Start:

$USDT_{PM(i)} \in 2000$ // Deposit USDT to PM[i] account

$GON_{PM(i)} \leftarrow 0$ // No GON tokens to start

$S(PM(i)) \leftarrow USDT_{PM(i)}$ // Invest PM[i] USDT in Strategy

$LastTradedDay \leftarrow 0$ // Counter for checking β_{break}

for d in N **do**

  for i in β_{PMs} **do** // for each PM[i] of Investor type Sophisticated

    $PnL_{PM(i)}(d) \leftarrow S(Pnl_{PM(i)}, d)$ // compute **Unrealised** PnL on Strategy

    $APY(\beta_{APY}) \leftarrow \left(\frac{R(S(d))}{R(S(d-\beta_{APY}))} - 1\right)$ // compute Strat. APY

    $R_{\frac{GON}{USDT}}(\beta_{GONRtn}) \leftarrow \left(\frac{P_{\frac{GON}{USDT}}(d)}{P_{\frac{GON}{USDT}}(d-\beta_{GONRtn})} - 1\right)$

    $\beta_{break} \in [5, 6..10]$ // Randomised break resets on every loop

    if $LastTradedDay \geq \beta_{break}$ **AND** $APY(\beta_{APY}) \geq \beta_{yield}$ **then**

Rationale: Strategy making money, strong belief in the project, invest more.

      $USDT_{strat} \leftarrow\in [500, 501, ...1000]$ // Source more USDT for PM[i]

      $S(PM(i)) \mathrel{+}=\leftarrow USDT_{strat}$ // Add PM[i] USDT in Strategy

      $USDT_{AMM} \leftarrow\in [500, 501, ...1000]$ // Source more USDT for PM[i]

      $USDT_{AMM}, GON_{AMM} \leftarrow ComputeLiqSplit(USDT_{AMM}, k)$ // Split USDT & GON per k AMM ratio

      $TVL_{AMM}(PM(i)) \mathrel{+}=\leftarrow USDT_{AMM}, GON_{AMM}$ // Add AMM liquidity

      $LastTradedDay \leftarrow 0$

    else

      $USDT_{PM(i)}, GON_{PM(i)} \mathrel{+}=\leftarrow S(PM(i))$ // Unvest

      $LastTradedDay \leftarrow 0$

    end if

    if $LastTradedDay \geq \beta_{break}$ **AND** $APY(\beta_{APY}) < 0$ **then**

      $USDT_{AMM} \leftarrow\in [500, 501, ...1000]$ // Source more USDT for PM[i]

      $USDT_{AMM}, GON_{AMM} \leftarrow ComputeLiqSplit(USDT_{AMM}, k)$ // Split USDT & GON per k AMM ratio

      $TVL_{AMM}(PM(i)) \mathrel{+}=\leftarrow USDT_{AMM}, GON_{AMM}$ // Add AMM liquidity

      $LastTradedDay \leftarrow 0$

    else

      $USDT_{PM(i)}, GON_{PM(i)} \mathrel{+}=\leftarrow S(PM(i))$ // Unvest from Strategy

      $USDT_{PM(i)}, GON_{PM(i)} \mathrel{+}=\leftarrow TVL_{AMM}(PM(i))$ // Remove AMM Liquidity

      $USDT_{PM(i)} \mathrel{+}=\leftarrow alg: UniswapV2TokenSwapGONforUSDT(\Delta GON_{PM(i)} \times 0.5)$ // Sell 50% of GON rewards

      $LastTradedDay \leftarrow 0$

    end if

    $LastTradedDay \leftarrow LastTradedDay + 1$

  end for

end for