Proposal Discussion Topic

This proposal asks the following question:

Should the NonFungiblePositionManager.sol contract be optimized to reduce gas consumption?

Description

The Uniswap core

smart contract for each pool handles flash loans, swaps, price oracles, and the deployment of liquidity (v3-core/UniswapV3Pool.sol at main · Uniswap/v3-core · GitHub).

While it is possible to directly interact with the core smart contracts, most users interact with the set of peripheral

smart contracts developed by Uniswap Labs (GitHub - Uniswap/v3-periphery: Peripheral smart contracts for interacting) Swapping and liquidity providing is carried through the SwapRouter.sol and NonfungiblePositionManager.sol smart contracts, respectively.

The SwapRouter has been updated recently to version (1.1.0), while the NonfungiblePositionManager (NFPM) hasn’t changed since it was deployed last May (it is still deployed as "Uniswap V3: Positions NFT” at address 0xC36442b4a4522E871399CD717aBDD847Ab11FE88).

One problem with the NFPM smart contract, however, is that it is consuming a significant amount of gas at every transaction.

If we look at this Dune analytics dashboard made by Shippooor DAO (@shippooor

on twitter): Dune Analytics, we see that, in total, more than to 9909 ETH has been used to mint all positions created through the Positions NFT smart contract:

[

image

1920×517 28.2 KB

](https://global.discourse-cdn.com/business6/uploads/uniswap1/original/2X/a/a2ea5742b6567e5681954e3d7ab43af0a66ce47d.jpeg)

Since the total supply is currently slightly above 200000, this means each LP position costs close to 0.05 ETH in gas to mint, on average. Similarly, the average burn cost is ~0.03 ETH, meaning that adding+withdrawing liquidity to Uniswap v3 through the NonFungiblePositionManager can cost up to 0.08 ETH.

If we look at the size distribution for all Uniswap positions, we see that 10% of all positions have less than $250 in total locked value and 20% have between $250 and $2500 in value locked. So these 30% of all positions will have to pay at least between 10-100% of the deposited value in gas fees.

[

image

1920×813 69.9 KB

](https://global.discourse-cdn.com/business6/uploads/uniswap1/original/2X/5/5f3e212cb99f6d5390d75e4266ba2c88eea2b288.jpeg)

Interestingly, about 53% of all positions generated profits with a median return equal to 1267$ without taking the gas fees into account. Taking fees into account, only 39% of all positions are profitable (64336/170161).

Proposal

The purpose of this Temperature Check is to gauge the interest of the Uniswap community in reducing the gas consumption of the NonFungiblePositionManager smart contract. Governance participants are more than welcome to chip in and discuss the contents of this proposal here.

A follow-up proposal may request funds to pay for a solidity dev (which could be from the Uniswap Labs team) to research alternative methods for managing Uniswap v3 liquidity positions with gas reduction as a first priority. Funds may also be needed to pay for the audit, deployment cost, etc.

Ultimately, Uniswap governance will consider the proposed solutions and vote on a resolution to deploy a “lite” version of the NonFungiblePositionManager contract that will be the one facing users on the app.uniswap.org interface.

Possible solutions include:

1. Recycling of existing NFT positions.

Adding liquidity to an existing NFT position costs much less gas than creating a new NFT. For instance, looking at a few LP transactions in the ETH-UNI pool:

- Mint new token: 431,925 (Ethereum Transaction Hash (Txhash) Details | Etherscan)
- increaseLiquidity: 235,142 (Ethereum Transaction Hash (Txhash) Details | Etherscan)
- decreaseLiquidity: 190,326 (Ethereum Transaction Hash (Txhash) Details | Etherscan)

The https://app.yewbow.org interface already facilitates the recycling of NFT position by displaying existing tokens owned by the current user whenever deploying a new position:

[

image

1920×1402 95.2 KB

](https://global.discourse-cdn.com/business6/uploads/uniswap1/original/2X/d/d547232c93e4dcc2116a406b77a5efd6bba21989.jpeg)

Piggybacking on this, one way to reduce gas costs with minimal change to the NFPM code would be to return the ownership of burned NFT positions to the new NFPM contract by default.

These positions could then be “recycled” by other users, where they would gain the ownership of the ERC721 token instead of minting a new token and simply add liquidity to it though increaseLiquidity()

1. Using ERC1155 instead of ERC721 to track positions

: Uniswap v3 positions are what I would call semi-fungible: two different users that deploy the same amounts of liquidity to the same pool using the same upper/lower ticks are effectively creating fungible positions.

For instance, all “Full Range” positions are fungible, and I could find 73 of them in the ETH-USDC-0.3% pool. Similarly, I was able to find 41 positions with liquidity between (1675, 2945) in the ETH-USDC-0.3% pool, all of the would be “fungible” (although a different feeGrowthInside0LastX128, feeGrowthInside1LastX128 would be tied to each owner).

ERC1155 could replace the current ERC721 implementation. One way would be to encode the pool Id (80 bits), upperTick (24 bits), and lowerTick (24 bits) inside the 256bit id of an ERC1155 so that the balance of the token issued would correspond to the amount of liquidity deployed.

1. Store all the information in the smart contract, no need to emit any token

All the relevant information about LPs position is already stored in the core contract inside the Position Struct. So issuing a token, whether it is a ERC721 or ERC1155 is not necessary per se if positions can be tracked off-chain (eg. using the Uniswap subgraph).

Many protocols directly interact with the core smart contracts to manage liquidity. Examples of such protocols include: Charm Finance, Gelato, Visor/Gamma protocol, DefiEdge.io, Unipilot, etc. Each one shows significant gas consumption improvements over the NFPM contract.

For example, calling the rebalance() function on an Alpha Vaults from Charm Finance directly interacts with the v3-core smart contract and only consumes 532,577 gas to call the burn() function twice

and mint() function twice

Ethereum Transaction Hash (Txhash) Details | Etherscan)

Looking through the Etherscan trace, the “mint” part of Charm Finance rebalance is only 113203 gas – although a more realistic goal for a revised NFPM may be 250,000 or lower, similar to the cost to addLiquidity().

[

image

1916×1520 127 KB

](https://global.discourse-cdn.com/business6/uploads/uniswap1/original/2X/7/780c70e7d284d69bbde1f879fb878cab77d2eb9f.jpeg)

Conclusions

I am not a solidity programmer, so my role in this proposal is to bring attention to this issue, which ultimately affects “retail” users (like me!) at a much higher level than whales/institutions.

Changing the NonFungiblePositionManager, however, would break the current subgraph and would require a rewriting of theGitHub - Uniswap/interface: An op interface for the Uniswap protocol to handle the new tokenId standard. So it is unclear whether the efforts would be worth it.

Lowering gas fees is also personal for me. I completed the Uni v3 LPing portion of my taxes last week. I was very active this year, deployed 200+ positions and testing out different strategies.

For those that follow me on twitter/medium, you know that I have written several articles about LPing on Uni v3 and you may be glad to know that I practice what I preach: deploying small 1-2 ETH LP positions in medium-to-high volatility assets, keep the ranges narrow (+/-10%) and leaving the position before “ImPeRmaNEnt LoSs”.

I’ve suffered some big losses, won big on others, but fortunately I was net positive in my USD-based PnL. I’d say I became more sophisticated as time went on, so my PnL might be a bit higher for the last quarter.

However, this net positive PnL does not take into account the gas fees. Deploying a LP position costs up to $250 per transaction, so I ended up being net negative in both ETH terms and USD terms. Obviously, this is 100% my fault for being so active, but part of me thinks some of these costs could be avoided if the NFT smart contract did not consume as much gas unnecessarily.

Snapshot vote: Will post the link soon? I cannot create a new proposal onsnapshot.org it seems