In this post, we describe another type of PoW (dual PoW) to produce a block, which reveals similar properties to classical PoW (namely, primal PoW) - a probability of producing a block is proportional to the miner's hash power, but the resulting statistics of block time and hash value are somewhat dual

. A similar property can be found for linear programming (LP) and so we name the algorithms as primal/dual PoW.

Primal PoW

: A list of miners (0, …, n - 1

) concurrently solves a hash-based puzzle so that a miner has the right to produce next time if the hash value of the block satisfies:

h_j \leq d

where j

is the index of the miner, h_j

is the hash value of the block mined, and d

is the difficulty.

Assuming there is no network latency, the miner who finds the block hash earliest will win, i.e.,

i = \arg \min_j (t_j)

,

where t_j

is the time that a miner solves the puzzle, and i

is the index of the miner that is chosen as the block producer in this round.

Dual PoW

: A list of miners (0, …, n - 1

) concurrently solves a hash-based puzzle in time t

. At t

, each miner reveals the block with the smallest hash value h_j

during mining, and the miner with the smallest hash value has the right to produce the block, i.e.,

i = \arg \min_j(h_j)

, and

t_j = t, \forall j \in {0, ..., n - 1}

.

With the definitions of the primal and dual PoW, we first have the following result:

Result 1: Linear Probability

: Assuming the hash powers of the miners are [H_0, H_1, ..., H_{n - 1}]

, the probability of a miner producing a block for both primal/dual

PoW is p_i = \frac{H_i}{\sum_j{H_j}}

.

Result 2: Dual Statistics

: Another interesting result is that the statistics of the block mined may exhibit dual property, which is summarized below:

Algorithm

Block Time

Block Hash

Primal PoW

Exponential(1/expected_block_time)

Uniformly distributed in [0, d]

Dual PoW

t

Exponential *

(*) Approximate from Beta distribution (link)

Application to Blockchain

: Directly applying dual PoW to the blockchain may be vulnerable to self-fish attack - if a miner finds a hash value that is small enough, it may start to mine the next block before t

expires. A further solution to alleviate the issue is under investigation. One direction may be that a block with a specific height is unknown until t

expires by incorporating the smallest hash values of other miners that are broadcasted after t

into the block.