Diversification Across Mining Pools: Optimal Mining Strategies under PoW

Panagiotis Chatzigiannis, Foteini Baldimtsi, Igor Griva and Jiasun Li
George Mason University
Fairfax VA 22030
Email: pchatzig@gmu.edu, foteini@gmu.edu, igriva@gmu.edu, jli29@gmu.edu

***Abstract*—Mining is a central operation of all proof-of-work (PoW) based cryptocurrencies. The vast majority of miners today participate in "mining pools" instead of "solo mining" in order to lower risk and achieve a more steady income. However, this rise of participation in mining pools negatively affects the decentralization levels of most cryptocurrencies.**

In this work, we look into mining pools from the point of view of a miner: We present an analytical model and implement a computational tool that allows miners to optimally distribute their computational power over multiple pools and PoW cryptocurrencies (i.e. build a mining portfolio), taking into account their risk aversion levels. Our tool allows miners to maximize their risk-adjusted earnings by diversifying across multiple mining pools which enhances PoW decentralization. Finally, we run an experiment in Bitcoin historical data and demonstrate that a miner diversifying over multiple pools, as instructed by our model/tool, receives a higher overall Sharpe ratio (i.e. average excess reward over its standard deviation/volatility).

I. Introduction

The majority of cryptocurrencies use some type of proof-of-work (PoW) based consensus mechanism to order and finalize transactions stored in the blockchain. At any given time, a set of users all over the world (called miners or maintainers) compete in solving a PoW puzzle that will allow them to post the next block in the blockchain and at the same time claim the "coinbase" reward and any relevant transaction fees. In the early years of cryptocurrencies solo mining was the norm, and a miner using his own hardware would attempt to solve the PoW puzzle himself, earning the reward. However, as the exchange rate of cryptocurrencies increased, the PoW competition become fiercer, specialized hardware was manufactured just for the purpose of mining particular types of PoW (e.g. Bitcoin or Ethereum mining ASICs [1]), and eventually users formed coalitions for better chances of solving the puzzle.

These coalitions known as *mining pools*, where miners are all continuously trying to mine a block with the "pool manager" being the reward recipient, enabled participating users to reduce their mining risks[1]. After the establishment of mining pools, it become nearly impossible for "solo" miners to compete on the mining game, even if they were using specialized hardware, or else they could risk not to earn any rewards at all during the hardware's lifetime.

The selection of a mining pool is not a trivial task. A large number of pools exist each offering different reward distribution methods and earning fees (as we further discuss in Section II-B). At the same time different pools control a different ratio of the overall hashrate consumed by a cryptocurrency and larger pools (in terms of hashrate) offer lower risk, as they typically offer more frequent payouts to the miners. But how can a miner make an optimal decision about which mining pools to participate in and for which cryptocurrencies at any given time considering the variety of possible options?

Our contributions. We present an analytical tool that allows risk-averse miners to optimally create a mining portfolio that maximizes their risk-adjusted rewards. We characterize miners by their total computational resources (i.e. hash power) and their risk aversion level, and mining pools by their total computational power (i.e. hash rate) and the reward mechanism they offer. We model the hash rate allocation as an optimization problem that aims to maximize the miner's expected utility. In Section III, we provide three different versions of our model. The first one, inspired by [2], concerns a miner that wishes to mine on a single cryptocurrency, while aiming to diversify among any number of mining pools (including the solo mining option). The second version captures miners which diversify across different cryptocurrencies that use the same PoW mining algorithm. In our third version, we model miners who also wish to diversify across cryptocurrencies with different PoW mining algorithms. Our modeling technique is based on standard utility maximization, and extends the Markowitz Modern portfolio theory [3] to multiple mining pools, rather than multiple assets.

In Section IV we present an implementation of our model. We develop a Python tool that uses the constrained optimization by linear approximation (COBYLA) [4] method to automate the pool distribution for an active miner. A miner can use our tool by providing as input its own mining power (for *any* PoW type) as well as his risk aversion rate and *any* number of pools he wishes to take into account when computing the optimal distribution of his mining power. As expected, we observe that for "reasonable" values of risk aversion level, the miner would generally allocate more of his resources to pools offering large hash power combined with small fees, without however neglecting other pools that are not as "lucrative".

Finally, to illustrate the usefulness of our tool, we run an experiment on Bitcoin historical data (Section V). We start by considering a Bitcoin miner who starts mining passively on a single chosen pool for 4 months and compute his earnings on a daily basis.

Then, we consider a miner with the same hash power who

[1] We measure a miner's "risk" by the variance of rewards over time.

using our tool, would "actively" diversify every 3 days over 3 Bitcoin pools and reallocate his hash power accordingly. In our experiment we observe that the "active" miner improves his reward over risk ratio (Sharpe ratio [5]) by 260% compared to the "passive" miner. In our experiments, for the "passive" miner we picked a large and reputable pool (Slush pool) while for the "active" miner we added one more pool of equivalent size and fee structure (ViaBTC), as well as a lower-fee smaller pool (DPool) for a better illustration. Note that there are several degrees of freedom in our experiments (i,e., time periods, set of pools selected etc.). Thus, we include replications with different parameters (time period, pools, risk aversion, miner's power and frequency of diversification) to show how each one of them can affect the end result.

Related Work. There exist a few running tools which on input a miner's hashing power suggest which cryptocurrency is currently most profitable. For instance, tools like Multipoolminer [6], Smartmine [7] or Minergate [8] start by benchmarking the CPU/GPU of the miner, (which we consider an orthogonal service to the third version of our model) and then suggest a cryptocurrency which would offer the best reward at that time. To make that decision, they look into various cryptocurrencies' parameters (i.e. block time, reward etc.) and the current difficulty (their exact model is unclear). Which pools the miner will use towards mining the suggested cryptocurrency is either hard-coded by the tool, or chosen by the miner. Our method differs from such tools in various aspects. Most importantly, in our model we take the risk aversion rate of the miner into account, which is an important factor when making financial decisions. Moreover, we focus on allocating a miner's power over *multiple* pools, as opposed to just different cryptocurrencies, which can benefit the decentralization within a cryptocurrency.

Miner's risk aversion was taken into account by Fisch et al [9] who provided a top-down (from pool's point of view) analysis, i.e. focused on optimal pool operation strategies towards maximizing the pool's expected utility. Cong et al. [2] also took risk aversion into account in their modeling, however the focus of their work was different from ours. In particular, they focus on *the interaction* between miners and pools. They first demonstrate the significant risk-diversification benefit offered by mining pools for individual miners, highlighting risk-sharing as a natural centralizing force. Then, they demonstrate that the risk-sharing benefit within a large pool could be alternatively obtained through miner diversification across multiple small pools. Finally, they present an equilibrium model where multiple pool managers compete in fees to attract customer miners. In our work, we focus on the miners' side: we develop tools to help miners diversify among different pools and cryptocurrencies to maximize their risk-adjusted earnings.

Because we emphasize on hash rate allocations across multiple mining pools, either within the same or over different cryptocurrencies, our analysis distinguishes from contemporary works such as [10], who present an economic model of hash power allocation over different cryptocurrencies sharing the same PoW algorithm in a Markowitz fashion. This perspective also sets us apart from [11] who study mining across different currencies in a strategic fashion, without accounting for pooled mining.

Finally, some recent works examined the case where miners change the mining pool they mine with, in order to optimize their rewards from a *network performance scope* (communications delay). Y.Lewenberg et al. [12] showed how network delays incentivize miners to switch among pools in order to optimize their payoffs due to the non-linearity introduced. X.Liu et al. [13] discussed how to dynamically select a mining pool taking into account the pool's computation power (hash rate) and the network's propagation delay. Network performance is an important aspect when diversifying across pools, and we view [12], [13] as complementary to our work.

II. Mining Background

As of today, the majority of blockchain-based cryptocurrencies use PoW for maintaining their ledger. Miners listen the network for (a) pending transactions and (b) new blocks of transactions to be posted on the ledger. The role of a miner is to select a subset of pending transactions, assemble them to a new block and perform computational work towards finding a random nonce r that will make the block valid and allow the miner to append it on the blockchain and thus win the reward. The brute-forcing process of finding a suitable nonce r such that together with the rest of the block contents b satisfies the property $H(b||r) < T$ for some target value T is called "mining". For Bitcoin, this translates to finding a suitable hash pre-image using the double SHA256 hash function.

Note that, although solving PoW puzzles was initially done using ordinary CPUs, the increasing prices of cryptocurrencies have resulted in a "hardware race" to develop the most efficient mining hardware using Application Specific Integrated Circuits (ASICs), designed to perform SHA256 hashing operations in many orders of magnitude faster than CPUs or GPUs.

A. Mining Pools

Mining is a random process, in most cryptocurrencies the computational power devoted to solve the PoW puzzle is very high, which implies a high variance on the miner's reward. In Bitcoin, even for a miner using state of the art ASICs, there is a good probability that he never gets a block mined during the hardware's lifetime. This led to the formation of mining pools, where coalitions of miners are all continuously trying to mine a block, with the "pool manager" being the reward recipient. If any of the participating miners finds a solution to the PoW puzzle, the pool manager receives the block reward R and distributes to the participants, while possibly keeping a small cut (or fee f). The block reward R distribution is based on how much work these miners performed. A method for the pool manager to measure how much effort each miner has put into the pool is by keeping a record of *shares*, which are "near-solutions" to the PoW puzzle (or "near-valid" blocks), satisfying the property $T_s < H(b||r) < T$ where T_s is the "share difficulty". There are several methods to distribute the reward R to the miners, which we analyze below.

B. Reward Methods in Mining Pools

Different pools offer slightly different reward methods (or a mix of them), with the most popular being: *pay per share* (PPS), *Proportional*, and *pay per last N shares* (PPLNS). In the PPS reward method, the miners are not immediately

TABLE I. NOTATION

Total number of pools	M
Hashrate and fee of pool m	Λ_m,f_m
Transaction fee	tx
Miner's hashing power allocated to pool m (and cryptocurrency c)	$\lambda_m(\lambda_{m,c})$
Miner's total hashing power (for mining algorithm α)	$\lambda_{\mathsf{A}}(\lambda_\alpha)$
Constant Absolute Risk Aversion (CARA)	ρ
Cryptocurrency c total hashrate	Λ_c
Block time and block reward of cryptocurrency c	D_c,R_c
Total number of cryprocurrencies	C
Total number of PoW mining algorithms	A
(Average) network difficulty	T
Cryptocurrency exchange rate	E_c
Number of blocks found on a day d	B_d
Diversification interval (days)	t
Sharpe ratio, Total accumulated reward/payoff	S,P

paid when a block is found, however each block reward is deposited into a "central pool fund" or "bucket". The miners are paid proportionally to the shares submitted throughout their participation in the pool, regardless of if and when the pool has found a solution to the PoW puzzle. PPS is generally considered to offer a steady, almost guaranteed income, independent of the pool's "luck" finding a block. In the proportional reward method, whenever a pool solves the puzzle for a new block, the new block reward is distributed to the pool's miners proportionally to the number of shares each miner has submitted to the pool for that particular block. This method however was found to be vulnerable to the "pool-hopping" attack, where the miners could exploit the pool's expected earnings, variance and maturity time and "hop away" to another pool or solo-mining when the pool's attractiveness is low [14]. The PPLNS method was implemented to counter this attack, where the miners' reward is distributed according to the "recent" number of submitted shares, thus invalidating shares submitted early. As shown in Appendix A, the PPLNS method in mining pools is the most popular reward method today. In addition, many other PPLNS variants are currently being used by mining pools, e.g the RBPPS (Round Based Pay Per Share) method where the pool only pays the reward after the block eventually gets confirmed by the network (thus excluding deprecated blocks). We refer the reader to [15] and [14] for thorough and complete analysis of pool reward methods. In our setting, we mostly consider pools that offer a PPLNS reward method (or its variants). We generally do not consider pools that use the PPS method, as the miners' expected earnings do not depend on the variance of finding blocks.

Mining pools offering multiple reward methods. As mentioned above, some mining pools offer multiple reward systems (i.e. the Coinotron pool [16] offers both PPS and PPLNS). We study these types of pools separately, as some miners might opt for different fee contracts within the same pool.

Let a mining pool m with total hashrate Λ_m, offering both PPLNS and PPS reward systems to choose from, where z is the percentage of pool's hashing power paid using a PPLNS fee contract, and $(1-z)$ is the percentage paid using a PPS fee contract. Also let λ_m be a miner's hashing power allocated to pool m and R_{λ_m} the miner's reward when a block is eventually mined by the pool bringing a total reward R.

The pool manager should make sure to keep paying its PPS miners at a steady rate, compensating for any pool "luck" fluctuations in any given period when trying to find a block. To achieve this, the manager needs to maintain a "bucket" containing an adequate amount of coins, and keep replenishing it with R_{PPS} (i.e. reward of PPS) each time a block is "mined" by the pool with block reward R, to keep paying PPS miners during periods of bad "luck". Consequently, when the pool collectively "mines" a block with reward R, the pool manager can select one of the following three miner payment strategies, which also determine the exact PPLNS miners' reward:

Strategy 1: The pool manager splits R into $R_{\mathsf{PPS}} = (1-z)R$ and $R_{\mathsf{PPLNS}} = zR$. By this strategy, $R_{\lambda_m} = \frac{\lambda_m}{z\Lambda_m} zR = R\frac{\lambda_m}{\Lambda_m}$. In other words, the miner having contributed a hashrate of λ_m will get a reward based on the percent of the hashrate he contributed with respect to the total hashrate of PPLNS miners, multiplied by R_{PPLNS} (i.e. reward of PPLNS).

Strategy 2: The pool manager pays the PPLNS miners based on the total hashrate of the pool Λ_m, then allocate the remainder of the rewards to the PPS bucket. By this strategy, $R_{\lambda_m} = R\frac{\lambda_m}{\Lambda_m}$ which effectively results to the same paid amount as in Strategy 1.

Strategy 3: First replenish the PPS "bucket" based on the total amount $\tilde{R}$ was paid off to the PPS miners since the last block was found, then pay PPLNS based on what is left of the total reward. Following this strategy: $R_{\mathsf{PPLNS}} = R\ -\tilde{R}$ and miner's reward is $R_{\lambda_m} = R_{\mathsf{PPLNS}}\frac{\lambda_m}{z\Lambda_m} = (R-\tilde{R})\frac{\lambda_m}{z\Lambda_m}$. Effectively, the pool manager by this strategy transfers some of his risk to the PPLNS miners.

To our knowledge, no mining pool that offers both PPLNS and PPS reward systems specifies which strategy it follows. Using public data to prove which strategy a mining pool follows is a non-trivial process. We assume that pools offering both PPS and PPLNS reward mechanisms follow either Strategy 1 or 2 which are the most intuitive and produce the same end result for the miners.

III. ACTIVE MINER'S PROBLEM

Our model builds on the recent study of [2], which considers the problem of an "active" miner, who given a set of M mining pools, where each pool m has a total hashrate Λ_m, and fee f_m, wishes to maximize his expected utility, or the weighted average of his expected earnings. In [2] the miner's utility is quantified based on the available pools' parameters and his Constant-Absolute-Risk-Aversion (CARA) ρ, as well as his own available hashing power and the expected reward value. We describe this model (which serves as the base of our schemes) in Section III-A.

A. Mining on a Single Cryptocurrency

We consider an active miner, who owns mining hardware with PoW hashing power λ_{A} and is mining on a single cryptocurrency with total hashrate Λ. Ideally he would like to distribute λ_{A} among M mining pools of different sizes offering different fee structures (while possibly keeping a portion of his power for zero-fee solo mining) in a way that maximizes his expected utility. Doing so is not trivial, as mining pools differ widely in their size and service fees, and in many cases there is no correlation between these two attributes.

A first attempt was done in [2] where the authors introduced an analytical utility function to capture miners' risk aversion when considering mining pools offering a PPLNS reward method. The idea is to express the expected utility of pool-mining, in terms of a miner's PoW hash power λ_{A}, pools' hashrates Λ_m, pools' fees f_m, a CARA value ρ chosen by the miner and block reward R. The CARA value essentially quantifies how risk averse a miner is, where $\rho = 0$ means that the miner is risk neutral. The expected utility is calculated by Equation 1, where an active miner should find the vector $\{\lambda_m\}_{m=1}^{M}$ maximizing his utility value. This vector expresses an allocation of his mining power λ_{A} over M different pools, where λ_m denotes the mining power allocated to pool m:

$$\sum_{m=1}^{M}(\lambda_m + \Lambda_m)(1 - e^{-\rho R_(1-f_m)\frac{\lambda_m}{\lambda_m+\Lambda_m}})+ \\ (\lambda_{\mathsf{A}} - \sum_{m=1}^{M}\lambda_m)(1 - e^{-\rho R}) \qquad (1)$$

under constraints:

$$\sum_{m=1}^{M}\lambda_m \leq \lambda_{\mathsf{A}} \quad \text{and} \quad \lambda_m \geq 0, \forall m \in M.$$

By solving this optimization problem, we are given the optimal distribution of the miner's total hashpower λ_{A} to M pools. Note that the second term of Equation 1 expresses the left-over hashpower for the miner to mine "solo". If the miner is risk-neutral (i.e. $\rho = 0$), solo mining (which has a zero fee) is the optimal solution.

The reasoning behind this model is that the total payoff P of a miner, who has allocated his resources as above, is equal to the weighted sum of each pool's expected reward, with the weights being the miner's hash rate percentage in the pool, plus the expected reward from solo mining, which is expressed as:

$$\mathsf{P} = \sum_{m=1}^{M}\frac{\lambda_m}{\lambda_m + \Lambda_m}R\tilde{N}_{pool,m} + R\tilde{N}_{solo} \qquad (2)$$

where $\tilde{N}_{pool,m} \sim Poisson(\frac{\lambda_m+\Lambda_m}{\Lambda})$ and $\tilde{N}_{solo} \sim Poisson(\frac{\lambda_{solo}}{\Lambda})$. The miner's expected utility $u(\cdot)$ is:

$$u(\cdot) = E[-e^{-\rho\mathsf{P}}] \qquad (3)$$

Notice that for a Poisson distributed variable x with parameter λ, its moment generating function $E[e^{wx}]$ for any parameter w, is given by $e^{\lambda(e^{w}-1)}$. Now from Equations 2 and 3 we can derive Equation 1. A similar reasoning applies to the rest of our models presented below.

B. Allowing Selection of PPS pools

The problem with the model of Eq. 1 is that it restricts the miners to choose between pools that only offer the PPLNS reward method. Below, we provide a model that allows miners to choose mining pools also offering PPS reward systems. In this case, a rational miner will choose to add only the PPS pool that offers the smaller fee, and disregard pools with higher PPS fees. Eq. 1 is then transformed as follows (changes denoted in blue color):

$$\sum_{m=1}^{M}(\lambda_m + \Lambda_m)(1 - e^{-\rho R(1-f_m)\frac{\lambda_m}{\lambda_m+\Lambda_m}}) \\ +(\lambda_{\mathsf{A}} - \lambda_{\mathsf{PPS}} - \sum_{m=1}^{M}\lambda_m)(1 - e^{-\rho R}) + \lambda_{\mathsf{PPS}}(1 - f_{\mathsf{PPS}})\rho R \quad (4)$$

under constraints:

$$\sum_{m=1}^{M}\lambda_m + \lambda_{\mathsf{PPS}} \leq \lambda_{\mathsf{A}}, \quad \lambda_m \geq 0, \forall m \in M, \\ \text{and} \quad \lambda_{\mathsf{PPS}} \geq 0.$$

The logic behind this model, similar to our basic single-cryptocurrency model, is that when assuming a miner who is willing to allocate his mining power towards PPS/PPLNS pools and solo-mining, we add to his total payoff P a constant reward amount proportionate to the power allocated to the PPS pool over the total cryptocurrency's hashpower, minus the pool's fee, expressed as follows:

$$\mathsf{P} = \sum_{m=1}^{M}\frac{\lambda_m}{\lambda_m + \Lambda_m}(1 - f_{\mathsf{PPS}})R\tilde{N}_{pool,m} + \\ R\tilde{N}_{solo} + R\frac{\lambda_{\mathsf{PPS}}(1 - f_{\mathsf{PPS}})}{\Lambda} \qquad (5)$$

From Equations 5 and 3 we derive Equation 4, where the last term $\lambda_{\mathsf{PPS}}(1 - f_{\mathsf{PPS}})\rho R$ expresses a steady income from the PPS pool.

C. Mining Across Multiple Cryptocurrencies

We now consider a miner who owns mining hardware with PoW hash power λ_{A} and wants to maximize his "risk-sharing benefit" value by mining over C different cryptocurrencies and M pools in total, provided that each cryptocurrency $c \in \mathsf{C}$ uses the same PoW mining algorithm α. The allocation of the miner's hashing power λ_{A} will now be $\{\lambda_{m,c}\}_{m=1}^{M}, c \in \mathsf{C}$ and the first constraint in Equation 1 takes the form:

$$\sum_{c\in\mathsf{C}}\sum_{m=1}^{M}\lambda_{m,c} \leq \lambda_{\mathsf{A}}.$$

However, each cryptocurrency c has its own block reward R_c[2] and its own *average* block time D_c. Thus, we also consider the *reward over time* ratio $\frac{R_c}{D_c}$, as it effectively normalizes R_c over different cryptocurrencies. In addition, a miner's hashrate λ_m allocated to pool m that mines cryptocurrency c should be normalized to each cryptocurrency's total hashrate Λ_c. So, in this (more general) case, Equation 1 takes the following form, under the new constraints outlined above:

$$\sum_{c\in\mathsf{C}}\left(\sum_{m=1}^{M}\frac{(\lambda_{m,c} + \Lambda_{m,c})}{D_c\Lambda_c}(1 - e^{-\rho R_c(1-f_{m,c})\frac{\lambda_{m,c}}{\lambda_{m,c}+\Lambda_{m,c}}}) \\ + \frac{\lambda_{0,c}}{D_c\Lambda_c}(1 - e^{-\rho R_c})\right) \qquad (6)$$

[2] We should be careful to express R_c in a fiat currency value (e.g. USD), as we do not take different cryptocurrency exchange rates into consideration.

where $\lambda_{0,c}$ denotes solo mining for cryptocurrency c, and the first constraint of Equation 1 is more precisely expressed as

$$\sum_{c \in \mathsf{C}} \left(\sum_{m=1}^{M} \lambda_{m,c} + \lambda_{0,c} \right) \leq \lambda_{\mathsf{A}}.$$

The above constraint shows that the miner could choose to diversify his solo-mining (which was initially expressed by the second term in Equation 1) over multiple currencies as well, in a similar fashion as he would do by diversifying across multiple mining pools.

D. Mining Across Cryptocurrencies with Different PoW Algorithms

In the previous sections we assumed that the active miner's goal is to maximize his "risk-sharing benefit" value, given that he mines on one or more different cryptocurrencies using the same PoW algorithm. We now generalize our consideration, by allowing a miner to distribute his power across C different cryptocurrencies, and across A different PoW algorithms[3].

Let A be the set of PoW algorithms. Since each algorithm solves a different version of the PoW puzzle, and uses a different set of hash functions, the miner's "total" hashrate λ_{α_i} for each algorithm α_i will change. However the miner might choose to allocate his hardware "power" among different mining PoW puzzles at the same time (in CPU mining, that would require setting priority levels to each Operating System process i). In this case, we can use Equation 6, but its first constraint will become

$$\sum_{\alpha_i \in \mathsf{A}} \sum_{c \in \mathsf{C}} \sum_{m=1}^{M} \frac{\lambda_{m,c}}{\lambda_{\alpha_i}} \leq 1.$$

Example: Assume a miner who owns some amount of computational power (CPUs and/or GPUs). With his hardware, he could mine exclusively cryptocurrency c_1 which uses PoW mining algorithm α_1, and his maximum hashrate would be λ_{α_1}. Alternatively, he could mine exclusively coin c_2 which uses PoW mining algorithm α_2, at a hashing rate of λ_{α_2}. Now he wishes to diversify his risk among these two cryptocurrencies (for simplicity we assume that he chooses to mine them only on a single pool each). The resulting constraint would be

$$\frac{\lambda_{1,1}}{\lambda_{\alpha_1}} + \frac{\lambda_{2,2}}{\lambda_{\alpha_2}} \leq 1.$$

Each term represents the "percentage" of the miner's CPU (and/or GPU) power devoted to mining on a specific PoW algorithm. The sum of the ratios cannot exceed 1 which represents the total CPU and/or GPU power of the hardware[4].

E. Transaction Fees in Mining Pools

In the early days of cryptocurrencies, transaction fees were negligible compared to block rewards. However during recent periods, transaction fees have risen to considerable amounts, especially in the case of Bitcoin [17]. It is up to the pool manager to decide if the transaction fees are distributed to the participating miners or is kept by the pool for profit. However, especially during periods of high transaction fees, not including these fees to the miners' reward R can be thought intuitively as a "hidden fee". Let tx_c the *average* transaction fee for cryptocurrency c observed during a recent period of time. Also we set $\mathsf{tx}_{m,c} = 1$ if the pool pays transaction fees to the miner, else we set $\mathsf{tx}_{m,c} = 0$. We modify Equation 6 to include (non-negligible) transaction fees as follows (changes denoted in color, the equations for other cases can be modified in a similar way):

$$\sum_{c \in \mathsf{C}} \left(\sum_{m=1}^{M} \frac{(\lambda_{m,c} + \Lambda_{m,c})}{D_c \Lambda_c} (1 - e^{-\rho(R_c + \mathsf{tx}_c \mathsf{tx}_{m,c})(1 - f_{m,c}) \frac{\lambda_{m,c}}{\lambda_{m,c} + \Lambda_{m,c}}}) + \frac{\lambda_{0,c}}{D_c \Lambda_c} (1 - e^{-\rho(R_c + \mathsf{tx}_c \mathsf{tx}_{m,c})}) \right). \tag{7}$$

IV. Implementation of Our Model

In this Section we present an implementation of our mining resources allocation mechanism. We developed a Python tool that automates the decision for an active miner, who owns either commercial off-the-shelf (COTS) hardware (e.g. CPUs/GPUs) or application-specific integrated circuit hardware (ASICs) [5]. Our tool covers all the cases discussed in Section III.

Choosing the Right Optimization Method: In order to find the best possible allocation of the miner's hash power we tried a few optimization methods. First, we applied the sequential least squares programming algorithm (SLSQP), which uses the Han-Powell quasi-newton method with a BFGS update of the B-matrix and an L1-test function for the steplength algorithm [18]. Second, we implemented a modification of Newton's method that solves the Lagrange system of equations for the active constraints. To our surprise both mentioned gradient based optimization methods experienced difficulties in obtaining accurate solutions to the optimization problem for very small values of ρ, about 10^{-5}. A possible explanation to that phenomenon is that the instances with a wide range of hashes/sec from a few to quintillion 10^{18}, make the gradients calculated with significant computational errors. The presence of exponential functions sensitive to the scale of their argument is a contributing factor for the loss of the accuracy for the obtained gradients under the finite precision computer arithmetic. Even though those methods could be used for the cases with larger values of ρ, we abandoned them.

The most successful algorithm for the optimization problem was the constrained optimization by linear approximation (COBYLA) [4]. COBYLA was developed for solving nonlinear constrained optimization problems via a sequence of linear programming subproblems, each solved on an updated simplex. COBYLA is a good fit for our optimization problem

[3] This of course assumes that the miner owns CPU/GPU mining hardware, since ASICs are restricted to specific PoW algorithms.

[4] In our assumption we do not take a "dual mining" GPU setup into account.

[5] Our implementation can be accessed at http://smart-miner.cs.gmu.edu/

for the following reasons. First, our feasible set is a simplex, so the vertices of the feasible set form a good initial linear approximation. Second, our problem is low dimensional (no more than a dozen of variables). The low dimensionality of the problem results in a relatively small number of simplexes that need to be constructed before the solutions is found. Finally, COBYLA is a gradient free algorithm. Therefore to update iterates, COBYLA does not rely on the gradient obtained locally in one point, which may be not accurate for this problem. Instead, in its search COBYLA relies on the slope of a linear n-dimensional approximation calculated out of readily available n+1 feasible points. We believe that all these factors together contribute to the efficiency of the algorithm for finding the best possible allocation of the miner's hash power. Therefore, our tool utilizes COBYLA optimization solving method.

Description and Instantiation Assumptions: The basic single - cryptocurrency version of our tool, given as input the miner's hashing power λ_{A}, coin's exchange rate E, chosen pool data $[\Lambda_i, f_i]_{i=1}^{M}$ (pool total hashrate and fee respectively) and risk aversion ρ, outputs the optimal distribution of his hashpower over these pools plus a "solo-mining" remainder. Some instantiations of our tool are outlined in Section IV-A as examples for a typical value range of ρ. Our tool can also provide the optimal distribution for the "multi-cryptocurrency, single PoW algorithm" (Section III-C) and "multi-cryptocurrency, multi-PoW algorithm" (Section III-D) extensions, and we also outline an extended instantiation in Section IV-B.

The results of our tool can be easily applied for mining in large-scale, where a miner can allocate a portion of his hardware to mine on a specific pool. The process of applying the results on a single mining hardware piece is not trivial, as to our knowledge, no ASIC or GPU miner application exists that enables the user to allocate his mining power over many pools by a specific percentage, even if theoretically it's technically feasible. The majority of ASICs utilize a fork of the `cgminer` tool, which initially offered a "multipool strategy" option for the miner, but was later deprecated as it was no longer compatible with the modern stratum mining protocol [19]. Multi-pool mining in a round-robin fashion is not efficient for the miner as well, as this would result to a decrease in his overall reward, given the nature of reward schemes that prevent pool "hopping". We encourage ASIC manufacturers and mining application developers to enable user-specified multi-pool mining in future releases, for the benefit of the miners and the whole community.

We assume that the miner possess an average wealth of $\mathcal{W}$ = \$100k, while having typical values for the constant relative risk aversion CRRA metric between 1 and 10 [20]. Given that CARA = CRRA/$\mathcal{W}$, we take as typical values for CARA ρ between $10^{-5} - 10^{-4}$, which we mostly assume throughout the rest of this paper. Note that changing our assumption for our miner's wealth is equivalent to changing the typical value range for ρ accordingly (we include additional evaluation analysis for a broader range of ρ values).

A. Single Cryptocurrency

1) Evaluation I: We instantiate the first experiment of our tool by using the following parameters: a miner with total hashpower λ_{A} = 40 hashes/sec, wishing to mine on a single cryptocurrency with block reward R = \$50000, having picked 4 mining pools with parameters shown in Table II: These values do not correspond to "real" mining pools (or an existing cryptocurrency), but are representatives for different classes of pools in terms of relative size, as larger real-world pools charge higher fees (but have less fluctuations on the miner's income) while smaller pools have lower (or even zero) fees to attract new miners to them. The results depicted on Figure 1 show that our model produces the expected choices for rational miners. For smaller values of ρ the miner is willing to "risk" more, and would dedicate much of his hashpower to the small Pool 4, but for larger values of ρ the miner would diversify among larger pools for a steadier income. Another important observation is that for $\rho > 6 \cdot 10^{-5}$ the miner would allocate some of his power at both Pools 1 and 2 to diversify his risk (which are the "largest" pools, having the same 2% fee), although he would show a strong preference for Pool 1 which is 10 times larger than Pool 2. Note that for simplicity, we do not take any transaction fees kept by pools into account, however using the methodology discussed in Section III-E our evaluation would produce equivalent results.

TABLE II. EVALUATION I POOL PARAMETERS.

Pool 1	Λ_1	10^6 hashes/sec
	f_1	2%
Pool 2	Λ_2	10^5 hashes/sec
	f_2	2%
Pool 3	Λ_3	10^4 hashes/sec
	f_3	1%
Pool 4	Λ_4	10^3 hashes/sec
	f_4	0%

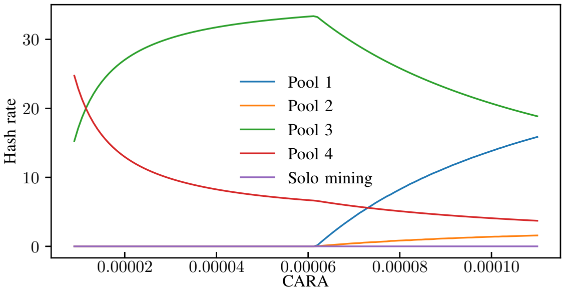


Fig. 1. Single cryptocurrency diversification.

2) Evaluation II: We then pick some actual Bitcoin pools: Slush pool, ViaBTC and KanoPool. These pools, as indicated by their parameters shown in Table III (values as of February 2019) are representatives of the options available to a miner, as they cover a wide range of pool hash-power Λ_m and pool fee f_m. The above pools use either PPLNS or Score (variant of proportional) reward methods. We do not include a PPS pool in this example, although taken into account in Equation 4, as the results turned out to be identical for the typical value range of ρ. Using our parameters, a PPS pool would participate in the diversification only for large values of ρ that are not within the typical range (we show such an example later in this section). For the other parameters, we consider a large-scale miner who owns total mining power of λ_{A} = 3000 TH/sec (roughly about 100 units of Antminer S15 ASICs), and the Bitcoin total block

TABLE III. EVALUATION II POOL PARAMETERS

Slush pool	Λ_1	4040 PH/sec
	f_1	2%
ViaBTC	Λ_2	3090 PH/sec
	f_2	2%
KanoPool	Λ_3	48 PH/sec
	f_3	0.9%

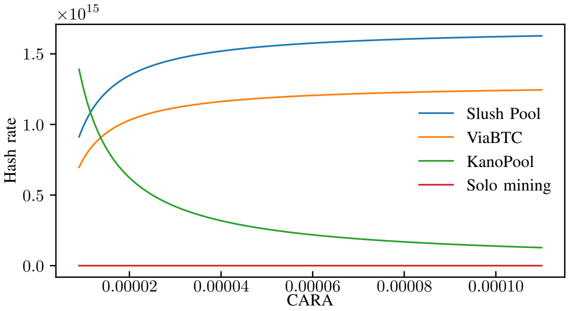


Fig. 2. Large-scale miner on Bitcoin pools.

reward R = $45441 [6]. The pool parameters are shown in Table III and the resulting diversification graph in Figure 2, where we observe a similar pattern to the previous "representative" pools example (i.e. a miner's preference for larger pools and steadier income as ρ increases).

3) Diversifying on a PPS Pool: In the previous evaluation we showed that a PPS pool would not participate in the diversification using parameters for actual pools shown in Table III and typical values for ρ. In Figure 4 we show how a PPS pool would affect a miner's diversification for non-typical large values of ρ, using the parameters in Table IV. This would be applicable only for a miner who is very risk-averse, as he would show a stronger preference to the steady income a PPS pool provides, as the value of ρ increases.

4) Small-scale miners: An interesting observation is in regards of smaller scale miners and pools with higher fees. For instance, an active miner with 10 ASICs instead of 100, when following our method, would allocate all his hash power to the smaller pool (KanoPool) with the lowest fee. In Figure 3 we show a small-scale (or "home") Bitcoin miner as an example (λ_A = 125 TH/sec). Given his relatively small hashpower, he would only choose the lowest-fee pool to mine (KanoPool), without allocating any resources to larger pools with higher fees, except for the upper values of ρ. Essentially, it is shown that risk-aversion has less effect on small-scale miners.

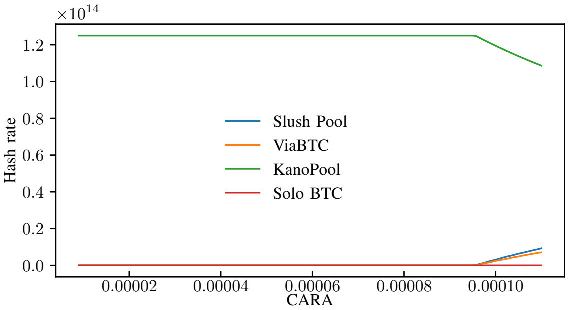


Fig. 3. Small-scale miner on Bitcoin pools, parameters as in Table III.

TABLE IV. BITCOIN POOL PARAMETERS INCLUDING PPS POOL

Slush pool	Λ_1	4040 PH/sec
	f_1	2%
ViaBTC	Λ_2	3090 PH/sec
	f_2	2%
KanoPool	Λ_3	48 PH/sec
	f_3	0.9%
PPS Pool	f_4	4%

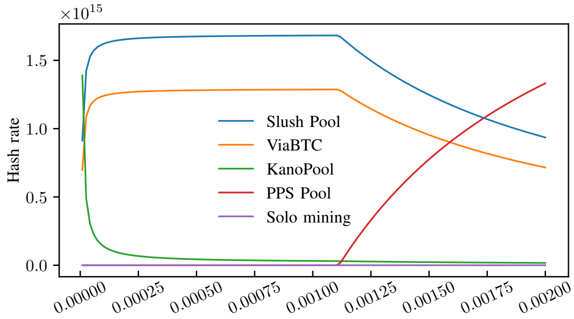


Fig. 4. Single currency with PPS pool and large values of ρ.

B. Multiple Cryptocurrencies

We now consider a miner who diversifies over different cryptocurrencies. For simplicity, we just switch[7] the currency in the 2nd pool (ViaBTC) from Bitcoin to Bitcoin Cash. The pool parameters are shown in Table V, Λ_{BTC} = 42.33 EH/sec, Λ_{BCH} = 1.43 EH/sec and R_{BCH} = $1547 [6]. The resulting graph in Figure 5 show the diversification of his computational power for various values of ρ. We observe that in this instance, for small values of ρ his optimal strategy would be to keep most of his resources for zero-fee Bitcoin solo mining. However, for increasing values of ρ he would diversify his power to larger pools, and he would also choose to allocate some of his power to the Bitcoin Cash pool, even though it has the same fee and the pool might not be as profitable as the Bitcoin pool.

Impact of Exchange Rates: We noted that our results are highly sensitive even to very small changes to any of the parameters, such as the exchange rates. For instance, the same miner having chosen the same pools, based on the historical data, would allocate all of his power to the Bitcoin Cash pool the previous day, while after a few days he would transfer all of his power to the Bitcoin pools. In Figures 6(a) and 6(b) we show how small daily fluctuations in the exchange rate between two same-PoW cryptocurrencies can affect the miner's diversification for these cryptocurrencies. While in our previous instantiation the exchange rate was BTC/BCH = 0.034, a small increase in favor of Bitcoin Cash's value totally eliminates the presence of Bitcoin pools from the diversification, leaving only the Bitcoin Cash pool and Bitcoin Cash solo mining for the miner as his options. On the other hand, a small increase in favor of Bitcoin's value eliminates the presence of the Bitcoin Cash pool, and the miner would only diversify among the Bitcoin pools.

[6] Parameters as of Jan 25 2019, retrieved from https://btc.com/ and https://bitinfocharts.com/.

[7] Many pools as shown in the Appendix host pool mining services for multiple cryptocurrencies.

TABLE V. MULTI-CURRENCY POOL PARAMETERS

Slush pool	Λ_1	4040 PH/sec
	f_1	2%
ViaBTC (Bitcoin Cash)	Λ_2	135 PH/sec
	f_2	2%
KanoPool	Λ_3	48 PH/sec
	f_3	0.9%

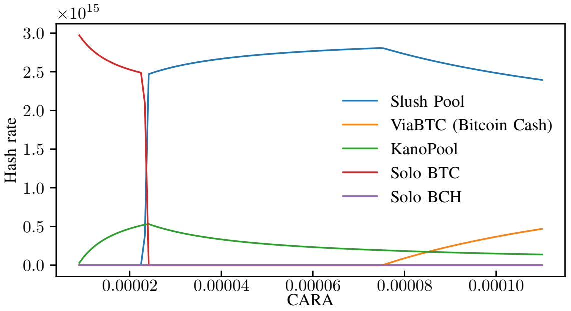


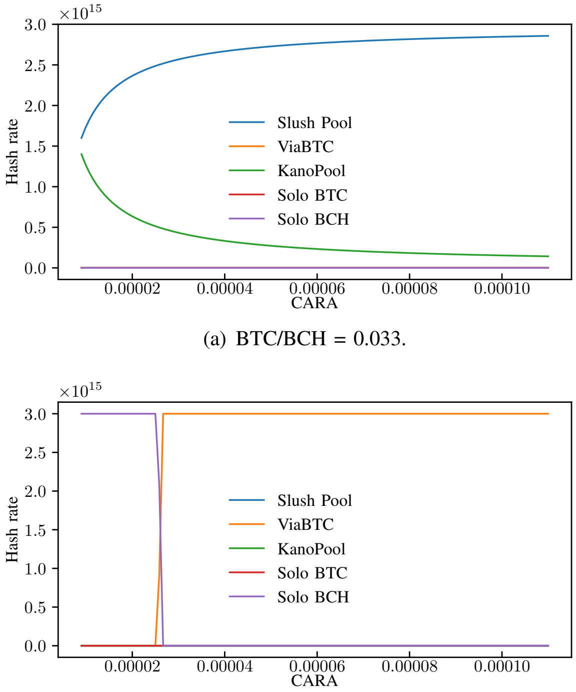
Fig. 5. Large-scale miner on SHA-256 pools.

V. EVALUATION AND SIMULATED RESULTS

To showcase the advantage (and risks) of diversifying over multiple pools, we present an evaluation of our model using Bitcoin data extracted from Smartbit Block Explorer API [21], as shown in Table VI. First, we consider a Bitcoin miner owning some hashing power λ_{A} = 1200TH/sec, who would start mining passively on a single chosen pool on February 1st 2018 for Δ = 4 months. Then, we consider a Bitcoin miner having the same hashing power λ_{A}, who using our tool over the same time period, would "actively" diversify every $t = 3$ days over $M = 3$ Bitcoin pools of his choosing and reallocate his hashpower accordingly. In Table VII we outline the mining pools chosen for our evaluation along with their respective fees and other simulation parameters. We choose these pools as representatives of different pool sizes and fees, and to show how such pools influence the active miner's diversification over time. As discussed in Section IV, we pick the mean value for ρ = 0.00005. Note that as we mentioned in the introductory section, all of the above parameters constitute several degrees of freedom in our experiments. In the subsequent sections we show how each one of them can affect the end result derived from our main evaluation.

We retroactively compute the earnings on a daily basis for both miners. To take both overall reward and miner's income variance into account, we utilize the Sharpe ratio $\mathsf{S} = \frac{\mathsf{P}-\mathsf{P}_{\mathsf{PPS}}}{\sigma_\Delta}$ as our main comparison metric, where P is the total accumulated reward of the miner during the time period Δ, $\mathsf{P}_{\mathsf{PPS}}$ is the estimated miner's reward during time period Δ using a PPS pool[8] and σ_Δ stands for the Standard Deviation of miner's reward over the time period Δ.

Remark. Here we should note that while the CARA utility function is used to capture preferences over different wealth levels, one could argue that the miner should be amortizing his costs and revenue over a time period, since the *average* payoff per block would have little variance and a diversification over multiple pools is not necessary. However the sum (or the discount rate weighted sum) of many Poisson variables linearly scales both the mean and variance of each single Poisson variable, so our analysis, which apparently looks like "to evaluate the reward from a single block", is indeed equivalent to capturing preferences over different wealth levels (this trap is a common "fallacy of large numbers" [22], [23]).



(a) BTC/BCH = 0.033.

(b) BTC/BCH = 0.035.

Fig. 6. Large-scale miner on SHA-256 pools, parameters in Table V

A. Evaluation Assumptions

We assume that both miners choose pools that do not use the PPS reward scheme and that all pool's fees and reward schemes remain constant over time. In addition, the miners are assumed to have constant hashpower (i.e. do not use any of their rewards to buy more mining hardware), and convert their rewards into USD on a daily basis. For our analysis, we derive the daily network hashrate Λ_d from the daily network difficulty T_d using the approximation $\Lambda_d = \frac{2^{32}}{600} T_d$. We also approximate the daily pool hashpower by $\Lambda_{m,d} = \frac{B_{m,d}}{B_d}$ where $B_{m,d}$ the number of blocks found by pool m and B_d the total actual number of blocks found on each day d. The smaller the mining pool however, the less precise this approximation becomes (i.e. a small pool might be "unlucky" and would not find a block for several consecutive days, while on some day it might become "lucky" and find several blocks in a single day), so we employ averaging techniques over a time window of 14 days (which we believe is a reasonable period) to improve our approximation accuracy for computing $\Lambda_{m,d}$. However a miner in the "real world" would better use the self-reporting pool hashrates (based on submitted shares) for a more precise result (to our knowledge, such historical data is not available on any block explorer). Also as noted before, we do not take any transaction fees kept by pools into account for simplicity purposes. However, a miner can use the methodology in

[8] We approximate $\mathsf{P}_{\mathsf{PPS}}$ using a large Bitcoin pool which offers a relatively steady income, and subtract f_{PPS} from those rewards.

TABLE VI. DATA FROM BITCOIN BLOCKCHAIN.

Days $\in \Delta$	d
Exchange rate	E_d
Network difficulty	T_d
Participating pools	m_d
Total number of blocks	B_d
Number of blocks found by pool m	$B_{m,d}$

TABLE VII. SIMULATION PARAMETERS

Miners' hashpower λ_{A}	1200TH/sec
Diversification interval t	3 days
Fee $f_{\mathsf{SlushPool}}$	2%
Fee f_{ViaBTC}	2%
Fee f_{DPOOL}	1%
CARA ρ	0.00005

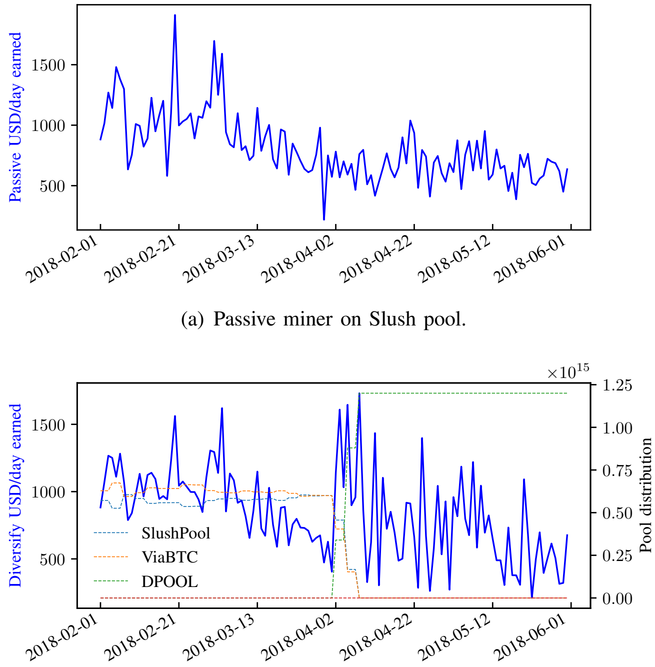
Section III-E to compute the average transaction fees over a recent period of time from online blockchain explorers and make a projection for fees in the future, then add $\mathsf{tx}_{m,c}$ and tx_c to Equation 7 accordingly. Finally, we show the earnings in USD instead of cryptocurrency (Bitcoins) in order to take the exchange rate into account, which is an important parameter for the active miner (used to calculate the block reward R).

B. Main Evaluation Results

In Figure 7 we show the earnings over time for both of the miners for comparison (in Figure 7(b) we also show how the diversification changes over time in light colors). We observe a slightly increased variance for the active miner (blue line spikes), since he chose to include a third pool (DPOOL) with smaller total hashpower Λ. However his total accumulated reward would be $\mathsf{P_A}$ = \$101221 for the active miner, compared to $\mathsf{P_P}$ = \$97101, which eventually leads to a Sharpe ratio $\mathsf{S_A}$ = 0.156 compared to $\mathsf{S_P}$ = 0.060. Note that since we assumed both miners' hashpower λ_{A} remains constant throughout this period, we observe a general decline pattern in their daily reward, because of the increasing difficulty T directly affecting $\frac{\lambda_{\mathsf{A}}}{\Lambda}$. We also include an equivalent analysis using the same pools over an one year period in the next evaluation, where such a decline can be observed more clearly. Lastly, an important parameter to consider is the block time D [24]. We used Bitcoin for our simulation, where D is relatively high (roughly 10 minutes) - this resulted in high reward variance, especially when using smaller pools as shown in Figures 7 and 8. If we used a cryptocurrency with more frequent blocks for our evaluation (e.g. Ethereum), then the result would be more predictable with lower observed overall variance values.

C. Analysis for a Large Δ

In Figure 8 we repeat the simulation discussed in Section V-B over the period of Δ = 1 year (January 1 2018 - December 31 2018). The decline of miner rewards due to the increasing difficulty can be observed more clearly. In such a case, a miner would most likely reinvest his earnings on mining hardware to keep $\frac{\lambda_{\mathsf{A}}}{\Lambda}$ as steady as possible. We also observe high variance around January 2018, generated by the high volatility in the Bitcoin/USD exchange rate E_{BTC}. The results for this evaluation are $\mathsf{P_A}$ = \$233293 and $\mathsf{S_A}$ = 0.064 vs. $\mathsf{P_P}$ = \$224293 and $\mathsf{S_P}$ = 0.023, which are consistent with the results derived from the 4-month simulation.



(a) Passive miner on Slush pool.

(b) Active miner on 3 pools.

Fig. 7. Hash power distribution over 4 month data

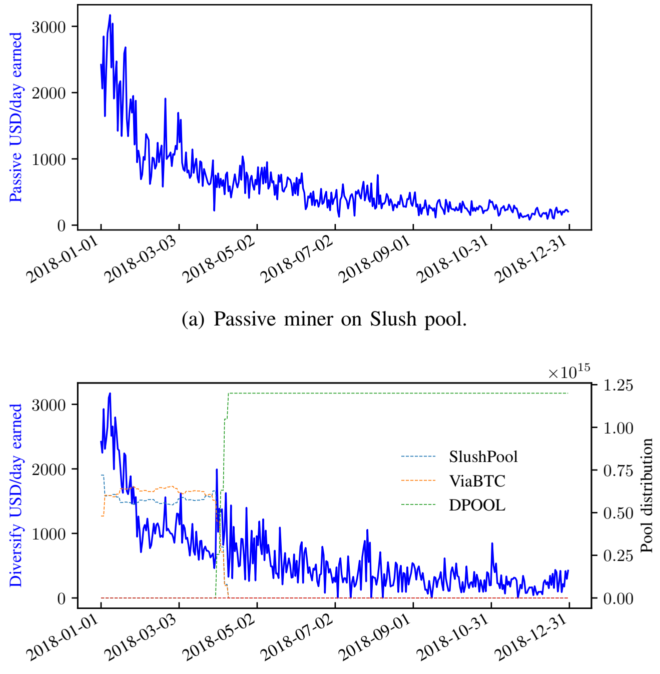
D. Analysis for a Different Δ and Different Pools

We now replace the third small pool (DPOOL) from our default set of pools with a larger one (AntPool) and set our period from (January 1 2018 - June 30 2018) for Δ = 6 months, thus diversifying over the three largest PPLNS pools during that period. Again we observe an improvement in our metrics, $\mathsf{P_A}$ = \$172719 and $\mathsf{S_A}$ = 0.041 vs. $\mathsf{P_P}$ = \$172092 and $\mathsf{S_P}$ = 0.032. Detailed analysis shown in Figure 9.

E. Analysis for Different Values of ρ

We now repeat our retroactive analysis discussed in Section V-B by setting ρ = 0.0001, which is at the upper bound of our considered typical values. By comparing Figure 10 with Figure 7(b), we observe that the miner chose to diversify on the smaller pool (DPOOL) less frequently, since he is more "sensitive" to risk. As expected, this translates to a lower-variance graph and a more steady income. However his overall reward decreases, which offsets the previous benefit. Having kept the rest of the analysis parameters to our original default values, the miner's total accumulated reward would be now $\mathsf{P_A}$ = \$99082, and the Sharpe ratio would be $\mathsf{S_A}$ = 0.104.

By further experimenting with a broader value range for ρ, we derive Figure 11, where we observe a decreasing trend for the Sharpe ratio as ρ increases. This may be counterintuitive at first sight, as traditional portfolio theory would otherwise predict a flat relationship between a strategy's Sharpe ratio and an investor's risk aversion. The reason for this difference is that our model captures any potential impact of a miner's decisions to the whole equilibrium. If the miner is relatively small, his effect on the equilibrium is negligible through first-order Taylor expansion. In this case however, the miner is so



(a) Passive miner on Slush pool.

(b) Active miner on 3 pools.

Fig. 8. Hash power distribution over 1 year data

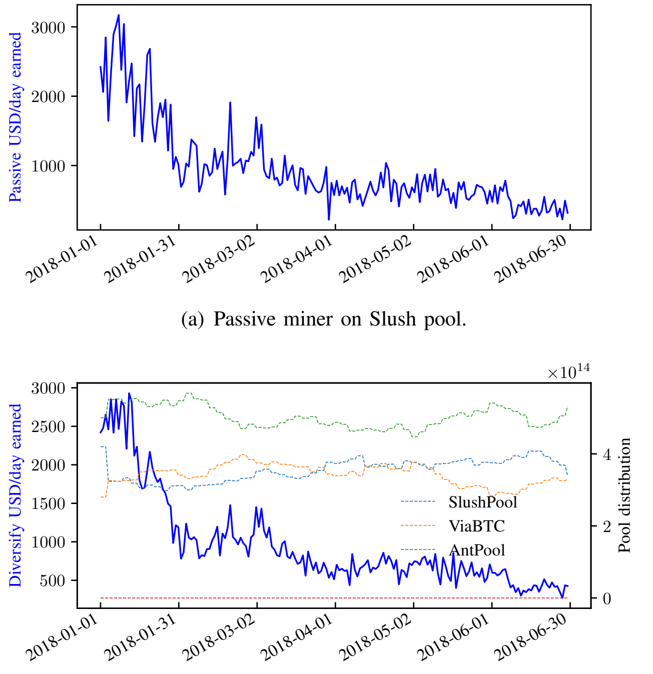
large that his power is comparable to the third pool (DPOOL), and his effect on the whole equilibrium is no longer negligible, leading to the declining graph. If we replace DPOOL with a larger pool (Antpool) as shown in Figure 12, we observe that the chosen ρ has eventually no effect on the Sharpe ratio.

F. Analysis for Different Values of Miner Power

By repeating our main evaluation for several different values of miner's computational power, we derive Figure 13 where we observe a negative correlation between Sharpe ratio and a miner's hash rates. As in the previous case, if the miner is large enough compared to the pools, our model captures his effect on the whole equilibrium which is negligible in typical portfolio analyses. When we replace the small pool (DPOOL) with a larger one (Antpool) and repeat our experiment, the traditional insight from portfolio theory reemerges: As shown in Figure 14, for a relatively small miner, we now observe no significant correlation with miner hash rate and the resulting Sharpe ratio.

G. Analysis for Different Diversification Intervals

While our main evaluation assumed the active miner runs our tool every 3 days, in Figure 15 we show how different diversification intervals affect the Sharpe ratio. The general observation is that small intervals (less than 1 week) help slightly to improve the results, while large intervals (more than 1 month) are generally not recommended. After all, if the time period of hashpower reallocation becomes very large, the miner is not very "active" and his behavior matches more that of a passive miner.



(a) Passive miner on Slush pool.

(b) Active miner on 3 pools.

Fig. 9. Hash power distribution over 6 months data and different pool set.

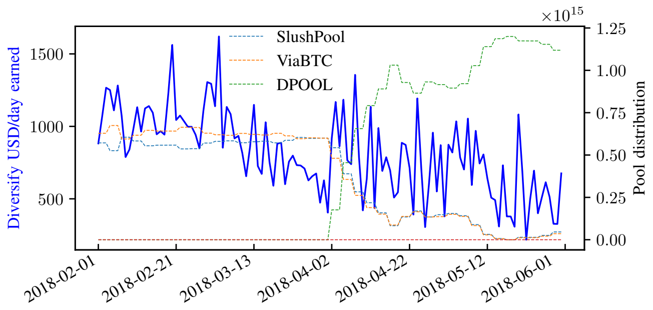


Fig. 10. Active miner diversifying over Bitcoin pools with increased $\rho = 0.0001$.

H. Analysis for Different Number of Available Pools

In Figure 16 we examine how our main evaluation metrics are affected both by the total number and the specific pools available to the miner. We observe that if the miner only picks one pool (e.g. ViaBTC), effectively he can only diversify between that pool and solo mining, which usually matches a passive miner for a typical value of ρ. However, as the miner includes additional pools into his consideration, his Sharpe ratio tends to increase, which indicates that a rational miner should consider as many pools as possible. However, given that we performed a retroactive analysis, adding "bad luck" pools into the miner's available pool set does not improve his Sharpe ratio any further.

I. Analysis for Different Cryptocurrencies

As discussed in Section III-C, our model also considers miners who diversify over multiple cryptocurrencies which use

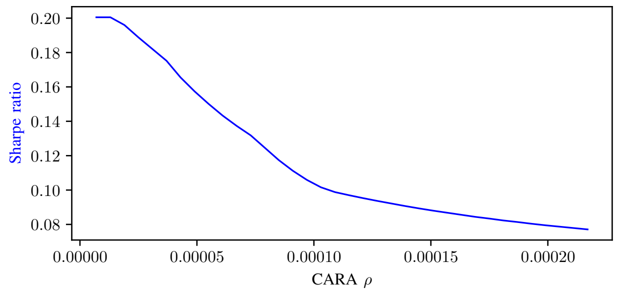


Fig. 11. Sharpe & rewards vs ρ, SlushPool - ViaBTC - DPOOL.

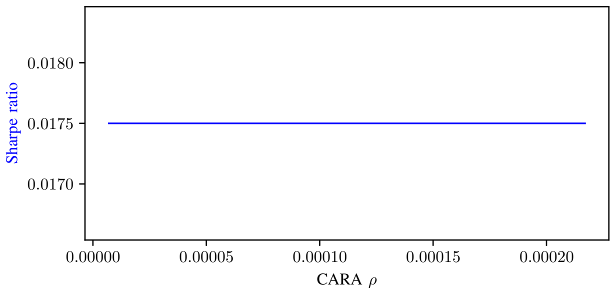


Fig. 12. Sharpe & rewards vs ρ, SlushPool - ViaBTC - AntPool.

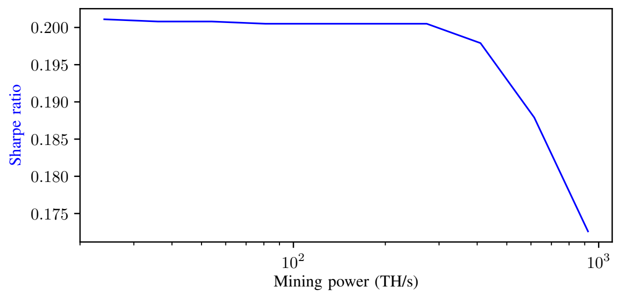


Fig. 13. Sharpe & rewards vs mining power, SlushPool - ViaBTC - DPOOL.

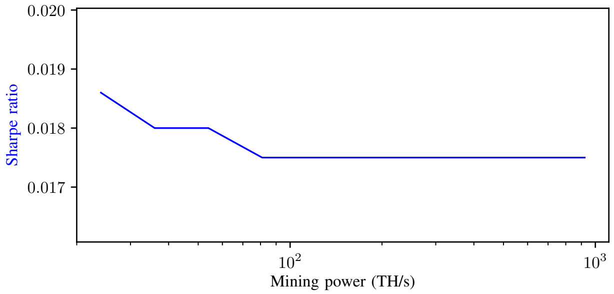


Fig. 14. Active/Passive Sharpe ratios vs. mining power, SlushPool - ViaBTC - AntPool.

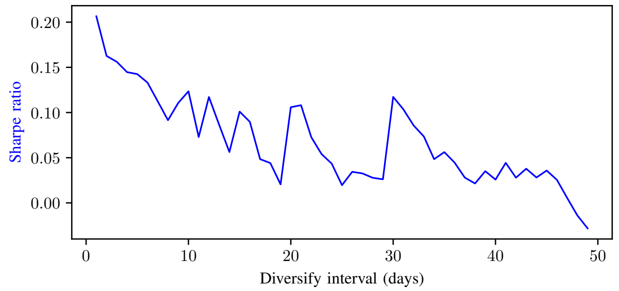


Fig. 15. Sharpe & rewards vs diversification interval t.

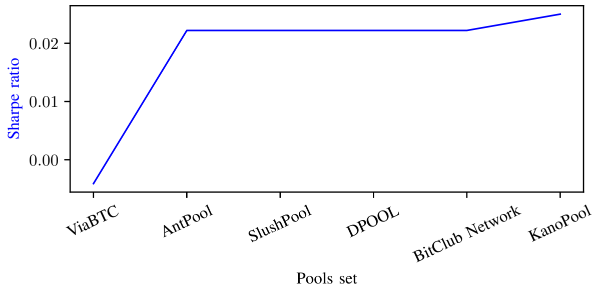


Fig. 16. Sharpe & rewards vs type and order pools.

the same PoW algorithm. Extending our evaluation results to such a case is relatively straightforward (assuming the equivalent data shown in Table VI are available for *all* considered cryptocurrencies) and we expect a similar derivation of results, as the only additional parameter in the problem is the reward over time ratio $\frac{R_c}{D_c}$, normalized to each cryptocurrency's total hashrate Λ_c, and a "passive" miner would just choose the most beneficial pool/cryptocurrency in the beginning of the experiment, without however taking any future changes of the above parameters into account.

We also note that the case of diversifying over different cryptocurrencies which *also* employ different PoW algorithms as discussed in Section III-D is hard to execute in practice, since as discussed it excludes ASIC hardware, while it requires fine-tuning process priorities in CPUs and GPUs.

VI. Conclusions

We present an analytical tool that allows risk-averse miners to optimally create a mining portfolio that maximizes their risk-adjusted rewards, using a theoretical model that optimally allocates miner's resources over mining pools based on their risk aversion levels. We provide multiple extensions of the base model to enable miners to optimally distribute their power between mining pools of different cryptocurrencies, which might even use different PoW algorithms. Then, we develop an analytical tool publicly available (as provided in Section IV) for miners to compute their optimal hash power allocation based on their inputs, and we present both time-static and historical-retroactive evaluations of our tool. The retroactive evaluation results show a direct benefit for the individual miner in terms of reward amount over reward standard deviation ratio (expressed by the Sharpe ratio).

As a final note, it is often argued that the massive participation on mining pools has lead to blockchain centralization (e.g. in bitcoin at the time of writing, over 50% of mining is done by 4 mining pools). Lack of decentralization can lead to various types of attacks, including double-spending, reversing confirmations of previous transactions or transaction censoring [25], [26], [27], [28]. Mining pools (especially mining pools of larger sizes) have been criticized for leading to a high rate of centralization. A number of academic works have studied the level and concerning effects of the centralization trend [2], [24], [28] while a number of solutions have been proposed spanning from decentralized mining pools (P2Pool [29] for Bitcoin), to alternative, non-outsourceable PoW mechanisms [30]. We believe that tools like the one we present here, are positive steps towards the "centralization" problem of PoW systems. Our tool provides incentives to miners in order for them to actively diversify among different pools and cryptocurrencies, potentially increasing power of a large number of smaller pools, while at the same time could also provide insights to mining pool managers in terms of how would rational miners behave.

Acknowledgement

Foteini Baldimtsi and Jiasun Li would like to thank the George Mason Multidisciplinary Research (MDR) Initiative for funding this project.

References

[1] "List of bitcoin mining asics," last accessed 19 Feb 2019. [Online]. Available: https://en.bitcoin.it/wiki/List_of_Bitcoin_mining_ASICs

[2] L. W. Cong, Z. He, and J. Li, "Decentralized mining in centralized pools," 2018. [Online]. Available: https://ssrn.com/abstract=3143724

[3] H. Markowitz, "Portfolio selection*," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[4] M. J. D. Powell, *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*. Dordrecht: Springer Netherlands, 1994, pp. 51–67. [Online]. Available: https://doi.org/10.1007/978-94-015-8330-5_4

[5] W. F. Sharpe, "Mutual fund performance," *The Journal of Business*, vol. 39, no. 1, pp. 119–138, 1966. [Online]. Available: http://www.jstor.org/stable/2351741

[6] "Multipoolminer," last accessed 15 Jan 2019. [Online]. Available: https://multipoolminer.io/

[7] "Smartmine," last accessed 15 Jan 2019. [Online]. Available: https://www.smartmine.org/

[8] "Minergate," last accessed 15 Jan 2019. [Online]. Available: https://minergate.com/

[9] B. Fisch, R. Pass, and A. Shelat, "Socially optimal mining pools," in *Web and Internet Economics*, N. R. Devanur and P. Lu, Eds. Cham: Springer International Publishing, 2017, pp. 205–218.

[10] G. Bissias, B. N. Levine, and D. Thibodeau, "Using economic risk to model miner hash rate allocation in cryptocurrencies," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2018 International Workshops, DPM 2018 and CBT 2018, Barcelona, Spain, September 6-7, 2018, Proceedings*, 2018, pp. 155–172. [Online]. Available: https://doi.org/10.1007/978-3-030-00305-0_12

[11] A. Spiegelman, I. Keidar, and M. Tennenholtz, "Game of coins," *CoRR*, vol. abs/1805.08979, 2018. [Online]. Available: http://arxiv.org/abs/1805.08979

[12] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '15. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927. [Online]. Available: http://dl.acm.org/citation.cfm?id=2772879.2773270

[13] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Communications Letters*, pp. 1–1, 2018.

[14] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *CoRR*, vol. abs/1112.4980, 2011. [Online]. Available: http://arxiv.org/abs/1112.4980

[15] "Comparison of mining pools," last accessed 15 Jan 2019. [Online]. Available: https://en.bitcoin.it/wiki/Comparison_of_mining_pools

[16] "Coinotron," last accessed 15 Jan 2019. [Online]. Available: https://coinotron.com

[17] "Bitcoin avg. transaction fee historical chart," last accessed 19 Feb 2019. [Online]. Available: https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

[18] D. Kraft, *A Software Package for Sequential Quadratic Programming*, ser. Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center - Institute for Flight Mechanics, Koln, Germany, 1988.

[19] C. Kolivas, "Cgminer," last accessed 15 Jan 2019. [Online]. Available: https://github.com/ckolivas/cgminer

[20] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory*. Oxford University Press, 1995.

[21] "Smartbit bitcoin block explorer," last accessed 15 Jan 2019. [Online]. Available: https://www.smartbit.com.au/

[22] S. A. Ross, "Adding risks: Samuelson's fallacy of large numbers revisited," *Journal of Financial and Quantitative Analysis*, vol. 34, no. 3, pp. 323–339, 1999.

[23] P. A. Samuelson, "Risk and uncertainty: A fallacy of large numbers," 1963.

[24] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," *CoRR*, vol. abs/1801.03998, 2018. [Online]. Available: http://arxiv.org/abs/1801.03998

[25] I. Eyal, "The miner's dilemma," in *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 89–103.

[26] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *FC 2014*, ser. LNCS, N. Christin and R. Safavi-Naini, Eds., vol. 8437. Springer, Heidelberg, Mar. 2014, pp. 436–454.

[27] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *FC 2016*, ser. LNCS, J. Grossklags and B. Preneel, Eds., vol. 9603. Springer, Heidelberg, Feb. 2016, pp. 515–532.

[28] Y. Sompolinsky and A. Zohar, "Bitcoin's underlying incentives," *Queue*, vol. 15, no. 5, pp. 50:29–50:52, Oct. 2017. [Online]. Available: http://doi.acm.org/10.1145/3155112.3168362

[29] "P2pool," last accessed 15 Jan 2019. [Online]. Available: http://p2pool.in/

[30] A. Miller, A. E. Kosba, J. Katz, and E. Shi, "Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions," in *ACM CCS 15*, I. Ray, N. Li, and C. Kruegel:, Eds. ACM Press, Oct. 2015, pp. 680–691.

APPENDIX A
MINING POOLS FOR MAJOR CRYPTOCURRENCIES

In Table VIII we summarize a list of mining pools for major cryptocurrencies and the reward type each offers, as of February 15 2019. PPS, PPLNS and proportional we already discussed in Section II-B. In the following table we also come across some variants of the standard reward types. In particular, FPPS (Full Pay Per Share) is similar to PPS but payments take average transaction fees into account, Score is based on the proportional reward method weighed by time the share was submitted, and Exponential is a PPLNS variant with exponential decay of share values. We manually collected the data using the respective mining pool websites and the cryptocurrencies' block explorers.

TABLE VIII. MINING POOLS FOR MAJOR CRYPTOCURRENCIES

Pool Name	**Coin**	**Reward type**	**Hash power**
BTC.com	BTC	FPPS	8.13 EH/s
	BCH	FPPS	207.97 PH/s
Antpool	BTC	PPLNS,PPS	5.06 EH/s
	BCH	PPLNS,PPS	149.48 PH/s
	ETH	PPLNS,PPS	323 GH/s
	LTC	PPLNS,PPS	21.2 TH/s
	ETC	PPLNS,PPS	6.26 GH/s
	ZEC	PPLNS,PPS	258 MSol/s
	DASH	PPLNS,PPS	138 TH/S
	SIA	PPLNS,PPS	816.92 TH/s
Slush	BTC	Score	3.27 EH/s
ViaBTC	BTC	PPLNS,PPS	2.87 EH/s
	BCH	PPLNS,PPS	84.49 PH/s
	ETH	PPLNS,PPS	878 GH/s
	LTC	PPLNS,PPS	26.52 TH/s
	ETC	PPLNS,PPS	6.032 GH/s
	ZEC	PPLNS,PPS	89.63 MSol/s
	DASH	PPLNS,PPS	117.36 TH/s
Miningpoolhub	ETH	PPLNS	8.33 TH/s
	LTC	PPLNS	277 GH/s
	ETC	PPLNS	1.494 TH/s
	ZEC	PPLNS	29.378 MH/s
	DASH	PPLNS	27.69 TH/s
	XMR	PPLNS	5.74 MH/s
	DGB	PPLNS	11.85 TH/s
Bitcoin.com	BTC	PPS	892.16 PH/s
	BCH	PPS	191.72 PH/s
Nanopool	ETH	PPLNS	20.96 TH/s
	ETC	PPLNS	1.65 TH/s
	ZEC	PPLNS	53.57 MSol/s
	GRIN	PPLNS	17.1 Kgp/s
	XMR	PPLNS	57.488 MH/s
Litecoinpool.org	LTC	PPS	33.49 TH/s
Ethermine	ETH	PPLNS	41.4 TH/s
	ETC	PPLNS	3.5 TH/s
	ZEC	PPLNS	396.9 MSol/s
f2pool	BTC	PPS	4.46 EH/s
	ETH	PPS	17.7 TH/s
	LTC	PPS	40.94 TH/s
	ETC	PPS	127.37 GH/s
	ZEC	PPS	379.73 MSol/s
	DASH	PPS	74.44 TH/s
	SIA	PPS	3.59 TH/s
	XMR	PPS	54.50 MH/s
Multipool	BTC	Exponential	0.66 PH/s
	BCH	PPLNS	3.658 PH/s
	LTC	PPLNS	122.22 GH/s
	DGB	Proportional	3.836 PH/s
Minergate	BTG	PPLNS,PPS	2.006 KSol/s
	ETH	PPLNS	14.68 GH/s
	ETC	PPLNS	4.374 GH/s
	ZEC	PPLNS	16.12 KSol/s
	XMR	PPLNS,PPS	3.776 MH/s
	BCN	PPLNS,PPS	1.641 MH/s
Suprnova	BTG	Proportional	0.25 MSol/s
	LTC	Proportional	24.09 GH/
	ZEC	Proportional	19.03 KSol/s
	DGB	Proportional	9.45 TH/s
	DASH	Proportional	9.26 TH/s
Coinotron	ETH	PPLNS,RBPPS	806.6 GH/s
	LTC	PPLNS,PPS	37.8 GH/s
	ETC	PPLNS,RBPPS	53.8 GH/s
	ZEC	PPLNS,PPS	491.5 MSol/s
	BTG	PPLNS,PPS	332.6 KH/s
	DASH	PPLNS,PPS	2.4 TH/s