JOURNAL OF CYBERSECURITY

Journal of Cybersecurity, 2022, 1–15
https://doi.org/10.1093/cybsec/tyab027
Research paper

Research paper

Diversification across mining pools: optimal mining strategies under PoW

Panagiotis Chatzigiannis, Foteini Baldimtsi, Igor Griva and Jiasun Li

Department of Computer Science George Mason University, 4400 University Drive MSN 4A5 Fairfax, VA 22030, USA

*Correspondence address. Department of Computer Science George Mason University 4400 University Drive MSN 4A5 Fairfax, VA 22030 USA - phone: +1-703-993-1530; E-mail: pchatzig@gmu.edu

Received 3 October 2019; revised 29 June 2021; accepted 21 November 2021

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. Our underlying techniques are drawn from both the areas of financial economy and computer science since we use computer science-based approaches (i.e. optimization techniques) to experimentally prove how parties (and in particular miners) interact with cryptocurrencies in a way of increasing their Sharpe ratio. To showcase our model, 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).

Key words: mining pools, proof-of-work, risk-sharing, cryptocurrency

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) competes 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 became 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 became 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 the "Reward methods in mining pools" section). At the same time different pools control a different ratio of the overall hash rate consumed by a cryptocurrency and larger pools (in terms of hash rate) 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

1 We measure a miner's "risk" by the variance of rewards over a given time [2].

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 the "Active Miner's Problem" section, we provide three different versions of our model. The first one, inspired by [3], concerns a miner who 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 who 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 [4] to multiple mining pools, rather than multiple assets.

In the "Implementation of Our Model" section, we present an implementation of our model. We develop a Python tool that uses the constrained optimization by linear approximation (COBYLA) [5] 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 (see the "Evaluation and Simulated Results" section). We start by considering a Bitcoin miner who starts mining passively on a single chosen pool for 4 months and computes his earnings on a daily basis.

Then, we consider a miner with the same hash power who using our tool, would "actively" diversify every 3 days over three 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 [6]) by 260% compared with the "passive" miner. In our experiments, for the "passive" miner we picked a large and reputable pool (SlushPool) 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 that on inputting a miner's hashing power suggest which cryptocurrency is currently most profitable. For instance, tools like MultiPoolMiner [7], SmartMine [8], or MinerGate [9] 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 that 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 toward 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.* [10] who provided a top-down (from pool's point of view) analysis, i.e. focused on optimal pool operation strategies toward maximizing the pool's expected utility. Cong *et al.* [3] 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 [11], 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 [12] 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). Lewenberg *et al.* [13] showed how network delays incentivize miners to switch among pools in order to optimize their payoffs due to the nonlinearity introduced. Liu *et al.* [14] 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 [13, 14] as complementary to our work.

Mining Background

As of today, the majority of blockchain-based cryptocurrencies use PoW for maintaining their ledger. Miners listen the network for (i) pending transactions and (ii) 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 toward 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.

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 later. (In Table 1: we provide a referece for notation used throughout our work).

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 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 [15]. 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 3, 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 [16] and [15] 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.

Table 1: Notation

Total number of pools	M
Hash rate 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 hash rate	Λ_c
Block time and block reward of cryptocurrency c	D_c, R_c
Total number of cryptocurrencies	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

Mining pools offering multiple reward methods

As mentioned earlier, some mining pools offer multiple reward systems (i.e. the Coinotron pool [17] 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 hash rate 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 hash rate of λ_m will get a reward based on the percent of the hash rate he contributed with respect to the total hash rate 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 hash rate of the pool Λ_m, and then allocates the remainder of the rewards to the PPS bucket. By this strategy, $R_{\lambda_m} = R\frac{\lambda_m}{\Lambda_m}$, which effectively results in the same paid amount as in Strategy 1.

Strategy 3: First, replenish the PPS "bucket" based on the total amount $\tilde{R}$ that was paid off to the PPS miners since the last block was found, and 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 nontrivial 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.

Active Miner's Problem

We study the problem faced by a miner, who given a set of C different cryptocurrencies and M mining pools for each currency, where each pool m has a total hash rate $\Lambda_{m,c}$ and fee $f_{m,c}$, maximizes his expected utility. In the rest of the section, we first characterize the miner's payoff, then specify his optimization problem, and finally derive a tractable version under constant-absolute-risk-aversion (CARA) utilities for later numerical analysis. We also discuss how miners with more general utilities may make use of our model.

Miner's payoff

A miner owns mining hardware with PoW hashing power λ_{A} and is mining on C different cryptocurrencies, each with total hash rate Λ_c. The miner can distribute λ_{A} among different cryptocurrencies and different mining pools that offer different fee structures (while possibly keeping a portion of his power for zero-fee solo mining). Therefore the miner's payoff P is given by the following equation.

$$\tilde{\mathsf{P}} = \sum_{c\in\mathsf{C}} \left(\sum_{m=1}^{M_c} \underbrace{\frac{\lambda_{m,c}}{\lambda_{m,c}+\Lambda_{m,c}}}_{\text{within-pool hashrate share}} \times \underbrace{(1-f_{m,c})(R_c/D_c + \mathsf{tx}_c \mathsf{tx}_{m,c})\tilde{N}_{pool,m,c}}_{\text{Pool's total reward to miners}} + \underbrace{(R_c/D_c + \mathsf{tx}_c \mathsf{tx}_{m,c})\tilde{N}_{solo,c}}_{\text{solo reward}} + \underbrace{(R_c/D_c + \mathsf{tx}_c \mathsf{tx}_{m,c})\frac{\lambda_{\mathsf{PPS},c}(1-f_{\mathsf{PPS},c})}{\Lambda_c}}_{\text{PPS pools reward}} \right) \quad (1)$$

The above payoff for each cryptocurrency c includes the following terms: (i) the weighted sum of each pool m's reward, with the weights being the miner's hash rate percentage in the pool, where $\lambda_{m,c}$ denotes the mining power allocated to pool m in cryptocurrency c, $\Lambda_{m,c}$ and R_c denote the total hashing rate of pool m and the block reward of cryptocurrency c, respectively, tx_c denotes the average transaction fee for cryptocurrency c observed during a recent period of time, $\mathsf{tx}_{m,c} = 1$ if the pool pays transaction fees to the miner, else we set $\mathsf{tx}_{m,c} = 0$, $f_{m,c}$ the pool fee percentage (which is subtracted from the miner's payoff), and $\tilde{N}_{pool,m,c} \sim Poisson(\frac{\lambda_{m,c}+\Lambda_{m,c}}{\Lambda_c})$ denotes the (random) number of blocks pool m for cryptocurrency c mines. Note as each cryptocurrency c has its own block reward R_c[2] and its own average block time D_c, we consider the reward over time ratio $\frac{R_c}{D_c}$, as it effectively normalizes R_c over different cryptocurrencies. (ii) The reward from solo mining, where $\tilde{N}_{solo} \sim Poisson(\frac{\lambda_{solo}}{\Lambda})$ denotes the (random) number of blocks a solo miner mines. (iii) The reward from a PPS pool in cryptocurrency c, where $\Lambda_{m,c}$ denotes the total hashing rate of that cryptocurrency.

Miner's allocation problem

Given the expression of miner's payoff in Equation (1), a miner with a utility function $u(\cdot)$ and initial wealth W_0 would solve the following problem. Choose vector $\{\lambda_{m,c}\}$ to maximize

$$E[u(W_0 + \tilde{\mathsf{P}})], \quad (2)$$

where $\tilde{\mathsf{P}}$ is given by Equation (1), subject to

$$\sum_c \sum_{m=1}^{M} \lambda_{m,c} + \lambda_{PPS,c} + \lambda_{solo,c} \le \lambda_{\mathsf{A}} \text{ and } \lambda_{m,c} \ge 0, \forall m \in M.$$

We can further generalize to allow a miner to distribute his power across C different cryptocurrencies, and across A different PoW algorithms, which of course assumes that the miner owns CPU/GPU mining hardware, since ASICs are restricted to specific PoW algorithms. We discuss this case in more detail in Appendix 1.

Analytical forms under CARA utility

For general utility functions, Equation (2) does not have an analytical expression, rendering further analysis difficult. One exception is with CARA utilities, where Equation (2) can be obtained as the moment-generating function of the weighted sum of independent Poisson random variables.

Specifically, suppose the miner has CARA utility $u(\cdot)$ as

$$u(W_0 + \mathsf{P}) = -e^{-\rho(W_0+\mathsf{P})}, \quad (3)$$

where the CARA parameter ρ quantifies how risk averse a miner is (e.g. $\rho = 0$ means that the miner is risk neutral). Then 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 $e^{\lambda(e^w-1)}$. Therefore, if we plug Equation (1) into Equation (3), the miner's objective function can be expressed as certainty equivalents in analytical forms. The most general one will be given in Equation (6) later. Before presenting the most general case, we also highlight a few special cases that will be further implemented in the "Implementation of Our Model" section.

Mining on a single cryptocurrency without PPS pools

If a miner only mines in one crytocurrency and only allocates hash rates to mining pools offering a PPLNS reward method, the miner's certainty equivalent is given by Equation 4, and the miner looks for a vector $\{\lambda_m\}_{m=1}^M$ to maximize it.[3] This vector expresses an allocation of the miner's mining power λ_{A} over M different pools, where λ_m denotes the mining power allocated to pool m

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

under constraints

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

By solving this optimization problem, we are given the optimal distribution of the miner's total hash power λ_{A} to M pools. Note that the second term of Equation (4) expresses the leftover hash power 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.

Allowing selection of PPS pools

Equation (4) restricts the miners to choose between pools that only offer the PPLNS reward method. We now allow miners to choose mining pools also offering PPS reward systems in our model. In this case, a rational miner will choose to add only the PPS pool that offers

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.

3 This special case was first studied in [3].

the smaller fee, and disregard pools with higher PPS fees. Equation (4) is then transformed as follows (changes denoted in blue color):

$$\sum_{m=1}^{M}(\lambda_m+\Lambda_m)(1-e^{-\rho R(1-f_m)\frac{\lambda_m}{\lambda_m+\Lambda_m}})+\left(\lambda_{\mathsf{A}}-\lambda_{\mathsf{PPS}}-\sum_{m=1}^{M}\lambda_m\right)\times(1-e^{-\rho R})+\lambda_{\mathsf{PPS}}(1-f_{\mathsf{PPS}})\rho R \quad (5)$$

under constraints

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

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 4 takes the form

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

In addition, a miner's hash rate λ_m allocated to pool m that mines cryptocurrency c should be normalized to each cryptocurrency's total hash rate Λ_c. We also now include the transaction fees into the block reward. So, in this (more general) case, Equation 4 takes the following form, under the new constraints outlined earlier:

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

where $\lambda_{0,c}$ denotes solo mining for cryptocurrency c, and the first constraint of Equation 4 is more precisely expressed as

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

The above constraint shows that the miner could choose to diversify his solo mining (which was initially expressed by the second term in Equation 4) over multiple currencies as well, in a similar fashion as the miner would do by diversifying across multiple mining pools.

A discussion over more general utility functions

From a theoretical perspective, the miner's problem in the "Miner's allocation problem" section is well specified for any increasing and concave utility function $u(\cdot)$. Our choice of CARA utilities here is solely for computational efficiency in later numerical solutions in the "Implementation of Our Model" section. Otherwise, without an analytical expression for the miner's objective function, the objective has to be calculated by first simulating random variables and then taking numerical integration. Furthermore, such procedures need to be repeated every time we iterate through a candidate allocation, dramatically increasing computation complexity.

That said, we are cognizant that a miner's utility may not necessarily conform to constant absolute risk aversion. For example, a miner may have a constant-relative-risk-aversion (CRRA) utility. We argue that such miners can nevertheless find our tool useful for finding approximate solutions to their problems. Specifically, if a miner has initial wealth W_0 and a CRRA utility with parameter η, i.e. $u(W_0+\mathsf{P})=\frac{(W_0+\mathsf{P})^{1-\eta}-1}{1-\eta}$, then since $E[\mathsf{P}]$ is typically small relative to W_0, one can approximate the miner's "pseudo" absolute risk aversion as $\rho=W_0^{-1}\eta$ and plug it into Equation (6). One can then efficiently solve the problem in the "Miner's allocation problem" section with the "pseudo" objective, and the resulting allocations would give an approximate solution to the miner's original problem. Over time, as the miner's wealth gradually changes to some W_1 (so that the utility becomes $\frac{(W_1+\mathsf{P})^{1-\eta}-1}{1-\eta}$), the miner can recalibrate their "pseudo" absolute risk aversion as $\rho=W_1^{-1}\eta$ and readjust their hash rate allocation. However, such readjustment only needs to be periodically, and the numerical solution is computationally efficient.

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).[4] Our tool covers all the cases discussed in the "Active Miner's Problem" section. In Appendix 2, we provide details for choosing the right optimization method of our tool.

Tool description and instantiation assumptions. The basic single cryptocurrency version of our tool, given as inputting the miner's hashing power λ_{A}, coin's exchange rate E, chosen pool data $[\Lambda_i, f_i]_{i=1}^{M}$ (pool total hash rate and fee, respectively) and risk aversion ρ, outputs the optimal distribution of the miner's hash power over these pools plus a "solo-mining" remainder. Some instantiations of our tool are outlined in the "Single cryptocurrency" section as examples for a typical value range of ρ. Our tool can also provide the optimal distribution for the "multicryptocurrency, single PoW algorithm" (see the "Analytical forms under CARA utility" section) and "multicryptocurrency, multi-PoW algorithm" (Appendix 1) cases, and we also outline an extended instantiation in the "Multiple cryptocurrencies" section.

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 [18]. Multi-pool mining in a round-robin fashion is not efficient for the miner as well, as this would result in a decrease in his overall reward, given the nature of reward schemes that prevent pool "hopping." We encour-

4 The latest version of our tool can be accessed at http://smart-miner.com.

Table 2: Pool parameters

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

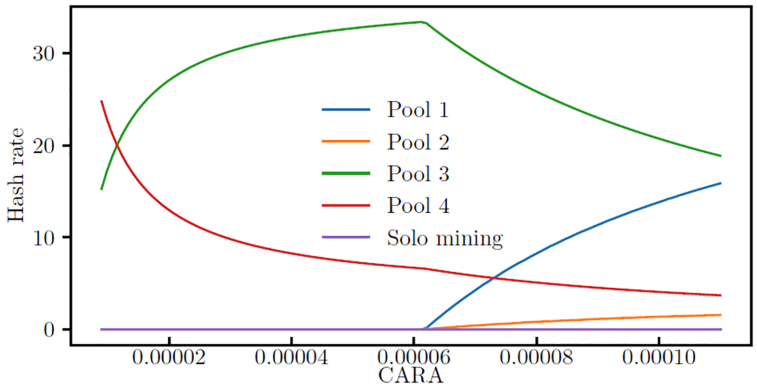


Figure 1: Single cryptocurrency diversification.

age ASIC manufacturers and mining application developers to enable user-specified multipool mining in future releases, for the benefit of the miners and the whole community.

We assume that the miner possesses an average wealth of $\mathcal{W}$ = \$100k, while having typical values for the constant relative risk aversion CRRA metric between 1 and 10 [19]. Given that CARA = CRRA/$\mathcal{W}$, we take as typical values for CARA ρ between 10^{-5} and 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).

Single cryptocurrency

Evaluation I

We instantiate the first experiment of our tool by using the following parameters: a miner with total hash power $\lambda_A = 40$ hashes/s, wishing to mine on a single cryptocurrency with block reward R = \$50 000, having picked four mining pools with parameters shown in Table 2. 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 in Fig. 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 hash power 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 that is 10 times larger than Pool 2. Note that for simplicity, we do not take any transaction fees kept by pools into account; however, as shown in the previous section, our evaluation would produce equivalent results.

Evaluation II

We then pick some actual Bitcoin pools: SlushPool, ViaBTC, and KanoPool. These pools, as indicated by their parameters shown in Table 3 (values as of September 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 5, 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/s (roughly about 100 units of Antminer S15 ASICs), and the Bitcoin total block reward R = \$129 502.[5] The pool parameters are shown in Table 3 and the resulting diversification graph in Fig. 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).

Table 3: Pool parameters

SlushPool	Λ_1	7380 PH/s
	f_1	2%
ViaBTC	Λ_2	6210 PH/s
	f_2	2%
KanoPool	Λ_3	194 PH/s
	f_3	0.9%

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 3 and typical values for ρ. In Fig. 3, we show how a PPS pool would affect a miner's diversification for nontypical large values of ρ, using the parameters in Table 4. 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.

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 Fig. 4, we show a small-scale (or "home") Bitcoin miner as an example (λ_A = 125 TH/s). Given his relatively small hash power, 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.

5 Parameters as of 20 September 2019, retrieved from https://btc.com/ and https://bitinfocharts.com/.

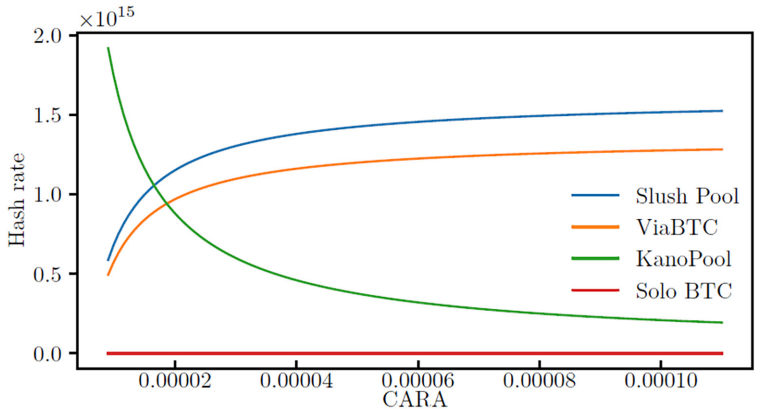


Figure 2: Large-scale miner on Bitcoin pools.

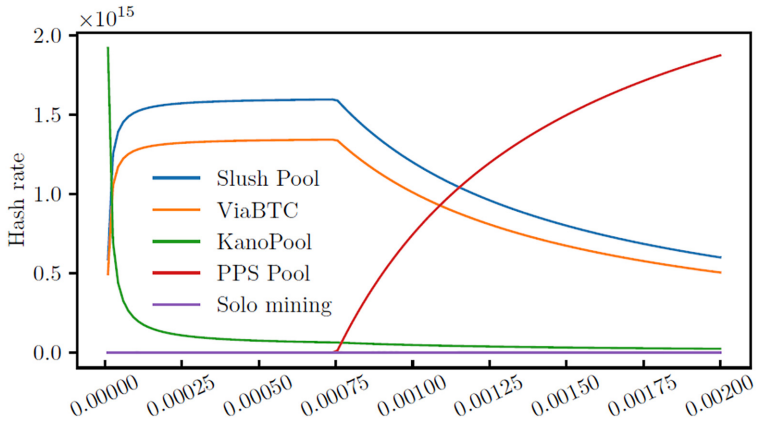


Figure 3: Single currency with PPS pool and large values of ρ.

Table 4: Bitcoin pool parameters including PPS pool

SlushPool	Λ_1	7380 PH/s
	f_1	2%
ViaBTC	Λ_2	6210 PH/s
	f_2	2%
KanoPool	Λ_3	194 PH/s
	f_3	0.9%
PPS pool	f_4	4%

Multiple cryptocurrencies

We now consider a miner who diversifies over different cryptocurrencies. For simplicity, we just switch[6] the currency in the 2nd pool (ViaBTC) from Bitcoin to Bitcoin Cash. The pool parameters are shown in Table 5: $\Lambda_{\mathsf{BTC}} = 91.26$ EH/s, $\Lambda_{\mathsf{BCH}} = 2.5$ EH/s and $R_{\mathsf{BCH}} = \$3967$.[5] The resulting graph in Fig. 5 shows 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, allocating some of his power to the SlushPool, even though it has the same fee and the pool might not be as profitable as the Bitcoin Cash pool.

6 Many pools as shown in Appendix 3 host pool mining services for multiple cryptocurrencies.

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 Fig 6a and 6b, 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 = 0.028, 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.

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 [20], as shown in Table 6. First, we consider a Bitcoin miner owning some hashing power $\lambda_A = 1200$ TH/s, who would start mining passively on a single chosen pool on 1 February 2018 for $\Delta = 4$ months. Then, we consider

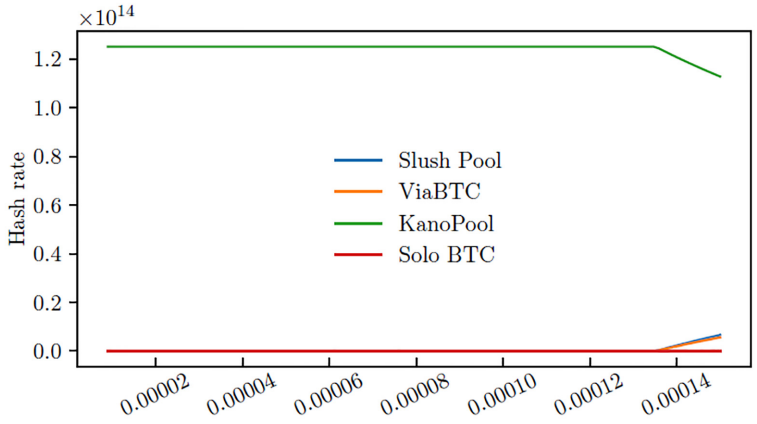


Figure 4: Small-scale miner diversifying over Bitcoin pools, parameters in Table 3.

Table 5: Pool parameters

SlushPool	Λ_1	7380 PH/s
	f_1	2%
ViaBTC	Λ_2	157 PH/s
(Bitcoin Cash)	f_2	2%
KanoPool	Λ_3	194 PH/s
	f_3	0.9%

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 hash power accordingly. In Table 7, 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 the "Implementation of Our Model" section, we pick the mean value for $\rho = 0.00005$. Note that as we mentioned in the "Introduction" 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[7] 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" [21, 22]).

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

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 hash power (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 hash rate 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 hash power 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 hash rates (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 discussed in the "Analytical forms under CARA utility" section 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 (6) 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).

Main evaluation results

In Fig. 7, we show the earnings over time for both of the miners for comparison (in Fig. 7 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 hash power Λ. However his total accumulated reward would be $\mathsf{P}_A = \$101\,221$ for the active miner, compared with $\mathsf{P}_P = \$97\,101$, which eventually leads to a Sharpe ratio $\mathsf{S}_A = 0.156$ compared with $\mathsf{S}_P = 0.060$. Note that since we assumed both miners' hash power 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 $\frac{\lambda_A}{\Lambda}$. We

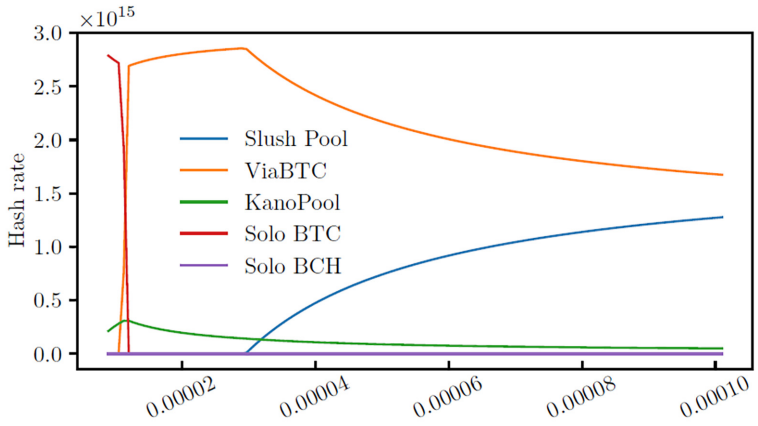


Figure 5: Large-scale miner on SHA-256 pools.

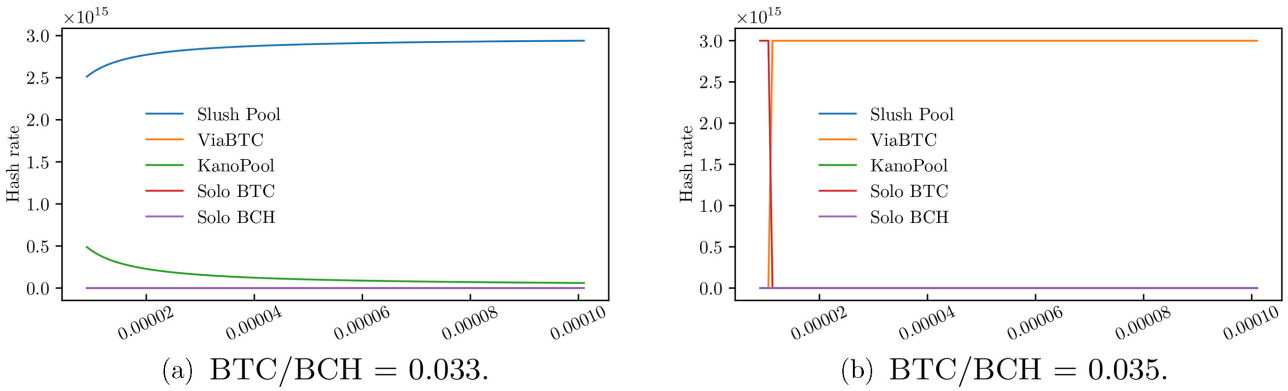


Figure 6: Large-scale miner on SHA-256 pools, parameters in Table 5.

Table 6: 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 7: Simulation parameters

Miners' hash power λ_A	1200 TH/s
Diversification interval t	3 days
Fee $f_{\mathsf{SlushPool}}$	2%
Fee f_{ViaBTC}	2%
Fee f_{DPOOL}	1%
CARA ρ	0.00005

also include an equivalent analysis using the same pools over a 1-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 [23]. We used Bitcoin for our simulation, where D is relatively high (roughly 10 min); this resulted in high reward variance, especially when using smaller pools as shown in Figs 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.

Analysis for a large Δ

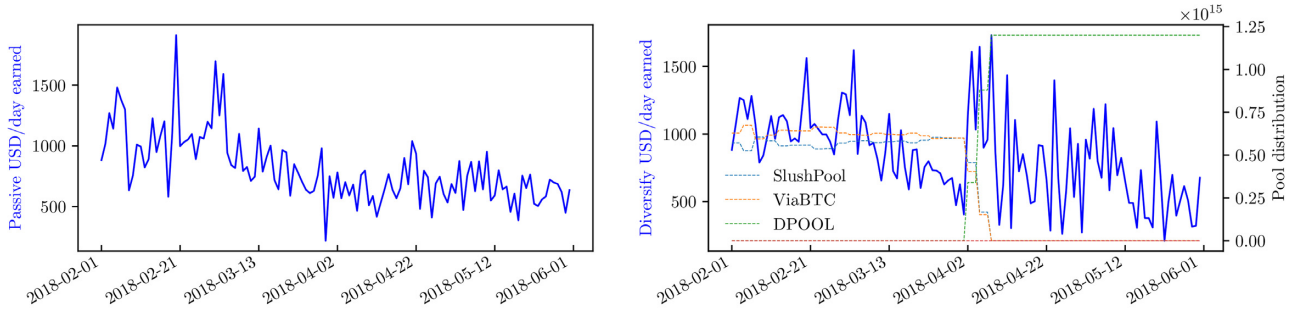
In Fig. 8, we repeat the simulation discussed in the "Evaluation and Simulated Results" section over the period of $\Delta = 1$ year (1 January 2018–31 December 2018). The decline of miner rewards due to the increasing difficulty can be observed more clearly. In such a case, a miner would most likely reinvest his earnings on mining hardware to keep $\frac{\lambda_A}{\Lambda}$ 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 $\mathsf{P_A} = \$233\,293$ and $\mathsf{S_A} = 0.064$ vs $\mathsf{P_P} = \$224\,293$ and $\mathsf{S_P} = 0.023$, which are consistent with the results derived from the 4-month simulation.

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 (1 January 2018–30 June 2018) for $\Delta = 6$ months, thus diversifying over the three largest PPLNS pools during that period. Again, we observe an improvement in our metrics, $\mathsf{P_A} = \$172\,719$ and $\mathsf{S_A} = 0.041$ vs $\mathsf{P_P} = \$172\,092$ and $\mathsf{S_P} = 0.032$. Detailed analysis is shown in Fig. 9.

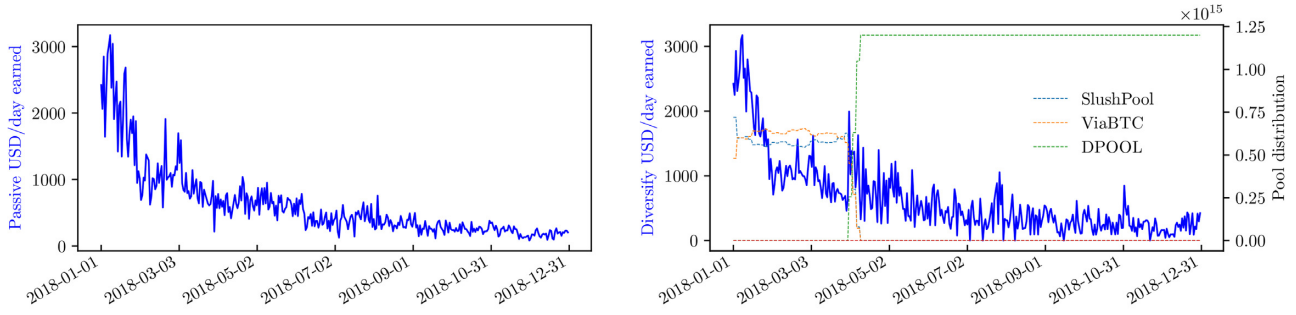
Analysis for different values of ρ

We now repeat our retroactive analysis discussed in the "Evaluation and Simulated Results" section by setting $\rho = 0.0001$, which is at the upper bound of our considered typical values. By comparing



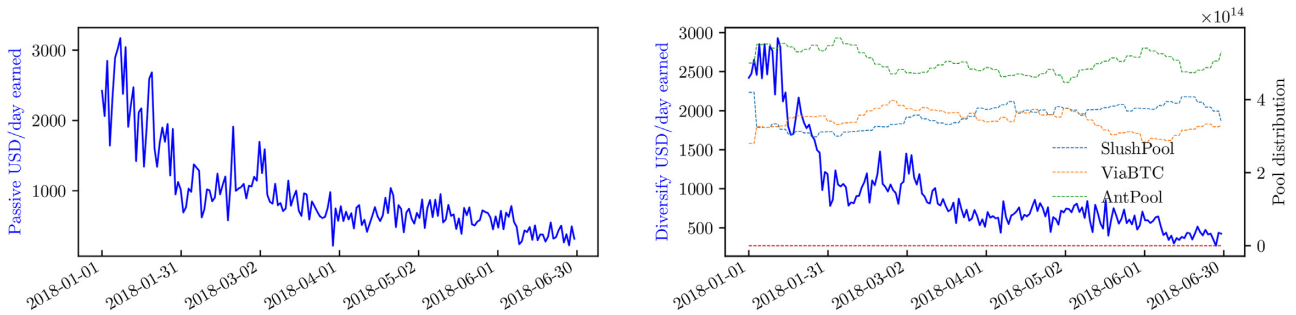
(a) Passive miner on Slush pool. (b) Active miner on 3 pools.

Figure 7: Active miner on SlushPool (up) and on three pools (down). Hash power distribution over 4-month data.



(a) Passive miner on Slush pool. (b) Active miner on 3 pools.

Figure 8: Active miner on SlushPool (up) and on three pools (down). Hash power distribution over 1-year data.



(a) Passive miner on Slush pool. (b) Active miner on 3 pools.

Figure 9: Active miner on SlushPool (up) and on three pools (down). Hash power distribution over 6-month data and different pool set.

Fig. 10 with Fig. 7, 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 $\mathsf{P}_\mathsf{A} = \$99\,082$, and the Sharpe ratio would be $\mathsf{S}_\mathsf{A} = 0.104$.

By further experimenting with a broader value range for ρ, we derive Fig. 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 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

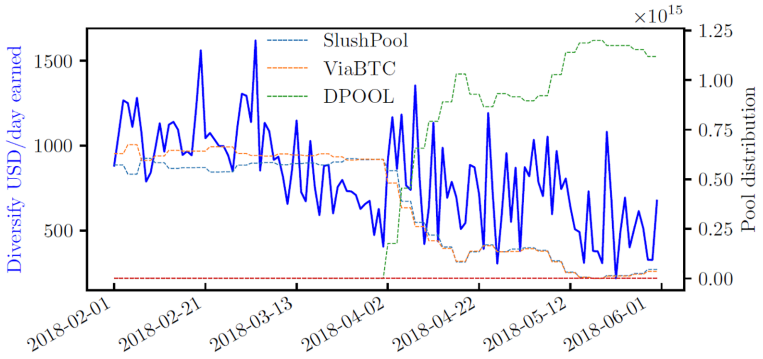


Figure 10: Active miner diversifying over Bitcoin pools with increased $\rho = 0.0001$.

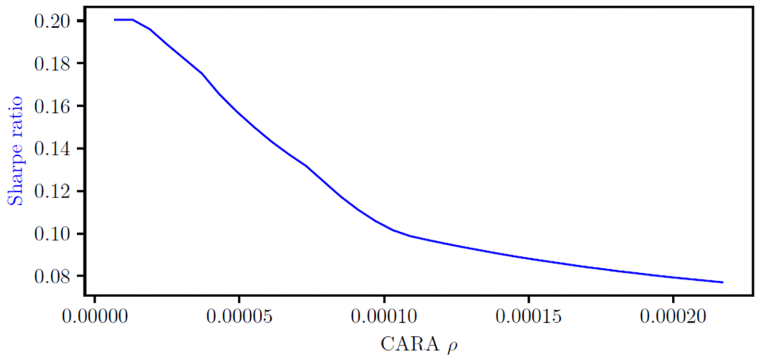


Figure 11: Sharpe & rewards vs ρ, SlushPool—ViaBTC—DPool.

shown in Fig. 12, we observe that the chosen ρ has eventually no effect on the Sharpe ratio.

Analysis for different values of miner power

By repeating our main evaluation for several different values of miner's computational power, we derive Fig. 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 with 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 Fig. 14, for a relatively small miner, we now observe no significant correlation with miner hash rate and the resulting Sharpe ratio.

Analysis for different diversification intervals

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

Analysis for different number of available pools

In Fig. 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.

Analysis for different cryptocurrencies

As discussed in the "Analytical forms under CARA utility" section, our model also considers miners who diversify over multiple cryptocurrencies that use the same PoW algorithm. Extending our evaluation results to such a case is relatively straightforward (assuming the equivalent data shown in Table 6 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 hash rate 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 that also employ different PoW algorithms as discussed in Appendix 1 is hard to execute in practice, since as discussed it excludes ASIC hardware, while it requires fine-tuning process priorities in CPUs and GPUs.

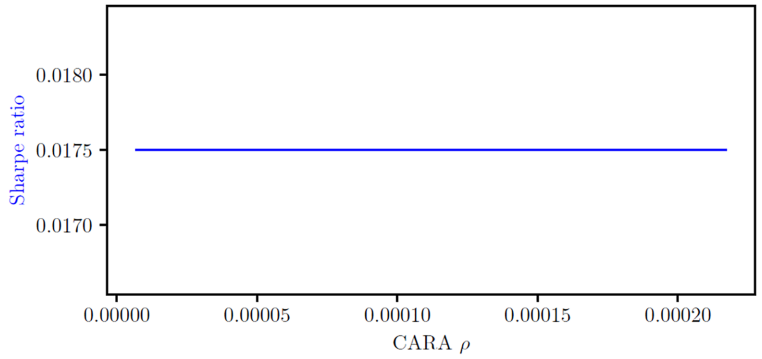


Figure 12: Sharpe & rewards vs ρ, SlushPool—ViaBTC—AntPool.

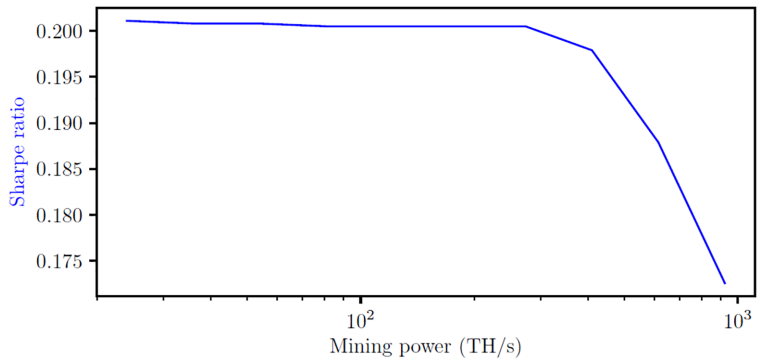


Figure 13: Sharpe & rewards vs mining power, SlushPool—ViaBTC—DPool.

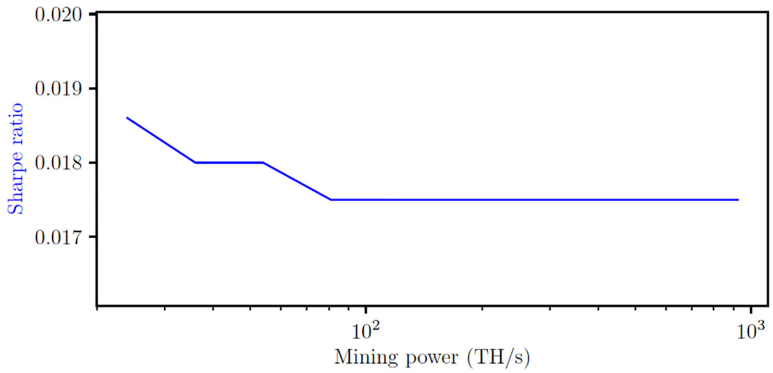


Figure 14: Active/passive Sharpe ratios vs mining power, SlushPool—ViaBTC—AntPool.

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 the "Implementation of Our Model" section) 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 led to blockchain centralization (e.g. in Bitcoin at the time of writing, over 50% of mining is done by four mining pools). Lack of decentralization can lead to various types of attacks, including double-spending, reversing confirmations of previous transactions or transaction censoring [24–27]. 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 [3, 23, 27], while a number of solutions have been proposed spanning

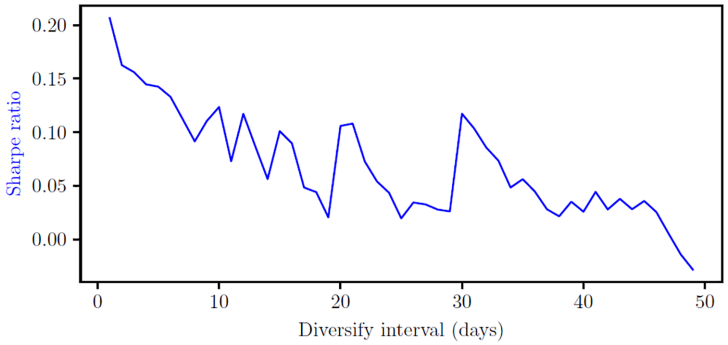


Figure 15: Sharpe & rewards vs diversification interval *t*.

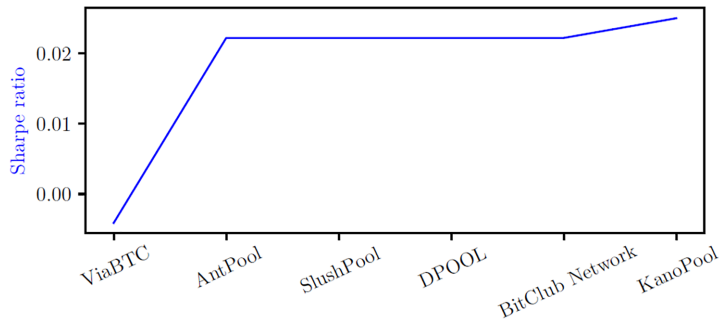


Figure 16: Sharpe & rewards vs type and order pools.

from decentralized mining pools (P2Pool [28] for Bitcoin) to alternative, non-outsourceable PoW mechanisms [29]. We believe that tools like the one we present here are positive steps toward 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 rational miners would behave.

Funding

This work was supported by the George Mason Multidisciplinary Research (MDR) Initiative and DHS/CINA (Department of Homeland Security / Criminal Investigations and Network Analysis) Award #205187.

Conflict of Interest

The authors reported no potential conflict of interest.

References

1. List of Bitcoin mining ASICs. [Online]. https://en.bitcoin.it/wiki/List_of_Bitcoin_mining_ASICs (19 February 2019, date last accessed).
2. Ingersoll JE. *Theory of Financial Decision Making*, Vol. 3. Totowa, NJ: Rowman & Littlefield, 1987.
3. Cong LW, He Z, Li J. Decentralized mining in centralized pools. 2018. [Online]. https://ssrn.com/abstract=3143724. revision, 10 Dec 2019.
4. Markowitz H. Portfolio selection. *J Finance* 1952;7:77–91.
5. Powell MJD. A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gomez S, Hennart JP (eds). *Advances in Optimization and Numerical Analysis. Mathematics and Its Applications*. Dordrecht: Springer Netherlands, 1994, 51–67. [Online]. https://doi.org/10.1007/978-94-015-8330-5_4.
6. Sharpe WF. Mutual fund performance. *J Bus* 1966;**39**:119–38. http://www.jstor.org/stable/2351741.
7. MultiPoolMiner. [Online]. https://multipoolminer.io/ (15 January 2019, date last accessed).
8. SmartMine. [Online]. https://www.smartmine.org/ (15 January 2019, date last accessed).
9. MinerGate. [Online]. https://minergate.com/ (15 January 2019, date last accessed).
10. Fisch B, Pass R, Shelat A. Socially optimal mining pools. In: Devanur NR, Lu P (eds). *Web and Internet Economics*. Cham: Springer International Publishing, 2017, 205–18.
11. Bissias G, Levine BN, Thibodeau D. 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. p. 155–72. https://doi.org/10.1007/978-3-030-00305-0_12.
12. Spiegelman A, Keidar I, Tennenholtz M. Game of coins. [Online]. https://ieeexplore.ieee.org/document/9546433.
13. Lewenberg Y, Bachrach Y, Sompolinsky Y. *et al.*. 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, p. 919–27. [Online]. http://dl.acm.org/citation.cfm?id=2772879.2773270.
14. Liu X, Wang W, Niyato D. *et al.*. Evolutionary game for mining pool selection in blockchain networks. *IEEE Wirel Commun Lett* 2018;7:760–3.
15. Rosenfeld M. Analysis of bitcoin pooled mining reward systems. [Online]. http://arxiv.org/abs/1112.4980.
16. Comparison of mining pools. [Online]. https://en.bitcoin.it/wiki/Comparison_of_mining_pools (15 January 2019, date last accessed).
17. Coinotron. [Online]. https://coinotron.com (15 January 2019, date last accessed).

18. Kolivas C. [Online]. https://github.com/ckolivas/cgminer (15 January 2019, date last accessed).
19. Mas-Colell A, Whinston M, Green J. *Microeconomic Theory*. Oxford: Oxford University Press, 1995.
20. Smartbit Bitcoin Block Explorer. [Online]. https://www.smartbit.com.au/ (15 January 2019, date last accessed).
21. Ross SA. Adding risks: Samuelson's fallacy of large numbers revisited. *J Financ Quant Anal* 1999;**34**:323–39.
22. Samuelson PA. Risk and uncertainty: a fallacy of large numbers. *Scientia* 1963;**98**:108–13.
23. Gencer AE, Basu S, Eyal I. *et al.*. Decentralization in Bitcoin and Ethereum networks. 2018. [Online]. https://link.springer.com/chapter/10.1007/978-3-662-58387-6_24.
24. Eyal I. The miner's dilemma. In: *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 89–103.
25. Eyal I, Sirer EG. Majority is not enough: Bitcoin mining is vulnerable, vol. 8437. In: Christin N, Safavi-Naini R, (eds). *FC 2014, ser. LNCS*, Vol. 8437. Springer, Heidelberg, 2014, 436–54.
26. Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in Bitcoin, vol. 9603. In: Grossklags J, Preneel B (eds). *FC 2016, ser. LNCS*. Springer, Heidelberg, 2016, 515–32.
27. Sompolinsky Y, Zohar A. Bitcoin's underlying incentives. *Queue* 2017;**15**:29–52. [Online]. http://doi.acm.org/10.1145/3155112.3168362.
28. P2pool. [Online]. http://p2pool.in/ (15 January 2019, date last accessed).
29. Miller A, Kosba AE, Katz J. *et al.*. Nonoutsourceable scratch-off puzzles to discourage Bitcoin mining coalitions. In: Ray I, Li N, Kruegel C (eds). *ACM CCS 15*. ACM Press, 2015, 680–91.
30. Kraft D. *A Software Package for Sequential Quadratic Programming, ser*. Tech. Rep. DFVLR-FB 88-28. Köln, Germany: DLR German Aerospace Center—Institute for Flight Mechanics, 1988.

Appendix 1: Mining Across Cryptocurrencies with Different PoW Algorithms

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" hash rate 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

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

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 that uses PoW mining algorithm α_1, and his maximum hash rate would be λ_{α_1}. Alternatively, he could mine exclusively coin c_2 that 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.[8]

Appendix 2: 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, 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 [30]. 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/s 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 COBYLA [5]. 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 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 solution 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 not be 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.

Appendix 3: Mining Pools for Major Cryptocurrencies

In Table C1, we summarize a list of mining pools for major cryptocurrencies and the reward type each offers, as of 24 September 2019. We have already discussed PPS, PPLNS and proportional in the "Reward methods in mining pools" section. 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.

8 In our assumption, we do not take a "dual mining" GPU setup into account.

Table C1: Mining pools for major cryptocurrencies

Pool name	Coin	Reward type	Hash power
BTC.com	BTC	FPPS	13.88 EH/s
	BCH	FPPS	211.00 PH/s
AntPool	BTC	PPLNS, PPS	10.07 EH/s
	BCH	PPLNS, PPS	160 PH/s
	ETH	PPLNS, PPS	497 GH/s
	LTC	PPLNS, PPS	29.9 TH/s
	ETC	PPLNS, PPS	76.3 GH/s
	ZEC	PPLNS, PPS	802 MSol/s
	DASH	PPLNS, PPS	949 TH/S
	SIA	PPLNS, PPS	206 TH/s
ViaBTC	BTC	PPLNS, PPS	6.78 EH/s
	BCH	PPLNS, PPS	155.90 PH/s
	ETH	PPLNS, PPS	219.79 GH/s
	LTC	PPLNS, PPS	24.76 TH/s
	ETC	PPLNS, PPS	13.51 GH/s
	ZEC	PPLNS, PPS	230.77 MSol/s
	DASH	PPLNS, PPS	16.81 TH/s
Miningpoolhub	ETH	PPLNS	8.22 TH/s
	LTC	PPLNS	257.02 GH/s
	ETC	PPLNS	1.67 TH/s
	ZEC	PPLNS	49.68 MH/s
	DASH	PPLNS	1.65 TH/s
	XMR	PPLNS	4.62 MH/s
	DGB	PPLNS	777.1 TH/s
Bitcoin.com	BTC	PPS	253.24 PH/s
	BCH	PPS	215.34 PH/s
Nanopool	ETH	PPLNS	22.28 TH/s
	ETC	PPLNS	1.96 TH/s
	ZEC	PPLNS	40.74 MSol/s
	GRIN	PPLNS	39.9 Kgp/s
	XMR	PPLNS	53.228 MH/s
Litecoinpool.org	LTC	PPS	25.98 TH/s
Slush	BTC	Score	5.36 EH/s
Ethermine	ETH	PPLNS	40.7 TH/s
	ETC	PPLNS	3.8 TH/s
	ZEC	PPLNS	442.6 MSol/s
f2pool	BTC	PPS	13.83 EH/s
	ETH	PPS	22.07 TH/s
	LTC	PPS	48.65 TH/s
	ETC	PPS	180.40 GH/s
	ZEC	PPS	945.76 MSol/s
	DASH	PPS	462.33 TH/s
	SIA	PPS	11.61 TH/s
	XMR	PPS	35.02 MH/s
Multipool	BTC	Exponential	0.36 PH/s
	BCH	PPLNS	0.081 PH/s
	LTC	PPLNS	14.45 GH/s
	DGB	Proportional	0.3 PH/s
MinerGate	BTG	PPLNS, PPS	4 Sol/s
	ETH	PPLNS	3.7 GH/s
	ETC	PPLNS	1.4 GH/s
	ZEC	PPLNS	7 KSol/s
	XMR	PPLNS, PPS	3.4 MH/s
	BCN	PPLNS, PPS	414.5 MH/s
Suprnova	BTG	Proportional	159 KSol/s
	ZEC	Proportional	4.29 KSol/s
	DGB	Proportional	3.1 TH/s
	DASH	Proportional	192.46 TH/s
Coinotron	ETH	PPLNS, RBPPS	806.6 GH/s
	ETC	PPLNS, RBPPS	630.7 GH/s
	ZEC	PPLNS, PPS	1.3 MSol/s
	BTG	PPLNS, PPS	13.9 KH/s
	DASH	PPLNS, PPS	65.5 TH/s