UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA

Master's Thesis

for obtaining the academic degree
Magister en Ciencias de la Ingeniería Informática

A Framework for Rational Mining in Proof-of-Work Blockchain Networks

by

Jean-Pierre Villacura

Thesis Committe

Supervised by:	Ph.D. Mauricio Solar	Universidad Técnica Federico Santa María, Valparaíso-Chile
Coreferent:	Ph.D. Raúl Monge	Universidad Técnica Federico Santa María, Valparaíso-Chile
Intern Evaluator:	Ph.D. J. Ricardo Ñanculef	Universidad Técnica Federico Santa María, Valparaíso-Chile
External Correferent	Ph.D. Romina Torres	Universidad Adolfo Ibañez, Viña del Mar-Chile
Commission chair:	Ph.D. J. Ricardo Ñanculef	Universidad Técnica Federico Santa María, Valparaíso-Chile

2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título X Tesis de Postgrado

Título del trabajo: _**A Framework for Rational Mining in Proof-of-Work Blockchain Networks**______________________

Nombre del candidato(a): _**Jean-Pierre Jesús Villacura Rojas**____

Carrera / Grado: _**Magíster en Ciencias de la Ingeniería Informática**____

Campus: _Casa Central_______________ **Departamento:** _Departamento de Informática__________________________

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Mauricio Solar, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

X El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 27-02-2026 **Firma:** Mauricio Solar Fuentes

Estudiante o Candidato(a):

Fecha: _17-12-2025____________ **Firma:** Jean-Pierre

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Declaration of authorship

I hereby declare that I wrote this thesis on the subject

A Framework for Rational Mining in Proof-of-Work Blockchain Networks

by

independently. I did not use any other aids, sources, figures or resources than those stated in the references. I clearly marked all passages that were taken from other sources and cited them correctly.

Furthermore I declare that – to my best knowledge – this work or parts of it have never before been submitted by me or somebody else at this or any other university.

Jean-Pierre Villacura

Valparaíso, January 28, 2026 ____________________________

Para mi Familia, Profesores,

Amigos, Personas geniales que tuve la oportunidad de conocer, mis gatos, el piano y por la Patria.

Acknowledgments

Agradezco enormemente a mi Familia, al Departamento de Informática de la UTFSM y a mis amigos.
A mi familia porque a pesar que fue complicado, pudimos salir adelante, mi mamá es la más fuerte y benevolente de todos.
Al Departamento de Informática de la Universidad Técnica Federico Santa María y la Universidad porque con ustedes pasé grandes momentos. Me gustaría destacar a Pabla por su infinito apoyo a todos los estudiantes y Profesores de Postgrado. A los profesores Mauricio y Raúl, quienes me guiaron en la tesis, y tambien me colaboraron con cartas de recomendación para todos los programas que postulaba.
Al voluntariado Technovation Girls Chile, que siento que entregue ayuda y recibí mil veces esa ayuda.
Agradezco mucho a mis amigos, estoy contento de tenerlos. Tambien considero a quienes dejamos de serlo, fueron gratos momentos que compartimos.
Agradecer tambien a Global Affairs Canada y la beca ELAP, cuya adjudicación me permitió realizar una pasantía de investigación de 6 meses en Queen's University, Canada. Puedo observar como la nieve se derrite de los techos y cae desde estos enormes edificios construidos con piedra caliza mientras escribo esta sección.
Additionally, I would like to express my gratitude to Lynn and Stew as you warmly received me on Canada. I am also grateful to Professor Mohammad Zulkernine and Amin Fakhereldine, whose hosted me on Queen's Reliable Software Technology and guided me.
I am deeply grateful to God and my grandparents, I know for sure that they have helped me until now.
Thank you UTFSM, Queen's U.
Wherever I go, may my persistence remain, my strength endure and the luck be on my side.

Abstract

This thesis proposes a framework for rational mining in Proof-of-Work blockchain networks through decision-making, addressing the challenge of selecting the most profitable mining configuration considering dynamic network conditions. Mining profitability depends on multiple factors, such as hardware, cryptocurrencies, mining pools and reward systems, making the selection of mining configuration complex. Existing academic models and online mining calculators simplify the impact of mining pools and reward systems.

To address these limitations, this work introduces a decision-support framework that integrates hardware devices, cryptocurrencies, mining pools, and reward systems using real-time blockchain data. The selection problem is formalized as a binary linear programming model that identifies the optimal mining configuration based on expected profitability. The framework incorporates reward system formulations and mining pools, supporting pool-based mining environments.

The proposed framework is validated through simulations, comparisons with an existing academic model, external mining calculators, and real mining executions. Experimental results show that the profitability estimations produced by the framework are consistent with observed real mining scenarios. Additionally, The framework successfully identified the rational mining option among available alternatives.

Overall, this thesis provides a practical and extensible foundation for rational mining decision-making in Proof-of-Work blockchain networks.

Contents

List of Tables 3

List of Figures 4

Chapter 1: Introduction 5

1.1 Motivation . . . 5

1.2 Problem Statement . . . 6

1.3 Contribution . . . 7

1.4 Organization of Thesis . . . 7

Chapter 2: Literature Review 8

2.1 Bibliographic Search Methodology . . . 8

2.2 Related Work . . . 11

2.3 Identification of Gaps . . . 12

Chapter 3: Background 13

3.1 Blockchain . . . 13

3.2 Mining . . . 13

3.3 Devices . . . 14

3.4 Pools and reward systems . . . 15

Chapter 4: Proposed Framework 17

4.1 General Overview . . . 17

4.1.1 Conceptual Diagram . . . 18

4.1.2 Comparison with tools and models . . . 19

4.1.3 Scope of the framework . . . 19

4.2 Data Management and Preprocessing . . . 20

4.2.1 Data Sources . . . 20

4.2.2 Entity-Relationship Model . . . 20

4.3 Calculation of Reward Systems and Assumptions . . . 21

4.3.1 Estimation of Sharerate . . . 21

4.3.2 Estimate Solved Blocks by Pools 22
4.3.3 Reward Systems 23
4.3.4 Proportional Reward System 24
4.3.5 PPS Reward System 24
4.3.6 FPPS Reward System 25
4.4 Linear Programming Model for Rational Mining 25
Chapter 5: Implementation, Simulation and Validation **28**
5.1 Implementation 28
5.1.1 Tools and Environment 28
5.1.2 Data Collection 29
5.1.3 Database Design 31
5.1.4 Implementation of Linear Programming Model 31
5.2 Simulation 33
5.2.1 Base Scenario 33
5.2.2 Proof of Optimality 36
5.3 Validation 37
5.3.1 Validation of profitability estimation 37
5.3.2 Functional Validation of the Framework 40
Chapter 6: Results and Discussion **42**
6.1 Results and Discussion 42
Chapter 7: Conclusions **47**
Bibliography **49**
Chapter 8: Appendix **51**
.1 Literature Review 52
.2 Database Schema Implementation 57
.3 Linear Programming Model 65

List of Tables

2.1 Keywords, databases, and number of papers retrieved 9
2.2 Summary of Related Work 11
5.1 Category, Variables and Sources used in the Implementation 30
5.2 Input of the Base Simulation 34
5.3 Output of the base simulation 35
6.1 Absolute Estimation Error of the Proposed Framework and External Mining Calculators Compared to Real Mining Execution 43
6.2 Optimal mining configurations identified by the proposed framework 45
.1 Literature review of selected works 52

List of Figures

2.1 PRISMA flowchart for the literature review. 10

4.1 Conceptual diagram of the Proposed Framework 18
4.2 Entity Relation Model 21

5.1 Database schema implemented for the framework 32
5.2 Comparison of profit across ProfitMax, Proposed Framework, Xu (2020), and Observed values. 39

6.1 Comparison of profitability across mining configurations as estimated by the framework, observed in real mining environment, and reported by external calculators 43
6.2 Comparison of absolute prediction error between the proposed framework and external mining calculators across four mining configurations. 44

Chapter 1

Introduction

1.1 Motivation

Mining in Proof-of-Work(PoW) cryptocurrencies is essential for the integrity of PoW blockchain networks. This process verifies the integrity of transactions stored in the distributed ledger through the solution of computationally intensive problems that requires computational effort from miners (Nakamoto (2009)). As an incentive for contributing with computational effort, miners receive a reward in cryptocurrency for being the first to succesfully find and declare a valid solution (Rosenfeld (2011)).

Because cryptocurrencies have an economic value in open markets, the reward obtained through mining can be exchanged for money (Delgado-Mohatar et al. (2019)). Therefore, rational mining focuses on maximizing these rewards, aiming to increase the profitability perceived by miners who dedicate computational resources to the process.

Miners usually join mining pools to increase their chances to solve blocks, which coordinates computational efforts and distributes the reward accordingly to the reward system adopted by the pool (Rosenfeld (2011)). The choice of a reward system impacts the miner's profitability under real-world conditions.

Miners have progressively adopted specialized hardware using devices as FPGA and ASICs to improve performance and energy efficiency, replacing the general-purpose CPU and GPUs used in early stages (Derks et al. (2018)). These specific-purpose devices has restriction-compatibility with certain algorithms of mining, and these algorithms has a restriction-compatibility with cryptocurrencies. As a result, the profitability of mining depends not only on market factors such as price and network difficulty, but also on technical compatibility between hardware, algorithms, and cryptocurrencies.

1.2 Problem Statement

In rational mining, miners aim to maximize profitability by making decisions that adapt to changing network conditions. Although there are various strategies to improve mining returns, this work focuses on identify the most profitable mining option. It proposes a framework designed to support mining decisions by integrating the main components that influence the profitability: hardware devices, mining algorithms and cryptocurrencies's network conditions, pools and reward systems. This framework integrates real-time data to support the decision-making of miners integrating pools and reward systems to determine the most rational mining option under dynamic conditions.

Mining profitability depends on multiple factors: Each hardware device supports only certain mining algorits, which in turn are compatible with specific cryptocurrencies. Mining Pools also work with selected cryptocurrencies and reward systems such as PPS, FPPS, Proportional, all of which influence the expected payout of a miner. Additionally, each cryptocurrency network has its own parameters: hashrate, difficulty and market price, values that fluctuate over time. Therefore, identifying the rational mining option is not a trivial task.

Several studies have analyzed mining profitability in Proof-of-Work networks from economic or energy-efficiency perspectives (Derks et al. (2018); Delgado-Mohatar et al. (2019)). However, most academic models simplify the role of mining pools and reward systems, even though these factors significantly influence expected returns. Regarding online tools, mining calculators provide practical estimations, but they cannot incorporate additional data that may affect the rational mining choice. They also lack configurability and, in most cases, do not consider the impact of reward systems. As a result, there is no tool or framework capable of integrating relevants elements involved in mining: hardware device, cryptocurrencies, mining pools and reward systems. Although a decision-making framework for mining ProfitMax (Cho et al. (2024)) has been proposed, it focuses on energy-related impact and it does not incorporate the effect of mining pools and reward systems on expected payouts. To address this gap, this thesis proposes a rational mining framework that integrates real-time blockchain data, mining pools, and reward systems, using the formulations of Rosenfeld (2011). This framework identifies the most profitable mining configuration through a binary linear programming model.

Research Question

How can a rational mining framework be designed to identify the most profitable mining configuration in Proof-of-Work blockchain networks under dynamic conditions?

1.3 Contribution

The main contributions of this thesis are as follows:

- Propose a rational mining framework for PoW networks that integrates key components influencing mining profitability, including hardware devices, cryptocurrencies, mining pools, and reward systems.
- Develop a binary linear programming model capable of identify the rational mining option.
- Implement a prototype of the proposed framework that integrates real-time blockchain data through APIs, using estimations and formulas based on the work of (Rosenfeld (2011)) to provide rational mining recommendations according to current network conditions and the available data.
- Validate the proposed framework through academic literature and real-world scenarios, demostrating its effectiveness in supporting rational mining decisions.

1.4 Organization of Thesis

This thesis is organized as follows:
Chapter 2 presents a literature review on cryptocurrency mining and reward systems, outlining the search keywords, sources consulted, and the identification of research gaps.
Chapter 3 provides the background necessary for the proposed framework, including considering hardware devices, pools and reward systems.
Chapter 4 describes the proposed framework, detailing its components, model and the formulas utilized for profitability estimation.
Chapter 5 presents the implementation of the framework and its validation through simulations and comparison with academic literature.
Chapter 6 discusses the results obtained from the proposed framework, and Chapter 7 concludes the thesis by summarizing the main findings.

Chapter 2

Literature Review

2.1 Bibliographic Search Methodology

To identify the most relevant academic works on rational mining in Blockchain Proof-of-Work (PoW) networks, with particular emphasis on mining pools and reward systems, a systematic search strategy was adopted following the PRISMA guidelines (Page et al. (2021)), ensuring a transparent and comprehensive review of the existing literature.

To guide the process, the following research questions were defined: 1. What research has been conducted on mining pool reward systems and profitability models in Proof-of-Work Blockchain networks?, 2. Are there existing frameworks that integrate cryptocurrency, pool reward schemes? The bibliographic search and screening process using PRISMA was designed to address these questions.

Based on the PRISMA methodology (Page et al. (2021)), the databases used were Scopus as the primary source and the WoS Core Collection as a complementary one, due to their extended in the field of computer science. Google Scholar was used only supplementarily to consult additional relevant articles when needed. Regarding database searches, the following keywords were used: ("Blockchain" OR "Proof of Work" OR "PoW") AND ("mining" OR "pool" OR "reward systems"). This strategy allowed the retrieval of papers related to mining in PoW blockchain networks, with a particular focus on reward systems and mining pools.

Table 2.1 summarizes the keywords used for the bibliographic search and the number of results obtained in Scopus and the WoS Core Collection. Scopus searches were filtered by "Search within Article Title, Abstract and Keywords", while WoS searchers used the "Topic" filter. All searches were carried out on September 10, 2025, at 15:00. Google Scholar was used used only as an additional source to access

Table 2.1: Keywords, databases, and number of papers retrieved

Keywords	Database	number of Papers
("Blockchain" OR "Proof-of-Work" OR "PoW") AND (mining OR pool OR reward systems)	Scopus	4347
	WoS Core Collection	2353

relevant publications not indexed in the primary databases.

The search queries were constructed using two main conceptual blocks: (1) technological terms such as "Blockchain," "Proof-of-Work," and "PoW"; and (2) context terms such as "mining," "pool," and "reward systems." Economic terms such as "profitability," "model," or "optimization" were deliberately excluded to avoid missing potentially relevant studies that do not explicitly use this terminology.

The literature review began with 6700 records retrieved through keywords searches in two primary databases (Scopus: 4347, WoS: 2353). After removing duplicates(including early access and final versions), 4057 uniques records remained. Screening was then conducted in three stages, each guided by specific question. First question relied primarily in titles, consulting abstracts when necessary, the second stage was based on abstract review and the third stage consisted of full-text assessment. The questions and exclusion criteria for each stage are described in the following paragraphs, Figure 2.1 refers to the methodology used.

For the screening stage, the following question was applied: **Does the paper address Proof-of-Work mining, pools, or reward systems in a way that could potentially contribute to a rational mining framework?**. Out of 4,059 records, 104 were retained after applying the first stage, which was primarily based on title review, reference in Appendix Table .1.
Exclusion criteria for this question include:

- Studies of miner behavior in attacks contexts.
- Energy-focused analyses.
- Game-theoretic simulations not related to profitability.
- Proposals of dynamic difficulty adjustment or optimization approaches focused on latency.
- Works on block size or the design of new reward systems.

The same screening question was applied consistently across three successive stages of increasing depth: title-level, abstract-level, and full-text review. This ensured consistency in the inclusion criteria while making the process feasible and efficient.

Although in the first stage the screening was primarily based on titles for first stage, abstracts were consulted when the relevance was unclear, to avoid excluding potentially useful studies.

In the second stage, based on abstract review, out of 113 records, 29 were retained. In the third stage, involving full-text review, out of 19 records, 6 were retained.

The adoption of a methodology inspired on PRISMA guidelines (Page et al. (2021)) provides transparency to the literature review process and supports the originality of the proposed work.

Figure 2.1 presents the adapted PRISMA flowchart, which illustrates the number of records identified, screened and excluded at each stage. The papers retained through this process constitute the core litera-

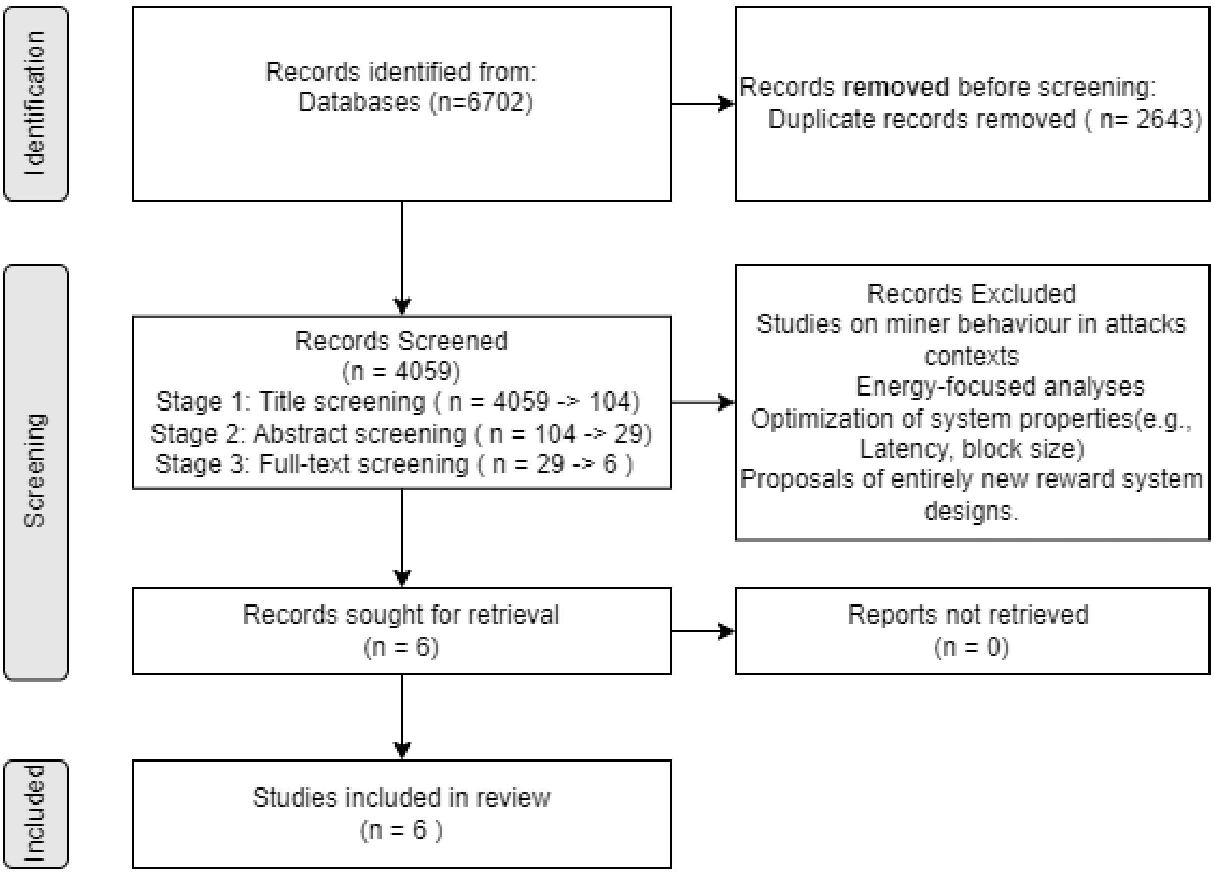

Figure 2.1: PRISMA flowchart for the literature review.

ture discussed in the next section, Related Work. The resulting papers provide the empirical foundation for identifying existing analytical models of mining profitability and gaps concerning the integration of reward systems, which will be used in the proposed framework section.

2.2 Related Work

This section reviews the main studies identified through the bibliographic search described earlier. The goal is to discuss their contributions, limitations and their relation to the proposed framework.

Table 2.2: Summary of Related Work

Reference	Main Contribution	Limitations
Rosenfeld (2012)	Provides the mathematical foundations for pool reward systems and payout formulas used in Proportional, PPS, and related schemes.	Purely theoretical, It requires practical estimations to apply its formulas to real-world mining scenarios.
Delgado (2019), Derks (2018), Soria (2022)	Analyze mining profitability, mainly in solo-mining settings, proposing economic or energy-based models.	Do not include mining pools and reward systems
Xu (2020)	Models dynamic pool selection using an evolutionary-game approach and provides formulas for PPS and PPLNS behavior.	Lacks practical implementation for real mining environments.
Cho (2024)	Introduces a decision-support framework centered on energy efficiency, dynamic pricing, and real-time analytics.	Focuses on energy optimization rather than profitability across different mining configurations and it does not include mining pools and reward systems.

Early work such as (Rosenfeld (2011)) established the mathematical foundation for analyzing rewards systems in Bitcoin mining Pools. This work is crucial for the development of the proposed framework as it provides formulas of reward systems that will be integrated. However, Rosenfeld's analysis remains theoretical and does not extend to real-world mining environment.

Several studies, such as (Delgado-Mohatar et al. (2019); Derks et al. (2018); Soria et al. (2023)), have focused on the profitability analysis of solo mining and therefore do not consider the impact of mining pools or their reward distribution systems. In contrast, this thesis incorporates profitability formulas for Proportional, PPS, and FPPS schemes, proposing practical estimations, which is essential given that most mining activity is carried out through pools.

(Xu et al. (2020)) formulates the dynamic pool selection problem as evolutionary game considering PPS and PPLNS reward's systems for a two-pool scenario. The formulas presented in this work can technically support the modeling of PPS and PPLNS schemes and are useful for simulation-based analyses. However, the study remains purely theoretical and lacks a practical implementation that can be applied in real mining environments. This thesis addresses part of this gap by adapting PPS-based estimations to real-world conditions within a decision-making framework for identifying profitable mining options. (Cho et al. (2024)) proposes a framework Profitmax that optimizes PoW mining focusing on energy efficiency strategies and profitability integrating dynamic energy pricing and real-time data to support miners. However, its primary focus lies in optimizing energy consumption and operational sustainability rather than modeling profitability from different options of miners, which are the main topic of our work. Our proposed framework extend the calculation of profitability including mining pools and reward systems, which have impact on miner's expected revenue and are not considered in Profitmax.

Overall, the literature addresses individual aspects of mining, but no study provides an integrated framework that supports miners in making profitability decisions while incorporating mining pools and reward systems in real-time environments. Addressing this gap is the focus of the present thesis. Table 2.2 summarizes the related work.

2.3 Identification of Gaps

According to the methodology applied in the literature review, no model has been developed that allows miners to select the most profitable cryptocurrency, taking into account the available hardware, cryptocurrencies, pools, and different reward systems. This work fills this gap by offering a framework that helps miners in the decision-making of mining, cryptocurrencies, mining pools and reward systems integrating real-time data.

Chapter 3

Background

3.1 Blockchain

A blockchain network operating under the Proof of Work (PoW) protocol is an asynchronous peer-to-peer system in which each node maintains a distributed ledger that records economic transactions grouped into blocks (Nakamoto (2009)). Blocks are created by miners, who dedicate computational resources to solve a cryptographic puzzle associated with the block (Bistarelli et al. (2022)). This process involves repeatedly modifying the nonce value within the block header until obtaining a hash value that, when processed through the cryptographic hash function, falls below the target defined by the network's current difficulty (Zhu et al. (2018)).

3.2 Mining

The Proof of Work (PoW) consensus mechanism ensures the integrity of the blockchain network by validating the transactions stored in blocks through the resolution of a computationally intensive mathematical problem. This challenge consists of the continuous calculation of cryptographic hash functions (Panuntun et al. (2023)),a process that requires a significant amount of computational power distributed across the network (Tripathi et al. (2023)).

The resolution of the computational problem consists of iteratively computing cryptographic hash functions over a nonce i, applied to a tuple that includes information from the previous block, the current block, the public key, and the nonce itself (Bard et al. (2022)). The problem is considered solved when the resulting hash value, obtained from this computation, is numerically smaller than the Target Difficulty defined by the network. This target value correspond to the difficulty of the Network.

Hashrate is a standard measure of the computational power available to a miner for solving the PoW puzzle. It represents the number of hash attempts a device can perform per second and is commonly used to estimate the efficiency of mining hardware (Bistarelli et al. (2022)). Difficulty **D** is a parameter periodically adjusted by the network to regulate the effort required to find a valid block. It is defined so that each computed hash has a probability of $\frac{2^{16}-1}{2^{48} \cdot D}$ of being valid (Rosenfeld (2011)) in Bitcoin and similar cryptocurrencies. In Bitcoin, the difficulty is recalibrated every 2016 blocks based on the total computational power of the network, ensuring that the average block mining time remains close to 10 minutes (Nakamoto (2009); Bard et al. (2022)). Other PoW-based blockchain networks employ similar adjustment mechanism, although the specific parameter values vary depending by protocol design.

The following formula estimates the expected number of blocks mined with a hashrate h over a time interval of t t: $\frac{(2^{16}-1)ht}{2^{48}D}$ (Malone and O'Dwyer (2014)) ; (Rosenfeld (2011)).

Mining pools distribute rewards among miners according to their contributed computational effort, measured through *shares*. A share is a hash that satisfies a lower target set by the pool and serves as proof to estimate the amount of work contributed by each participant to the pool (Zhu et al. (2018)). On average, the number of shares submitted by a miner is proportional to the total number of hashes computed (Rosenfeld (2011)).

The cost of producing a bitcoin can be estimated using the formula $\frac{D\,2^{32}k}{3.6 \cdot 10^{12}\,\epsilon R}$ which can also be applied to other PoW-based cryptocurrencies (Delgado-Mohatar et al. (2019)). In the formula, k represents the electricity cost, ϵ the device efficiency, and R the block reward obtained by successfully solving a block (Delgado-Mohatar et al. (2019)). This formula is appropriate for solo mining with a single device. However, it becomes limited in the context of mining pools, as miners use heterogeneous hardware configurations and operate in different locations with varying electricy prices. To the best of our knowledge, no previous work has incorporated pool reward systems into profitability estimation models in real-world scenarios. Therefore, it is necessary to develop a model to estimate profitability including mining pools and reward systems which can be applied to a real mining environment.

3.3 Devices

In PoW-based mining, different types of devices are employed, including CPUs, GPUs, FPGAs, and ASICs (Bedford Taylor (2017); Khosravi and Säämäki (2023)). FPGAs are integrated circuit designed to be configured by the user, with its main advantage being its reconfigurability. This feature provides high versatility, enabling the mining of multiple algorithms, while also offering high performance and support for parallel programming (Farooq et al. (2012)).

An ASIC is an integrated circuit designed for a specific purpose, allowing high optimization in performance, energy consumption, and size for the task it is built to perform (Taraate (2022).)

Both components typically achieve higher performance and lower energy consumption in the mining process compared to general-purpose computers. However, their main disadvantage is the requirement of a considerable initial investment, in addition to being designed for specific purposes only.

Important metrics in mining include the hardware hashrate and its energy consumption. These parameters are commonly used to compare mining devices: hashrate and power efficiency (Malone and O'Dwyer (2014); Delgado-Mohatar et al. (2019)). Since ASICs are specifically designed and optimized for mining, they generally achieve higher performance and lower energy consumption compared to CPUs and FPGAs across these metrics.

3.4 Pools and reward systems

There exists two types of rewards: Block reward and transaction fees (Malone and O'Dwyer (2014)) (Xu et al. (2020)). This reward structure incentivizes miners to contribute computational resources to the network. The possibility to being the first to complete the solution depends on the relation between the global hashrate and one's own hashrate, making it difficult to succesfully solve when mining solo (Rosenfeld (2011)).

Mining in pools increases the probability of solving a block by coordinating the computational power of multiple miners, who then receive a shared reward from the pool (Xu et al. (2020)). It is estimated that up to 98% of Bitcoin's total computational power comes from mining pools (Zhu et al. (2018)). Joining a pool typically requires meeting a minimum hashrate threshold, and network delay is a critical factor, as it may lead to potential losses (Zhu et al. (2018)). For the purposes of this work, it is assumed that miners meet the hashrate and network connectivity requirements typically needed to join a mining pool.

To distribute rewards in a mining pool according to each miner's contribution, mining pools employ different reward systems: Proportional, PPS and FPPS, PPLNS, and others (Rosenfeld (2011)).

The objective of any mining pool is to maximize the rewards obtained for its members. However, when miners delay reporting a valid block to the pool, there is a risk that an external miner may find and report the block first, reducing the collective gains received by the pool (Schrijvers et al. (2017)). A reward system is therefore considered *incentive-compatible* if miners are motivated to report solutions immediately (Schrijvers et al. (2017)), which is also regarded as a positive attribute with respect to fairness within the pool. The design of the reward system strongly influences miner behavior (Schrijvers et al. (2017)).

Rational mining refers to the set of decision made by miners to maximize their expected profitability. In this work, a mining option is considered rational when it offers the highest expected profit among the available alternatives.

When a pool mine a block, his luck increases. Pools with 100% of luck implies that they solve all the blocks estimated according to the pool hashrate, global hashrate, block generation time in a timelapse.

The proportional reward system is one of the most widely used methods and consists of distributing the block reward according to the proportion of shares contributed during a round (Rosenfeld (2011)). A round is defined as the time interval between the discovery of two consecutive blocks by the pool, meaning that miners receive earnings only when the pool successfully finds a block. Other widely adopted schemes are PPS and FPPS, which pay miners for every share they contribute, regardless of whether the pool successfully finds a block (Rosenfeld (2011)). PPS distributes the payment according to block reward and FPPS includes transaction reward fees. (Albrecher et al. (2023)) studies the presence of transactions fees in the block rewards. The concepts presented in this section provide the foundation for the rational mining framework proposed in Chapter 4.

Chapter 4

Proposed Framework

4.1 General Overview

The purpose of this work is to provide a comprehensive decision-support system for mining in PoW networks that incorporates Pools and rewards systems, elements that are often simplified in academia literature. Current tools, such as mining calculators, generally do not allow users to integrate detailed information about pools, offering a limited set of options and tipically simplifying the impact of reward systems on the profitability estimation.

This framework integrates the main components of pool mining: Hardware devices, cryptocurrencies, pools and rewards systems to help miners in selecting the most profitable option by integrating real-time data and practical implementations. Because the mining environment is highly dynamic and multiple configurations exist, optimal choice can change continously as networks parameters change.

Each component is linked within the framework to form a decision-making model for miners. Different types of devices can be used to perform mining such as CPU, GPU, FPGA and ASICs (Bedford Taylor (2017)), each providing a specific hashrate and energy consumption, two key metrics for rational mining. These devices are compatible with different cryptocurrencies through mining algorithm supported. Additionally, each cryptocurrency has difficulty, block reward and a global hashrate. Mining pools group miners to cooperatively mine blocks, and the rewards they distribute depend on the reward system implemented by the pool. All of this information is integrated into the proposed decision-making model.

Real-time data from different cryptocurrencias are extracted by this implementation, allowing the model to provide a solution considering the current state of the networks. The model identifies the most profitable option considering Device, Cryptocurrency, Pool and reward system. As a result, it serves both as a theoretical framework for rational mining and a practical tool capable of guiding miners in their decision-making.

4.1.1 Conceptual Diagram

Figure 4.1 present a conceptual diagram for the proposed framework for rational mining.

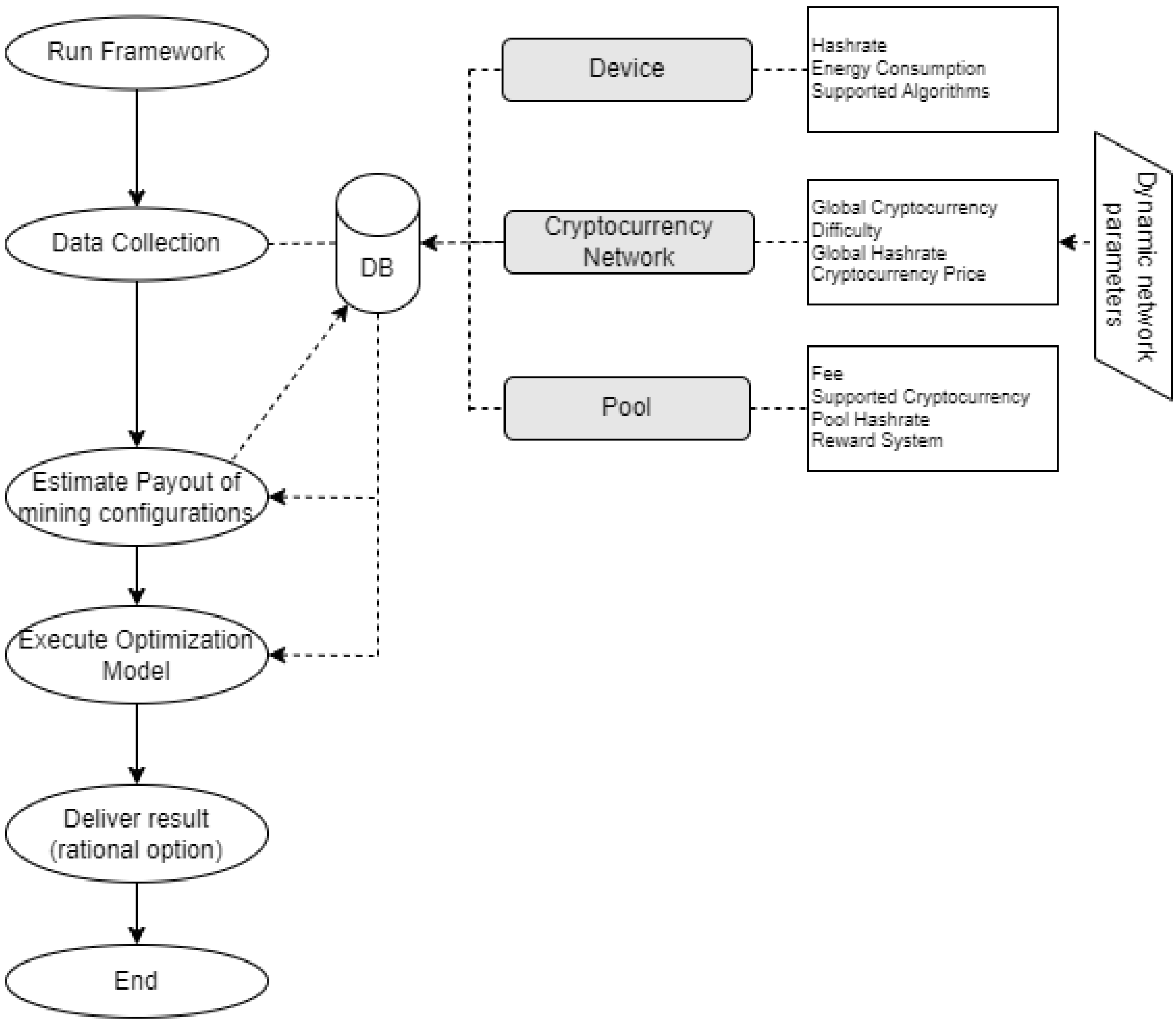

Figure 4.1: Conceptual diagram of the Proposed Framework

The diagram illustrates four key elements relevant to mining: the device, the cryptocurrency network, mining pools, and reward systems. The framework operates through five main steps. First, it is executed to collect data from various sources and store it in a database. Using this information, the framework then estimates the expected payout for different mining configurations. Next, the optimization model is applied to identify the most profitable configuration. Finally, the result is delivered, completing the execution of the framework.

4.1.2 Comparison with tools and models

It is important to compare the proposed framework to existing academic models and mining calculators available on Internet to evaluate contribution.
The profitability calculations in this work are compared with the Profitmax Framework (Cho et al. (2024)) under equitative conditions. Specifically, an equivalent baseline scenario is defined by neutralizing the parameters associated with mining pools and reward systems in the proposed framework (assigning zero pool fees), in order to ensure homogeneous evaluation conditions. This allows isolating the impact of explicitly modeling mining pools and reward mechanisms.
Regarding online calculators, widely used tools as WhatToMine[1] allow users to input hardware specifications and calculate estimated daily profits. However, they do not consider mining pools or reward systems, and their fixed, limited set of supported cryptocurrencies restricts the range of possible solutions.
In contrast to Minerstat[2], which provides a predefined and closed set of cryptocurrencies, mining pools and reward systems, the proposed framework is designed to be extensible and configurable by construction. Minerstat operates as a black-box calculator, where users are restricted to the data sources, pools, reward mechanisms and assumptions, without the possibility of incorporating additional cryptocurrencies, custom pools, alternative reward schemes or user parameters.

The proposed framework explicitly exposes mining pools and reward systems as inputs to the decision model. This allows users to integrate new cryptocurrencies, pools and reward systems as they emerge, as well as to modify parameters such as pool fees, payout schemes, network difficulty, hashrate or reward structures. As a result, the framework enables the exploration of a significantly larger configuration space, which is particularly relevant in the context of rational mining, where profitability is highly sensitive to pool-specific rules and rapidly changing network conditions.
The proposed framework enables the integration of dynamic blockchain and pool data, while also allowing users to incorporate information about Pools, rewards systems and network conditions. This approach expands the range of possible configurations providing a more accurate set of optimal solutions.

4.1.3 Scope of the framework

The following points contain assumptions in the creation of this framework to land it in real case scenario.

- The framework evaluates one hardware device at a time. Cases where miners have more than one

[1] https://whattomine.com/
[2] https://minerstat.com/coins

device need to run it separately.

- Network connectivity is assumed to be optimal. Therefore phenomenons related to Delay as Stale shares (Rosenfeld (2011)) are not considered in this work.

4.2 Data Management and Preprocessing

Mining profitability depends on parameters that fluctuate over time: network difficulty, cryptocurrency price and global hashrate. Therefore, it is required an approach to obtain data from sources and manage it appropriately within the framework.

4.2.1 Data Sources

Data are collected from public and reliable sources that provide information about:

- **Blockchain Networks**: network difficulty, cryptocurrency price and global hashrate. Although some parameters like Difficulty it can be obtained by accessing Blockchain stats, it is defined the retrieve through APIs.
- **Mining Pools**: Operational fees, mining pools parameters and the election of reward systems can be obtained through APIs. Estimation such as number of block solved daily by the pool, it is calculated with information recollected through APIs.
- **Hardware specifications**: hashrate, energy consumption and supported algorithms for different devices are introduced manually to database.

Data is recollected each time the framework is executed, producing a dataset aligned with the requeriments of the the linear programming model.

4.2.2 Entity-Relationship Model

Figure 4.2 represents the entity relation model of the proposed framework.

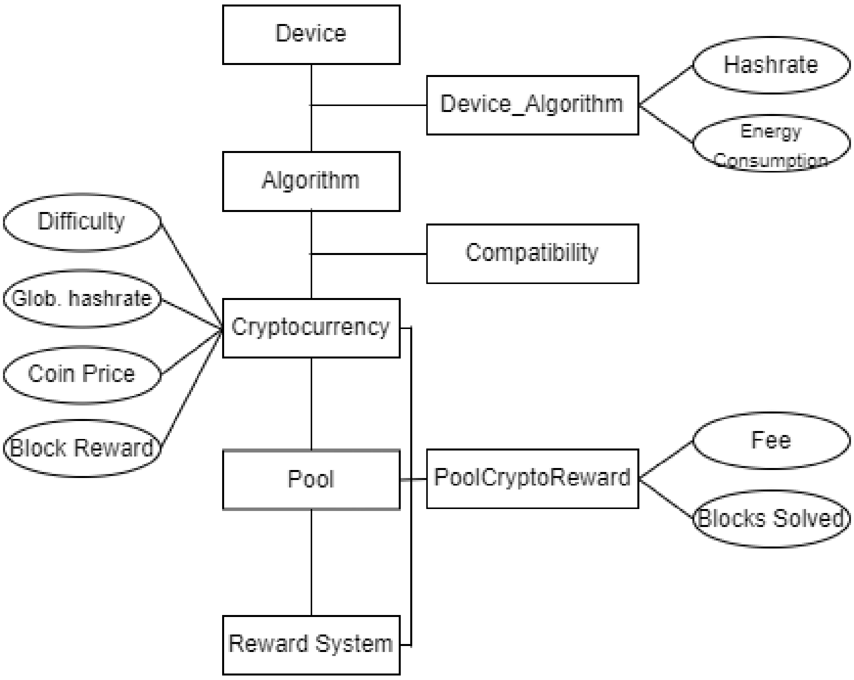

Figure 4.2: Entity Relation Model

The proposed framework includes a relational database model that integrates the main entities of the rational mining ecosystem: devices, algorithms, cryptocurrencies, mining pools, and reward systems.
The relational structure should incorporate one-to-many and many-to-many relationships through intermediate tables that allow the capture of possible compatibilities and combinations between components. This design makes it possible to feed the optimization model with up-to-date and reproducible information, maintaining consistency between static components (hardware, algorithms) and dynamic parameters (network, difficulty, rewards, and fees).

4.3 Calculation of Reward Systems and Assumptions

4.3.1 Estimation of Sharerate

Sharerate represents the rate at which a miner produces shares when participating in a mining pool. A share correspond to a hash that satisfies a target difficulty d imposed by the pool, which is intentionally set lower than the network difficulty in order to allow frequent verification of the computational work performed by miners Rosenfeld (2011).
The payout of the miners it is established accordingly to the number of shares (Rosenfeld (2011)).
Reward systems can be adapted to a variable difficulty. using the formula $p' = \frac{d}{D}$ with d the share difficulty and D the networok difficulty (Rosenfeld (2011)).

In the case of Bitcoin, the share difficulty d is adjusted constantly by the pool for each miner to accomplish that miners report shares in a timelapse of 10 to 30 seconds [3]. This mechanism is extendable to other cryptocurrencies and pools achieve reduced server load.

Therefore, the estimation of Sharerate represents the miner's computational contribution, determined by the hashrate, constant c and share difficulty d assigned by the pool, as it is shown in formula 4.1. The pool regulates d to maintain stable submission of shares from miners.

$$S = \frac{h_{miner}}{c_l\, d} \tag{4.1}$$

where:

- S is the *sharerate* (shares per second),
- c_l is a constant of normalization for the cryptocurrency l.
- h is the miner's hashrate (hashes per second),
- d is the share difficulty assigned by the pool.

This relationship derives from the probability that a hash meets the share target, where the expected number of hashes to find a share of difficulty d is $2^{32}\, d$ (Rosenfeld (2011)). Therefore, the expected sharerate is given by the miner's hashrate divided by this expected number of hashes.

4.3.2 Estimate Solved Blocks by Pools

Proportional and PPLNS reward systems depend on the pool succesfully solving blocks to distribute the rewards to miners (Rosenfeld (2011)). The process of solving a block is a random event, but the number of expected solved blocks it can be effectively approximated using the formula 4.3.

The process of solving a block can be estimated with Poisson because it corresponds to a binary event: Block is solved or not. Each hash attempt is an independent event with a small and constant probability of success and it does not occur simultaneously. Under this assumption, the block discovery rate λ_m can be estimated by either one of this formulas:

$$\lambda_{m1} = \frac{h_{\text{pool},m}\, T_l}{h_{\text{net, l}}} \qquad \text{and} \qquad \lambda_{m2} = \frac{h_{\text{pool},m}}{D_l \cdot c_l} \tag{4.2}$$

Equation 4.2. Estimation of the block discovery rate λ_m for pool m using (left) the hashrate ratio method and (right) the global difficulty formulation.

[3] Stack Exchange: Does a logical mining pool assign harder and harder difficulty to the same miner? https://bitcoin.stackexchange.com/questions/71898/does-a-logical-mining-pool-assign-harder-and-harder-difficulty-to-the-same-miner

where:

- $h_{pool,m}$ is the *pool hashrate* (hash per second)
- T_l is an approximated time that a cryptocurrency l is discovered (Block per seconds).
- $h_{net,l}$ is the network hashrate (hashes per second).
- D_l is the network difficulty and c_l the constant of normalization of the cryptocurrency.

The two expressions in equation 4.2 are equivalent. The first expression models the number of blocks discovered by a mining pool, which is a stochastic process driven by its computational power relative to the total network. This approach is practical in the context of the framework because hashrate of the pool is a parameter obtained previously and T_l is a parameter of approximated time of discovering a block by the cryptocurrency network l.

Alternatively, network difficulty and D and constant of normalization of cryptocurrency c_l can be used instead of Hashrate of the Network. As example, the normalization constant c for Bitcoin is $c = 2^{32}$ and Monero c=1.

The framework uses the first approach of λ_{m1} due to the hashrate parameters of the network are obtained by the framework. The expected number of blocks found by a pool m during a time window can be estimated as:

$$\mathbb{E}[G_m] = \lambda_i \, t \tag{4.3}$$

Equation 4.3. Formula 4.3 presents the expected number of blocks $\mathbb{E}[G_m]$ discovered by pool m during a time period t, given its block discovery rate λ_m. This expected value represents the theoretical average number of valid blocks that a pool is likely to discover over a specific period. This formula will be used as an input for computing expected rewards under proportional scheme, since the payout depends directly on the number of blocks solved by the pool.

For the purposes of this framework, only the expected value $\mathbb{E}[B_i]$ is considered, rather than modeling the full probability distribution. This simplification ignores the variance and mining luck associated with block discovery, but it provides a practical and consistent approximation for profitability estimation of the Proportional scheme.

4.3.3 Reward Systems

Reward systems are important to mining profitability, as they define how the total revenue obtained by a pool is distributed among participants (Rosenfeld (2011)). This framework integrates the most commonly adopted reward mechanisms in PoW Networks: Proportional, Pay per Per Share and Full Pay Per Share. Each scheme is modeled using formulations derived from the work of Rosenfeld (2011), adapting them to real-world scenarios and providing practical estimations for profitability analysis.

4.3.4 Proportional Reward System

This system distributes the total block reward among all participants in proportion to their contribution in a round. A round is defined as the interval between two succesfully mined blocks by the pool (Rosenfeld (2011)). According to (Rosenfeld (2011)), the payout for a miner who submits n valid shares of difficulty d of a total N shares in a round, subject to a pool fee f in a round is given by:

$$\text{Payout}_{prop} = (1 - f) \cdot \frac{n}{N} \cdot B \tag{4.4}$$

Due to the number of shares send by a miner is proportional to their hashrate (Rosenfeld (2011)), and in turn, the total shares of the pool is proportional to the pool hashrate, it generates the equation 4.5.

$$\frac{n}{N} = \frac{h_{miner}}{h_{pool}} \tag{4.5}$$

The expected number of blocks that a pool solves in a day can be estimated using Equation 4.6.

$$E[\text{blocks}_{pool}] = \frac{h_{pool}}{h_{global}} \times \frac{t}{T_{block}} \tag{4.6}$$

By combining the formulas 4.4, 4.5 and 4.6, the expected payout for this reward system is the following equation:

$$Payout_{miner,Pool} = (1 - f)(\frac{h_{miner}}{h_{global}})\frac{t}{T_{block}}B \tag{4.7}$$

Therefore, Equation 4.7 estimates the expected payout for a miner with device hashrate h_k in a pool operating on a cryptocurrency network with hashrate H_l, given a block reward B_l, an evaluation period of $t = 86\,400$ seconds (one day), and an estimated block generation time T_l.

$$Payout^{Prop}_{k,l,m} = (1 - f_m)(\frac{h_k}{h_l})\frac{t}{T_l}B_l \tag{4.8}$$

4.3.5 PPS Reward System

Miners receive an immediate and fixed payment for every valid share submitted, independently of wheter the pool finds a block (Rosenfeld (2011)). The expected payout is proportional to the miner's hashrate and is adjusted by the pool fee f_m and the network's block reward B_l.

$$\text{Payout}_{PPS} = (1 - f) \cdot p \cdot B \tag{4.9}$$

To consider the Dynamic share Difficulty mechanism implemented by Pools, $p = \frac{1}{D}$ can be replaced by $p = \frac{d}{D}$ to estimate profit with a difficulty share of d (Rosenfeld (2011)). It is important to notice that p is originally the possibility of a Hash be valid with difficult D, and the integration of dynamic share difficulties implies that each submitted share has a possibility d/D to be valid.

The following equation 4.9 represents the PPS formula adapted to this framework for estimating profitability.

$$\text{Payout}_{k,l,m}^{PPS} = (1 - f_m) \cdot \frac{d_m}{D_l} \cdot B_l \tag{4.10}$$

4.3.6 FPPS Reward System

Similar to PPS Reward System, miners receive an immediate and fixed payment for every valid share submitted, independently of wheter the pool finds a block. The expected payout is proportional to the miner's hashrate including two types of rewards: Transaction fee BT and block reward B.

$$\text{Payout}_{FPPS} = (1 - f) \cdot p \cdot (B + BT) \tag{4.11}$$

While block reward B is defined over a given period, transaction reward BT fluctuates over time. Therefore, a practical approach is required to estimate transaction fee reward.

$$\text{Payout}_{k,l,m}^{FPPS} = (1 - f_m) \cdot \frac{d_m}{D_l} \cdot (B_l + BT_l) \tag{4.12}$$

Formula 4.12 represents the original equation adapted to this framework to estimate profitability within the FPPS reward system.

4.4 Linear Programming Model for Rational Mining

The model corresponds to a binary integer linear programming problem and takes the following parameters and decision variables as input:

Technical characteristics of mining device $k \in K$**:** each device is defined by its hashrate H_k, energy consumption E_k, and set of supported mining algorithms A_k.

Available cryptocurrencies $l \in L$**:** each cryptocurrency is associated with a mining algorithm a_l. Thus, a cryptocurrency l can be mined by device k only if $a_l \in A_k$.

Mining pools $m \in M$**:** for each cryptocurrency l there is a set of pools M_l that support it. Each pool charges a fee f_m and offers a set of reward systems $N_{l,m}$ compatible with cryptocurrency l in pool m.

Network and economic parameters: for each cryptocurrency l we consider block reward R_l, global network hashrate H_l, and network difficulty D_l. The parameter $P^{(E)}$ denotes the electricity price and T the operating time (e.g. T = 86 400 seconds for one day).

Reward estimation: Payout$_{l,m,n}$ is the estimated gross reward that device k would receive when mining cryptocurrency l in pool m under reward system n.

Decision variable:

$$x_{k,l,m,n} = \begin{cases} 1, & \text{if device } k \text{ mines cryptocurrency } l \text{ in pool } m \text{ under reward system } n, \\ 0, & \text{otherwise.} \end{cases}$$

Objective function

$$\max_{x_{k,l,m,n}} \sum_{k\in K} \sum_{l\in L} \sum_{m\in M_l} \sum_{n\in N_{l,m}} x_{l,m,n} \left(\text{Payout}_{k,l,m,n}(1 - f_m) - \frac{E_k\, P^{(E)}\, T}{1000} \right) \tag{4.13}$$

where the term $\frac{E_k\, P^{(E)}\, T}{1000}$ represents the operative costs due to energy consumption kWh.

Constraints

$$x_{k,l,m,n} \in \{0,1\}, \quad \forall\, k \in K,\ \forall\, l \in L,\ \forall\, m \in M_l,\ \forall\, n \in N_{l,m}, \tag{4.14}$$

$$\sum_{l\in L} \sum_{m\in M_l} \sum_{n\in N_{l,m}} x_{k,l,m,n} = 1, \quad \forall\, k \in K \quad \text{(each device selects exactly one configuration)} \tag{4.15}$$

$$x_{k,l,m,n} \le \delta_{k,l}, \quad \forall\, k,l,m,n \quad \text{(device–cryptocurrency compatibility)}, \tag{4.16}$$

$$x_{k,l,m,n} \le \gamma_{l,m}, \quad \forall\, k,l,m,n \quad \text{(cryptocurrency–pool compatibility)}, \tag{4.17}$$

$$x_{k,l,m,n} \le \theta_{l,m,n}, \quad \forall\, k,l,m,n \quad \text{(pool–reward-system compatibility)}. \tag{4.18}$$

Restriction Compatibility

- **Device–cryptocurrency compatibility.** Let A_k be the set of mining algorithms supported by device k, and let a_l denote the mining algorithm of cryptocurrency l. The parameter $\delta_{k,l}$ is defined as:

$$\delta_{k,l} \in \{0,1\}, \quad \forall k \in K,\ \forall l \in L,$$

$$\delta_{k,l} = \begin{cases} 1, & \text{if } a_l \in A_k, \\ 0, & \text{otherwise.} \end{cases}$$

- **Cryptocurrency–pool compatibility.** Let $M_l \subseteq M$ be the set of mining pools that support cryptocurrency l. The parameter $\gamma_{l,m}$ is defined as:

$$\gamma_{l,m} \in \{0,1\}, \quad \forall l \in L,\ \forall m \in M,$$

$$\gamma_{l,m} = \begin{cases} 1, & \text{if } m \in M_l, \\ 0, & \text{otherwise.} \end{cases}$$

- **Pool–reward system compatibility.** Let $N_{l,m} \subseteq N$ be the set of reward systems available for cryptocurrency l in pool m. The parameter $\theta_{l,m,n}$ is defined as:

$$\theta_{l,m,n} \in \{0,1\}, \quad \forall l \in L,\ \forall m \in M,\ \forall n \in N,$$

$$\theta_{l,m,n} = \begin{cases} 1, & \text{if } n \in N_{l,m}, \\ 0, & \text{otherwise.} \end{cases}$$

Chapter 5

Implementation, Simulation and Validation

5.1 Implementation

This chapter presents the implementation of the proposed framework for rational mining, detailing how components were translated from conceptual model into a functional system. The implementation integrates data extraction, relational database design and estimation of pool rewards and the construction a linear programming model supporting miners to select the most profitable mining configuration. The following subsections explain the tools used, the data collection and organization, the database design, and the implementation of the optimization model.

5.1.1 Tools and Environment

Python[1] is the programming language selected to develop the framework due to its ecosystem designed for data manipulation. The implementation consists of the following steps:

- Data Extraction: Collect real-time information from blockchain networks and mining pools through APIs(Miningpoolstats), including dynamic network parameters such as difficulty, global hashrate, block reward and cryptocurrency price.
- Data Storage: Stores previously obtained information in databases with logic that detects errors that could affect calculations.
- Optimization Phase: Execute the linear programming model delivering the most convenient con-

[1] Python

figuration of device, cryptocurrency, pool and reward system based on recollected data and defined constraints.

The logic and structure of the database is modelled using the library SQLAlchemy[2], as it allows to work effectively with relational schemas maintaining compatibility with SQL. PuLP[3] is used for the execution of the linear programming model because of its simplicity for defining optimization problems and contraints.

5.1.2 Data Collection

This framework require real-time information which will be provided by APIs mentioned at the end of this section. Table 5.1 details organize data obtained in category, variable, temporal behavior and data source.

Variables hashrate, Energy consumption, and supported algorithms belong to the Device Category and they are defined as static because they do not change over time. The compatibility between mining algorithms and cryptocurrencies are also considered static. Each of the previously mentioned variables must be manually provided by the user of the framework.

Within the network category, difficulty, global hashrate, and price are treated as temporal variables since they fluctuate over time and are retrieved through external APIs. In contrast, block reward and transaction fees are derived through estimations based on available data and the formulas presented in the previous section.

Regarding the Pool category, the pool fee is a static value and the Pool hashrate can be obtained through an API. The number of solved blocks in the lapse of 24 hours and sharerate are important estimations for computing payouts of different reward systems. These values are obtained by applying the formulas described along supositions and actual data.

To acquire real-time network and market parameters, **Miningpoolstats**[4] was selected as an API to extract information because it provides difficulty, price and Hashrate for numerous popular cryptocurrencies. In addition, it contains relevant information of Pools such as pool Hashrate, pool fee and adopted reward system.

[2] SQLAlchemy
[3] PuLP
[4] Miningpoolstats

Table 5.1: Category, Variables and Sources used in the Implementation

Category	Variable	Temporal/Static	Source
Device	Hashrate	Static	Manual Input
	Energy Consumption	Static	Manual Input
	Algorithms-supported	Static	Manual Input
Algorithm	Algorithm-Cryptocurrency	Static	Manual Input
Network	Difficulty	Temporal	API
	Global Hashrate	Temporal	API
	Price	Temporal	API
	Block Reward	Static	Manual Input
	Transaction Fees	Temporal	Estimation
Pool	Fee	Static	Manual Input
	Pool Hashrate	Temporal	API
	Solved Blocks	Temporal	Estimation
	Sharerate	Temporal	Estimation
	Reward System	Static	API

5.1.3 Database Design

A relational database is used as the core structure of the framework. It supports the interactions between devices, mining algorithms, cryptocurrencies, mining pools, reward systems and network parameters. The database's design purposely include constraints to avoid values out of range. Figure 5.1 illustrates the database schema implemented for this work, showing entities and their attributes. Because the framework requires to integrate multiple components of mining such as devices, mining algorithms, cryptocurrencies, pools, and reward systems, a relational database is suitable option. These components can be modelled using primary/foreign keys and there are defined relationships between them.

This structure allows the framework to enforce compatibility contraints, represent relationships and integrate static and dynamic parameters. The Network table stores real-time parameters, while Estimations contains computed values required for reward calculations. Compatibility contraints are enforced through table Compatibility.

Devices and mining algorithms are linked through the device-algorithm table, which specifies the hashrate and energy consumption associated with each supported algorithm. The compatibility table defines the relationship between algorithms and cryptocurrencies, ensuring that each cryptocurrency is associated with its corresponding mining algorithm. The network table stores dynamic parameters for each cryptocurrency, such as difficulty, global hashrate, and price. Finally, the $pool_c rypto_r eward$ table models the relationship between pools, cryptocurrencies, and available reward systems, while also storing operational attributes such as the pool fee and pool hashrate. Snapshot represents a 'snapshot' of each configuration, containing the computed payout, cost, and other relevant data.

5.1.4 Implementation of Linear Programming Model

The implementation of the linear programming model detailed in Chapter 4 is carried out using the Python library PuLP[5], which is specifically designed for the formulating and solution of linear programming models. The interface of PuLP allows the model to be defined easily, including the incorporation of the objective function and constraints. The code is presented in Appendix, Listing 2.

Each feasible mining configuration stored in the snapshot table is associated with a binary decision variable in PuLP, allowing the model to determine whether that configuration is selected or not.

Due to the design of the Database, contraints approach the compatibility between devices, mining algorithms, cryptocurrencies, pool and reward systems. The implemented optimization model does not require the associated restrictions to compatibility, instead it requires only the constraints that restricts the selection to at most one configuration per device and enforcing the binary nature of the decision

[5] PuLP

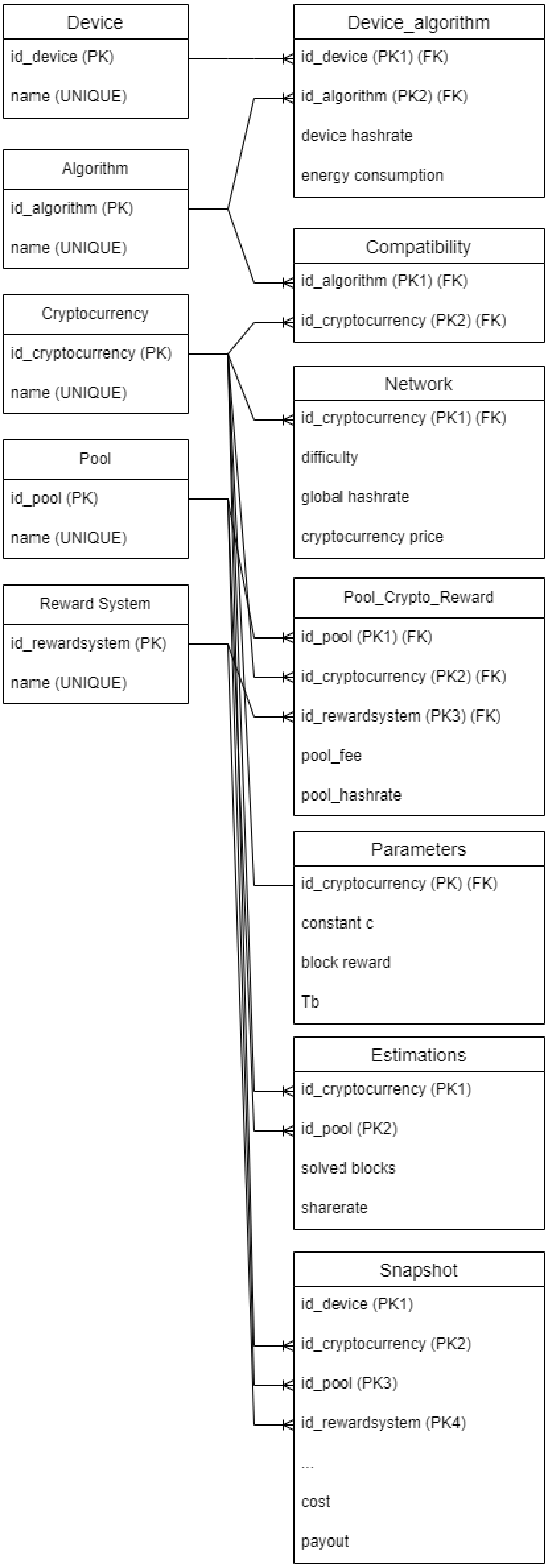

Figure 5.1: Database schema implemented for the framework

variables.

5.2 Simulation

5.2.1 Base Scenario

To evaluate the behaviour of the implemented framework, it is defined a controlled simulation environment. This scenery includes a representative subset of 4 mining configuration containing: Device, Cryptocurrency, Pool, reward systems and variables required to calculate the payout. The goal of the simulation is to observe how the model select the optimal combination for each device under supervised conditions. Each value of combinations was defined using values that are consistent with those typically observed in real-world mining environments.

Table 5.2 present parameters used in four mining configurations using 2 devices.

Table 5.2: Input of the Base Simulation

Parameters	Configuration 1	Configuration 2	Configuration 3	Configuration 4
Device	L7 9050	L7 9050	U3	U3
Cryptocurrency	Dogecoin	Litecoin	Bitcoincash	Bitcoin
Pool	f2pool	mining-dutch.nl	viabtc.com	antpool.com
Reward System	PPS	PROP	FPPS	PPS
Block Reward	10,000	6.25	3.125	3.125
Fee	0.04	0.01	0.01	0.04
Device Hashrate	9.05e9	9.05e9	1.16e+15	1.16e+15
Pool Hashrate	1.10e15	4.82e12	2.86e18	3.12e20
Global Hashrate	3.5e15	3.17e15	6.74e18	1.20e21
Network Difficulty	4.14e7	1.22e8	9.22e11	1.49e14
Constant c	2e32	2e32	2e32	2e32
Energy Consumption	3,425	3,425	11,020	11,020
Coin Price	0.15	87.12	589.62	9.41e4
Profit	-5.12	-11.32	9.66	7.19

Framework is executed using the parameters mentioned in Table 5.2. The model calculated the expected profitability of each device considering payout and operational costs. Table 5.3 summarizes the optimal combination selecting two devices and showing resultant profitability.

Table 5.3: Output of the base simulation

Output	Configuration 3	Configuration 1
Device	U3	L7 9050
Cryptocurrency	Bitcoincash	Dogecoin
Pool	viabtc.com	f2pool
Reward System	FPPS	PPS
Payout	46.68	6.38
Cost	37.02	11.5
Profit	9.66	-5.12

As shown in Table 5.3, the model successfully identifies the most profitable configuration for each device. The corresponding output of the linear programming model is presented below ing Listing 5.1:

Listing 5.1: Linear programming model output

```
Optimal configuration per device
  id_device device_name id_cryptocurrency cryptocurrency_name id_pool
3 3 U3 2 bitcoincash 62
1 1 L7 - 9050 52 dogecoin 34

   pool_name id_rewardsystem rewardsystem_name fee payout cost
3 viabtc.com 4 FPPS 0.01 46.687973 37.0272
1 f2pool.com 2 PPS 0.04 6.387226 11.5080

    profit
3 9.660773
1 -5.120774
Output:
Device n | device name | cryptocurrency | pool | reward system | payout | cost |
    profit_24H
Device 3 | U3 | bitcoincash | viabtc.com | FPPS | 46.69 | 37.03 | 9.66 USD.
Device 1 | L7 - 9050 | dogecoin | f2pool.com | PPS | 6.39 | 11.51 | -5.12 USD.
Optimal profit: 4.54 USD.
Solution time: 0.74 seconds
```

As observed, the implemented model selected the configurations that maximizes the profitability according to both devices. These results confirm that Framework is capable of compare different mining configurations, estimating profitability according to their parameters and identifies the configuration that maximizes profit considering multiple devices. Therefore, this scenario validates the correct functioning of the framework under controlled conditions.

5.2.2 Proof of Optimality

This subsection presents a formal proof of optimality for the proposed decision model. The aim is to demonstrate that the proposed model consistently selects optimal mining configuracions according to the defined objective function.

The decision model is formulated a binary integer linear programming problem where each device requires to select one configuration fom a set of feasible alternatives. The objective function maximizes profitability, and the profitability is defined as the sum of the individual profit from selected mining configurations.

Because the total profit is computed as the sum of the individual profits of each device, and because the selection constraints apply independently to each device, the optimization problem can be transformed in independent subproblems, one per device.

This separability allows that the optimal solution can be derived by applying an argmax operation over the set of mining configurations for each device. A controlled experiment was conducted using four mining confugrations and two solutions were compared: The solution obtained using the solver and the analytical solution obtained by selecting the maximum profit configuration for each device.

```
id_row   id_device device_name cryptocurrency_name        pool_name
0        0           L7 - 9050            litecoin  mining-dutch.nl
1        1           L7 - 9050            dogecoin       f2pool.com
2        2           U3             bitcoin      antpool.com
3        3           U3         bitcoincash       viabtc.com

rewardsystem_name      profit
PROP                -11.328539
PPS                 -5.120774
PPS                  7.193352
FPPS                 9.198516

=== MILP solution ===
Solver status: Optimal
```

```
device_name   cryptocurrency_name   pool_name    rewardsystem_name profit
U3            bitcoincash           viabtc.com   FPPS 9.198516
L7 - 9050     dogecoin              f2pool.com   PPS -5.120774

MILP optimal profit: 4.077742
Solution time: 0.2248 seconds
```

The results show that both approaches produce identical decisions for all devices. In addition, the solver reports an Optimal status, confirming global optimality with respect to the formulated model.

5.3 Validation

The previous section presented a simulation considering four configurations across two mining devices, where the proposed framework successfully computed profitability and selected the most profitable configuration. In the present section, we extend the validation process by first evaluating the framework profitability estimation through a comparison with existing academic work and a real mining experiment. Additionally, a functional validation of the framework is provided to demonstrate its correct operational behavior.

5.3.1 Validation of profitability estimation

The proposed framework for rational mining is validated by comparing its estimated profitability under equitative conditions against the ProfitMax model (Cho et al. (2024)), which provides a decision-making framework based on minimizing operational costs using real-time information and focusing primarily on energy consumption. However, Profitmax does not consider mining pools and the impact of reward systems, factors that can significantly affect profitability depending on network conditions. The proposed framework complement this gap integrating mining pools and principal reward systems (Proportional, PPS and FPPS) expanding the scope of profitability analysis. In addittion to this academic comparison, the validation includes an empirical mining experiment for assessing the model's accuracy.
We evaluate profitability estimation across four scenarios under a baseline that imposes equitative conditions with respect to mining pools and reward systems: (1) an implementation of (Xu et al. (2020)) formulas, (2) an implementation of the academic ProfitMax model, (3) our proposed framework, and (4) a real mining operation using the configurations presented. All scenarios share the same key parameters, including a fixed electricity cost of 0.14 USD/kWh and the use of the same mining device. Specifically,

parameters associated with pools and reward mechanisms are neutralized (pool fees are set to zero) and an NVIDIA RTX 4060 GPU was employed in all experiments to ensure consistency and comparability across methods.

The purpose of this comparison is to illustrate how closely the profitability estimated using the proposed framework aligns with the academic model Profitmax and real mining environment under mentioned conditions. We can measure the accuracy of the framework through contrasting the results produced with each method. This test is essential to determine wheter the proposed model can reliable reproduce real-world profitability results and effectively helping miners in his decision-making through measuring profitability of different combinations.

In addition to the comparison with ProfitMax and the empirical mining experiment, the proposed framework is also contrasted with the analytical model introduced by (Xu et al. (2020)), which studies miners pool selection strategies under different reward mechanisms. For this comparison, an adapted implementation of the model proposed by Xu is developed using the same network, device, and cost parameters employed in the previous scenarios. Unlike the original formulation, which assumes homogeneous hashrate among miners within each pool, this validation relaxes such an assumption and instead uses measured hashrate values, reflecting the heterogeneous nature of real-world mining environments.

For the observed scenario, a real mining experiment was conducted using Awesome Miner as a monitoring tool. XMRIG Miner 6.24.0 was employed as the mining software for mining Zephyr, while Gminer 3.44 and CcMiner were used for mining Ravencoin and Groestlcoin, respectively. The mining process was executed for a continous 24-hour period accounting the valid shares submitted to the pool, the payout was computed using the PPS reward formula.

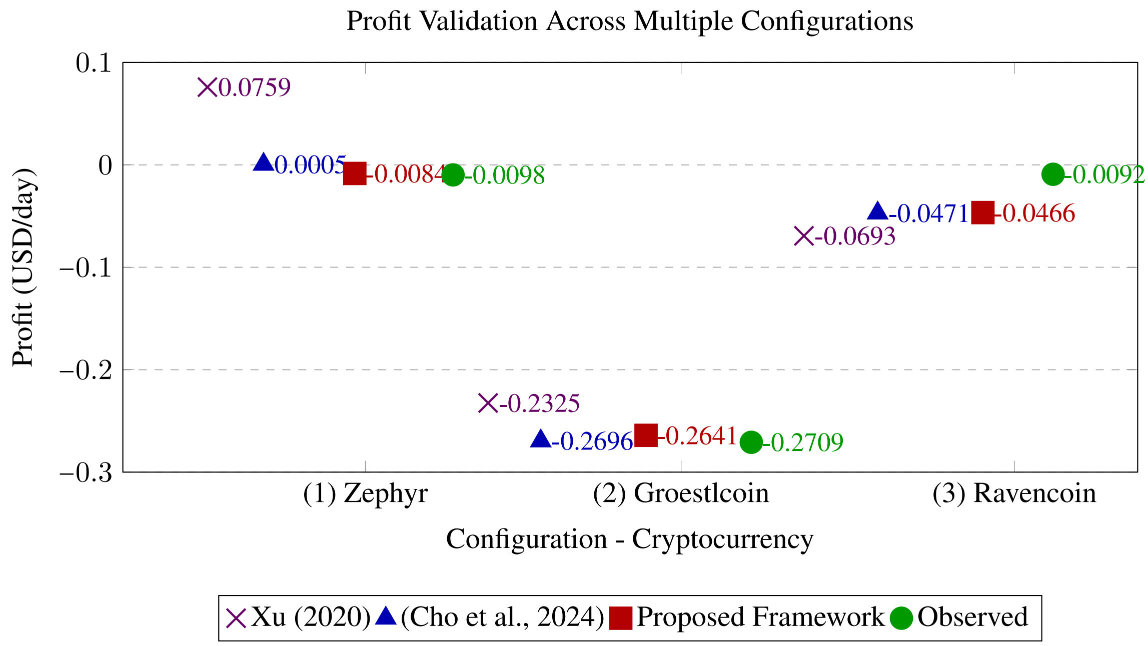

Figure 5.2: Comparison of profit across ProfitMax, Proposed Framework, Xu (2020), and Observed values.

Figure 5.2 illustrates the profitability obtained across three cryptocurrencies under the four escenarios: Xu, ProfitMax, the proposed Framework and the observed mining experiment. It is possible to observe that each configuration reaches similar values of profitability when using Xu formulas, the academic model Profitmax, our proposed framework and the real experiment. This consistency indicates that the Framework accurately provides the expected profitability under controlled conditions and therefore it can be used as a effective tool for decision-making in real mining environments.

Under this adapted implementation, the profitability values estimated by the Xu (2020) model are 0.0759 USD/day for Zephyr, -0.2325 USD/day for Groestlcoin, and -0.0693 USD/day for Ravencoin. As illustrated in Figure 5.2, these estimates deviate more significantly from the observed profitability compared to the proposed framework, particularly in the Zephyr scenario where the model overestimates profitability. This can be due to the theoretical nature of the framework and simplifying assumptions that are not fully aligned with real mining behavior.

The comparison accross mining Zephyr, Groestlcoin and Ravencoin cryptocurrencies between Cho, Proposed Framework and Observed with current device shows that profitability remains the same order of magnitude for each case. Because the profitability values are very small, with an order 10^{-2} , relative error becomes disproportionately large and it does not represent the similarity observed. For this reason, the evaluation is made using absolute error. The absolute error in profitability between Observed and proposed framework are 0.0014, 0.0068 and 0.0374 USD/Day for Zephyr, Groestlcoin and Ravencoin, respectively. These values repreent small deviations on the order of magnitude of 10^{-2} USD/Day. Such small difference indicates that framework is capable of accurately reproducing real mining profitability under controlled conditions. It is important to notice that although similar values are obtained employing the framework compared to Profitmax, the advantages of the proposed framework are the following:

- It explicitly incorporates mining pools and reward systems, providing a more comprehensive framework that better reflects real-world mining practices, where the majority of Proof-of-Work mining activity is conducted through pools.
- For short-term, the framework can provide more accurate profitability estimates by explicitly modeling specific pool parameters and reward mechanisms, integrating real-time network conditions of mining pools.

5.3.2 Functional Validation of the Framework

A functional evaluation was conducted to verify that the framework operates adequately. This evaluation executes the implemented framework using real mining hardware specifications and a large dataset of real-time data containing cryptocurrencies and mining pools. The devices selected for the evaluation consist of two ASIC miners, the L7-9050 and the U3S23H, with hashrates of 9.05 GH/s and 1.16 PH/s and energy consumptions of 3425 W and 11 020 W, respectively. Framework processed data containing 342 cryptocurrencies and 291 mining pools, generating several feasible mining configurations and select the optimal for each device. The results of this execution are shown in Figure 5.2. The framework identified 77 valid mining configurations between the two devices and computed the optimal profitability for each device using the linear programming model. For the device 'L7-9050' the configuration that provides most profitability was mining Luckycoin on the pool cloverpool with the FPPS reward system, resulting in a profit of -7.71 USD. For the second device 'U3S23H', an adequate configuration was mining ecash in the pool mining-dutch.nl using the proportional reward system, producing a profit of 10.14 USD. Listing 5.2 presents the output when running the framework for the devices mentioned.

Listing 5.2: Linear programming model output

```
Number of combinations: 50
Number of distinct devices: 2
Combinations per device:
id_device
1 26
2 24
Name: count, dtype: int64

Optimal configuration per device
```

```
    id_device device_name id_cryptocurrency cryptocurrency_name id_pool
600 2 U3S23H 4 ecash 29
60 1 L7 - 9050 52 dogecoin 33

         pool_name id_rewardsystem rewardsystem_name fee payout
600 pool.kryptex.com 3 PPS+ 0.02 47.164717
60 cloverpool.com 4 FPPS 0.00 3.794039

      cost profit
600 37.0272 10.137517
60 11.5080 -7.713961
Output:
Device n | device name | cryptocurrency | pool | reward system | payout | cost |
    profit_24H
Device 2 | U3S23H | ecash | pool.kryptex.com | PPS+ | 47.16 | 37.03 | 10.14 USD.
Device 1 | L7 - 9050 | dogecoin | cloverpool.com | FPPS | 3.79 | 11.51 | -7.71 USD.
Optimal profit: 2.42 USD.
Solution time: 0.37 seconds
```

These results demonstrate that the Framework is capable to estimate profitability using real-time data and select the optimal configuration for the considered devices.

Chapter 6

Results and Discussion

6.1 Results and Discussion

The results presented in Figure 6.1 provides a comparative analysis of profitability estimation across four mining configurations using three sources: The proposed framework, a real mining execution and external online calculators.

The experimental evaluation is conducted using two mining devices: an Intel Core i5-13500 CPU and an NVIDIA RTX 4060 GPU. The proposed framework is executed and the two most profitable mining configurations for each device are displayed in the Figure 6.1, namely Monero, Zephyr, Ravencoin and Neoxa.

Monero and Zephyr are CPU-based configurations using the RandomX mining algorithm. All the three sources consistently reported negative profitability values and the estimations remain close to the real measurement, with deviations of 0.0058 USD/day and 0.00784 USD/day between the proposed Framework and the real mining execution for Monero and Zephyr, respectively.

In contrast, Ravencoin and Neoxa are GPU-based configurations relying on the KawPow mining algorithm. For these cryptocurrencies, the deviation between the framework estimates and the real mining execution are 0.01056 and 0.04290 USD/day, respectively. These values are considered within an acceptable range considering a volatile environment and the variability of mining performance.

Regarding the expected profitability reported by external mining calculators, data were obtained from mining platforms such as CryptoCompare, WhatToMine, and Minerstat. The profitability values provided by these calculators are similar to those estimated by the proposed Framework, supporting the validity of the framework. It is important to note that real mining environments are subject to fluctuations in network difficulty, cryptocurrency prices and mining pool conditions. For this reason, the values obtained should be interpreted as estimations rather than exact measurements.

Estimations provided by the the proposed Framework, real execution and external calculators are equivalent, with a difference that the Framework explicitly incorporates mining pool reward systems and pool fees into the profitability calculations, whereas external calculators typically rely on simplified assumptions based on solo mining. The figure below illustrates a comparison of profitability accross mining configurations.

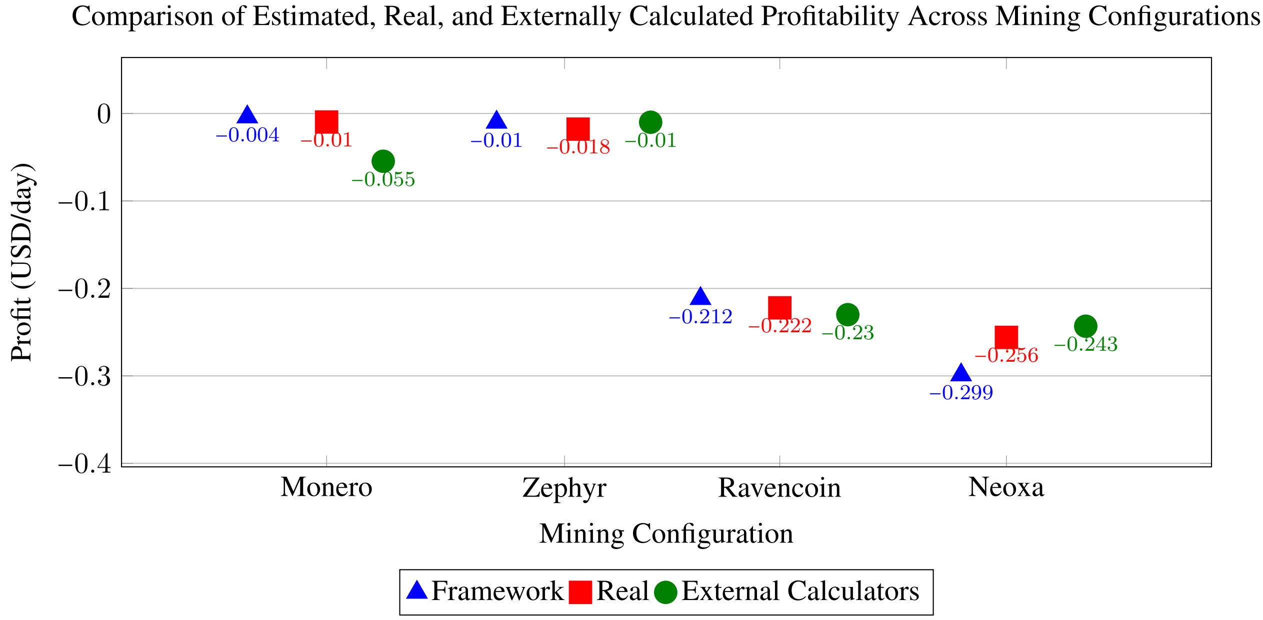

Figure 6.1: Comparison of profitability across mining configurations as estimated by the framework, observed in real mining environment, and reported by external calculators

The absolute estimation error obtained by the proposed framework and external mining calculators for each configuration is presented below. Figure 6.2 provides a visual comparison of both approaches.

Table 6.1: Absolute Estimation Error of the Proposed Framework and External Mining Calculators Compared to Real Mining Execution

Cryptocurrency	Absolute Error (Framework)	Absolute Error (Calculator)
Monero	0.00580	0.04484
Zephyr	0.00784	0.00790
Ravencoin	0.01056	0.00780
Neoxa	0.04290	0.01280

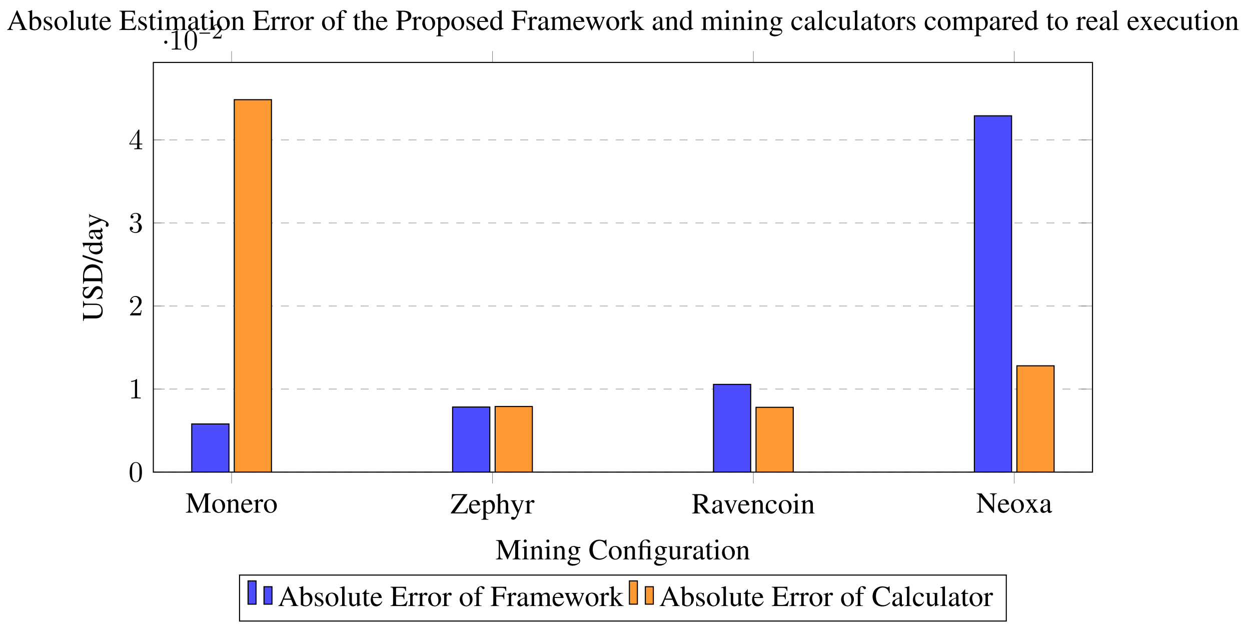

Figure 6.2: Comparison of absolute prediction error between the proposed framework and external mining calculators across four mining configurations.

According to Table 6.1 and Figure 6.2, the proposed framework shows consistently low estimation error across the evaluated mining configurations. In three of the four configurations, the absolute error remains below USD 0.01 per day, indicating appropriate alignment with real mining execution. In the remaining configuration, a deviation of approximately USD 0.04 per day is observed. This deviation may be influenced by multiple factors, including the highly volatile nature of network conditions and natural variability in share submission during the mining process. With respect to external mining calculators, similar deviations with respect to real mining execution are observed in most cases, except for the Monero configuration, where the calculator exhibits a deviation of approximately USD 0.04 per day compared to USD 0.0058 per day obtained by the proposed framework. Although the profitability estimations provided by the framework and external calculators are generally similar, their overall predictive performance is also comparable, but the proposed framework offers the advantage of evaluating a broader set of configuration options, which may lead to the selection of more profitable mining strategies.

Listing 6.1 presents the results of the framework execution for the two selected mining devices. As defined in the experimental setup, the Intel Core i5-13500 is restricted to mining cryptocurrencies based on the RandomX algorithm, whereas the NVIDIA RTX 4060 is restricted to KawPow-based cryptocurrencies. Under these constraints, the framework evaluates a total of 13 feasible mining configurations, considering hardware specifications, compatible cryptocurrencies, available mining pools and reward systems, and current network conditions.

Table 6.2: Optimal mining configurations identified by the proposed framework

Output	Configuration 1	Configuration 3
Device	I5 13500	RTX 4060
Cryptocurrency	Monero	Ravencoin
Pool	ntminerpool.com	ntminerpool.com
Reward System	PPS+	PPS+
Payout	0.052 USD/day	0.099 USD/day
Cost	0.054 USD/day	0.299 USD/day
Profit	-0.002 USD/day	-0.200 USD/day

The optimal mining configurations identified by the proposed framework for each device are summarized in the table 6.2. For the CPU I5-13500, mining the cryptocurrency Monero in the pool ntminerpool.com with PPS+ system reward is selected as the optimal option, reaching an estimated payout of 0.052 USD/day, an operation cost of 0.054 USD/day, and a resulting profit of -0.002 USD/day. For the RTX 4060, the Framework identifies mining Ravencoin in the ntminerpool.com, also under the PPS+ reward system as the optimal configuration. In the last case, the estimated payout, operational cost and profit are 0.099, 0.299 and -0.200 USD/day.

Listing 6.1: Output of the framework execution for two mining devices

```
Number of combinations: 13
Number of distinct devices: 2
Combinations per device:
id_device
2 8
1 5
Name: count, dtype: int64

Optimal configuration per device
    id_device device_name id_cryptocurrency cryptocurrency_name id_pool \
26 1 I5 13500 254 monero 26
165 2 RTX 4060 183 ravencoin 26

        pool_name id_rewardsystem rewardsystem_name payout \
26 ntminerpool.com 3 PPS+ 0.051719
165 ntminerpool.com 3 PPS+ 0.098999
```

```
      cost profit
26 0.05376 -0.002041
165 0.29904 -0.200041
Output:
Device n | device name | cryptocurrency | pool | reward system | payout | cost |
    profit_24H
Device 1 | I5 13500 | monero | ntminerpool.com | PPS+ | 0.052 | 0.054 | -0.002 USD.
Device 2 | RTX 4060 | ravencoin | ntminerpool.com | PPS+ | 0.099 | 0.299 | -0.200
    USD.
Optimal profit: -0.20 USD.
Solution time: 0.03 seconds
```

Under the evaluated conditions, the expected profit estimated by the proposed framework for both devices is approximately -0.20 USD/day. In comparison, the profit obtained from the real mining execution is -0.232 USD/day, indicating that the results are consistent.

Overall, the presented results demonstrates that the Framework provides reliable profitability estimations while considering hardware devices, cryptocurrencies, mining pools and reward systems.

Chapter 7

Conclusions

This thesis addresses the problem of rational decision-making in Proof-of-Work blockchain mining, where profitability depends on a combination of hardware specifications, cryptocurrency characteristics, mining pool conditions, and reward systems. Given the dynamic nature of these factors, identifying the most profitable mining configuration is a non-trivial task. Existing academic models and online mining calculators often simplify or overlook the role of mining pools and reward systems. To address this gap, this work proposes a framework to support decisions that integrates hardware devices, cryptocurrencies, mining pools, reward systems, and dynamic blockchain network conditions to assist miners in making rational decisions respect to the mining configuration.

The research question in this thesis: *How can a rational mining framework be designed to identify the most profitable mining configuration in Proof-of-Work blockchain networks under dynamic conditions?* has been succesfully answered. The results of the proposed framework indicates that it is capable to identify the optimal mining configurations integrating real-time blockchain data, mining pools, reward systems formulations and a binary linear programming model.

The main contributions of this work include the proposal of a framework for rational mining that integrates hardware devices, cryptocurrencies, mining pools, and reward systems. A prototype of the proposed framework was implemented, incorporating real-time blockchain data and theoretical reward system formulations. In addition, an optimization model was developed to identify the optimal mining configuration under given network conditions. The framework was validated through comparisons with existing academic models and empirical evaluation in a real mining environment.

The results of the implementation show that the profitability estimations produced by the proposed framework is closely aligned with the real mining experiment, with an deviation of approximately 0.005 USD/day for the evaluated mining configuration. The framework succesfully identified the rational mining option among the 13 available configurations.

Since a CPU and GPU were used in the experimental mining execution, the resulting profitability was negative for both devices with a estimated profit of -0.20 USD/day. This result is consistent with the estimation provided by the framework at -0.232 USD/day, an expected outcome as CPU and GPU devices are generally not profitable under current network conditions.

With respect to predictive capability, the proposed framework achieves results comparable to those obtained from external mining calculators. Four mining configurations were evaluated, of which three exhibited an absolute deviation below USD 0.01 per day with respect to real mining execution, while the remaining configuration showed a deviation of approximately USD 0.04 per day. These results indicate that the framework is capable of providing profitability estimations with accuracy on the order of a few cents per day under the evaluated experimental conditions. Although the results obtained using the framework and external calculators are generally similar, the proposed framework offers an additional advantage by allowing the evaluation of a broader set of cryptocurrencies and mining pools, which may lead to the identification of more profitable mining configurations. However, such configurations were not observed under the specific experimental conditions considered in this study, mainly due to the limited mining capabilities of the employed devices, which were restricted to only two mining algorithms.

This work presents several limitations that should be considered when interpreting the results. Profitability estimations produced by the proposed framework are based on short-term network data collected over a 24-hour period, and extreme scenarios involving abrupt price fluctuations may affect the resulting decisions. Additionally, the framework assumes stable network conditions throughout the entire 24-hour mining period. A potential improvement would be to consider shorter time intervals within this period, capturing variations in network conditions at finer temporal resolutions and integrating them into the overall profitability estimation, rather than assuming constant conditions over the full 24-hour window.

Future work may extend the framework by incorporating others reward systems such as PPLNS and integrating variance in the decision-making process.

Overall, this thesis provides a foundation for mining decision-making in PoW blockchain networks by integrating hardware devices, cryptocurrencies, mining pools and reward systems. Overall, this thesis provides a practical and extensible foundation for rational mining decision-making in Proof-of-Work blockchain networks, integrating hardware devices, cryptocurrencies, mining pools, reward systems and real-time information within a unified framework.

Bibliography

Albrecher, H., Finger, D., and Goffard, P.-O. (2023). Empirical risk analysis of mining a proof-of-work blockchain. *Decisions in Economics and Finance*.

Bard, D. A., Kearney, J. J., and Perez-Delgado, C. A. (2022). Quantum advantage on proof of work. *Array*, 15:100225.

Bedford Taylor, M. (2017). The evolution of bitcoin hardware. *Computer*, 50(9):58–66.

Bistarelli, S., Giuliani, G., Mercanti, I., and Santini, F. (2022). An in-depth analysis of mining pools revenue.

Cho, D., Ramachandran, G., Jurdak, R., and Kanhere, S. (2024). Profitmax: Optimizing blockchain mining for energy efficiency and profitability. In *2024 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 25–30.

Delgado-Mohatar, O., Felis-Rota, M., and Fernández-Herraiz, C. (2019). The bitcoin mining breakdown: Is mining still profitable? *Economics Letters*, 184:108492.

Derks, J., Gordijn, J., and Siegmann, A. (2018). From chaining blocks to breaking even: A study on the profitability of bitcoin mining from 2012 to 2016. *Electronic Markets*, 28(3):321–338.

Farooq, U., Marrakchi, Z., and Mehrez, H. (2012). *FPGA Architectures: An Overview*, pages 7–48. Springer New York, New York, NY.

Khosravi, A. and Säämäki, F. (2023). Beyond Bitcoin: Evaluating Energy Consumption and Environmental Impact across Cryptocurrency Projects. *Energies*, 16(18):1–23.

Malone, D. and O'Dwyer, K. (2014). Bitcoin mining and its energy footprint. pages 280–285.

Nakamoto, S. (2009). Bitcoin : A peer-to-peer electronic cash system.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart,

L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., and Moher, D. (2021). The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372.

Panuntun, L. H., Susanti, B. H., and Rosdiana, S. (2023). Performance test of hash functions in proof of work schemes in blockchain. In *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*, pages 47–52.

Rosenfeld, M. (2011). Analysis of bitcoin pooled mining reward systems.

Schrijvers, O., Bonneau, J., Boneh, D., and Roughgarden, T. (2017). Incentive compatibility of bitcoin mining pool reward functions. In Grossklags, J. and Preneel, B., editors, *Financial Cryptography and Data Security*, pages 477–498, Berlin, Heidelberg. Springer Berlin Heidelberg.

Soria, J., Moya, J., and Mohazab, A. (2023). Optimal mining in proof-of-work blockchain protocols. *Finance Research Letters*, 53:103610.

Taraate, V. (2022). *ASIC Design*, pages 403–410. Springer Singapore, Singapore.

Tripathi, G., Ahad, M. A., and Casalino, G. (2023). A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. *Decision Analytics Journal*, 9:100344.

Xu, C., Zhu, K., Wang, R., and Xu, Y. (2020). Dynamic selection of mining pool with different reward sharing strategy in blockchain networks. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6.

Zhu, S., Li, W., Li, H., Hu, C., and Cai, Z. (2018). A survey: Reward distribution mechanisms and withholding attacks in bitcoin pool mining. *Math. Found. Comput.*, 1(4):393–414.

Chapter 8

Appendix

.1 Literature Review

Table .1: Literature review of selected works

Author and Year	Title
Abdel-Basset et al., 2023	Optimizing jointly mining decision and resource allocation in a MEC-enabled blockchain networks
H. Abdulmonem et al., 2020	Hardware Acceleration of Dash Mining Using Dynamic Partial Reconfiguration on the ZYNQ Board
Abed et al., 2020	Enhanced SAT Solvers Based Hashing Method for Bitcoin Mining
Abed et al., 2021	An analysis and evaluation of lightweight hash functions for blockchain-based IoT devices
P. Adewumi et al., 2020	Inner For-Loop for Speeding up Blockchain Mining
Aggarwal et al., 2021	Architecture of blockchain
A. G. Agung et al., 2019	Proof of work: Energy inefficiency and profitability
Ahmad et al., 2024	Crypto Mining Data: Nature and Inferences
A. Ahmed Memon et al., 2018	Modeling of Blockchain Based Systems Using Queuing Theory Simulation
Ajwalia et al., 2025	Energy Performance, Carbon Footprint and Hardware Efficiency of Blockchain Networks
Akbarnavasi et al., 2023	An optimization strategy for enhancing energy consumption performance in digital currency miner's building
Alambardar Meybodi et al., 2025	Optimal blocks for maximizing the transaction fee revenue of Bitcoin miners
A. Alambardar Meybodi et al., 2022	Optimal Mining: Maximizing Bitcoin Miners' Revenues from Transaction Fees
Albrecher et al., 2022	Blockchain mining in pools: Analyzing the trade-off between profitability and ruin
Albrecher et al., 2024	Empirical risk analysis of mining a Proof-of-Work blockchain
Albrecher et al., 2022	On the Profitability of Selfish Blockchain Mining Under Consideration of Ruin
Albuquerque et al., 2023	Analyzing the Solo Mining Profitability of Zcash Cryptocurrency in the United States of America

(Continued on next page)

(Continued from previous page)

Author and Year	Title
A. Aljabr et al., 2019	Mining process in cryptocurrency using blockchain technology: Bitcoin as a case study
Kh Alkaeed et al., 2020	Highlight on cryptocurrencies mining with CPUs and GPUs and their benefits based on their characteristics
Almubarak et al., 2023	Energy Consumption and Power Quality in Bitcoin Mining Facilities in Texas
Alofi et al., 2022	Optimizing the Energy Consumption of Blockchain-Based Systems Using Evolutionary Algorithms: A New Problem Formulation
Alonso et al., 2021	Cryptocurrency Mining from an Economic and Environmental Perspective. Analysis of the Most and Least Sustainable Countries
Alshahrani et al., 2023	Sustainability in Blockchain: A Systematic Literature Review on Scalability and Power Consumption Issues
Anandhabalaji et al., 2023	Examining the Volatility of Conventional Cryptocurrencies and Sustainable Cryptocurrency during Covid-19: Based on Energy Consumption
A. Aponte-Novoa et al., 2021	The 51% Attack on Blockchains: A Mining Behavior Study
H. Austin et al., 2020	SharedWealth: Disincentivizing Mining Pools Through Burning and Minting
Basile et al., 2022	A Rational Mining Strategy for Proof-of-Work Consensus Algorithms
K. Battina et al., 2024	Secure Pool Mining through SVM-Based Miner Classification and Computation Validation in Blockchain Networks
Beer et al., 2022	A quick look at Cryptocurrency Mining: Proof of Work
Belotti et al., 2018	Bitcoin Pool-Hopping Detection
Belotti et al., 2020	Rewarding Miners: Bankruptcy Situations and Pooling Strategies
D. Bhaskar et al., 2015	Bitcoin Mining Technology
Biais et al., 2019	The Blockchain Folk Theorem
Bistarelli et al., 2024	An In-depth Analysis of Mining Pools Revenue
Bitir-Istrate et al., 2021	The transition towards an environmental sustainability for Cryptocurrency mining

(Continued on next page)

(Continued from previous page)

Author and Year	Title
Brown et al., 2024	How Do Cryptocurrency Miners Perceive the Ecological Implications of Their Work?
Y. Bukhari et al., 2024	Current and future implications of bitcoin mining on energy and climate change
Can et al., 2022	On reward sharing in blockchain mining pools
Cao et al., 2019	Application of ZD strategy in mining pool game
Capponi et al., 2023	Proof-of-Work Cryptocurrencies: Does Mining Technology Undermine Decentralization?
Y. Chang et al., 2020	Mining Power Misestimation in PoW Blockchain
Y. Chang et al., 2020	Dynamic power control for rational cryptocurrency mining
Chatterjee et al., 2019	Hybrid mining exploiting blockchain's computational power for distributed problem solving
Chatzigiannis et al., 2022	Diversification across mining pools: optimal mining strategies under PoW
J. G. Chavez et al., 2016	Automatic hopping among pools and distributed applications in the Bitcoin network
Chen et al., 2021	Impact of Temporary Fork on the Evolution of Mining Pools in Blockchain Networks: An Evolutionary Game Analysis
Chen et al., 2023	Pooling under the Sun: A Mining Pool Centralized Revisit and Solution
Chen et al., 2019	An axiomatic approach to block rewards
Chen et al., 2020	Decentralized mining pool games in blockchain
Cho et al., 2024	Cost-aware Blockchain Mining Strategy: Dynamic Energy Pricing (DEP) Model
Cho et al., 2024	ProfitMax: Optimizing Blockchain Mining for Energy Efficiency and Profitability
Cho et al., 2025	Bitcoin Halving Events: Historical Analysis and Strategic Insights for Miners
Ciaian et al., 2021	Interdependencies between mining costs, mining rewards and blockchain security

(Continued on next page)

(Continued from previous page)

Author and Year	Title
Cocco et al., 2016	Modeling and simulation of the economics of mining in the Bitcoin market
W. Cong et al., 2021	Decentralized Mining in Centralized Pools
Delgado et al., 2019	The Bitcoin mining breakdown: Is mining still profitable?
Derks et al., 2018	From chaining blocks to breaking even: A study on the profitability of bitcoin mining from 2012 to 2016
Deshmukh et al., 2023	A Survey on Blockchain and Cryptocurrency-Based Systems
A. Dev et al., 2014	Bitcoin mining acceleration and performance quantification
Dos Santos et al., 2019	An efficient miner strategy for selecting cryptocurrency transactions
Dos Santos et al., 2024	On the Impact of the Lightning Network on Bitcoin Transaction Fees and Network Value
D. Dos Santos et al., 2020	Candidate Set Formation Policy for Mining Pools
Göbel et al., 2016	Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay
Gudmundsson et al., 2024	Blockchain-based Decentralized Reward Sharing: The Case of Mining Pools
Hobbs et al., 2022	Algorithmic balancing of hashrate in a Proof-of-Work (PoW) consensus protocol
Y. Huang et al., 2018	Estimating Profitability of Alternative Cryptocurrencies (Short Paper)
Jiang et al., 2019	Bitcoin mining with transaction fees: A game on the block size
Jiang et al., 2021	Taming Propagation Delay and Fork Rate in Bitcoin Mining Network
Keyang et al., 2019	Poster: A novel mechanism for rewards distribution in pool mining of proof of work
E. Khairuddin et al., 2019	An exploration of bitcoin mining practices: Miners' trust challenges and motivations
Laneve et al., 2020	A formal analysis of the bitcoin protocol
Leelavimolsilp et al., 2019	Selfish Mining in Proof-of-Work Blockchain with Multiple Miners: An Empirical Evaluation
Lewenberg et al., 2015	Bitcoin mining pools: A cooperative game theoretic analysis

(Continued on next page)

(Continued from previous page)

Author and Year	Title
Liu et al., 2018	Evolutionary game for mining pool selection in blockchain networks
G. Motlagh et al., 2021	Analysis of Selfish Miner Behavior in the Bitcoin Network
B. Pandya et al., 2022	GPU and FPGA Based Deployment of Blockchain for Cryptocurrency - A Systematic Review
Poongodi et al., 2020	Bitcoin price prediction using ARIMA model
S. Puppala et al., 2023	Trust Model to Identify Reputed Miners in Blockchain Pool Mining
S. Puppala et al., 2024	Optimizing pool mining performance: A VIKOR-based model for identifying reputed miners in blockchain networks
S. Puppala et al., 2024	Strategic Miner Selection for Optimizing Block Generation Time in PoW-Based Blockchain Pool Mining Using SMNST Framework
Qin et al., 2018	Research on the selection strategies of blockchain mining pools
Qin et al., 2018	Optimal Share Reporting Strategies for Blockchain Miners in PPLNS Pools
Qin et al., 2019	A novel hybrid share reporting strategy for blockchain miners in PPLNS pools
Qin et al., 2018	Economic Issues in Bitcoin Mining and Blockchain Research
Raju et al., 2018	A Study of Current Cryptocurrency Systems
Ren et al., 2019	Pooled mining is driving blockchains toward centralized systems
Rosenfeld, 2012	Analysis of Bitcoin Pooled Mining Reward Systems
Roughgarden et al., 2021	Ignore the Extra Zeroes: Variance-Optimal Mining Pools
Sajjad et al., 2024	What Motivates Bitcoin Miners to Practice Bitcoin Mining: An Assessment Based on Behavioral Reasoning Theory
B. Santi et al., 2020	Solving Cryptographic Puzzles: How to Mine?
Smuseva et al., 2024	Selfish Mining in Public Blockchains: A Quantitative Analysis
Soria et al., 2023	Optimal mining in proof-of-work blockchain protocols
Tedeschi et al., 2024	Mining Profitability in Bitcoin: Calculations of User-Miner Equilibria and Cost of Mining
Tovanich et al., 2022	The Evolution of Mining Pools and Miners' Behaviors in the Bitcoin Blockchain

(Continued on next page)

(Continued from previous page)

Author and Year	Title
W. Wang et al., 2024	An Accurate Analytical Model for A Proof-of-Work Blockchain with Multiple Selfish Miners
M. Werner et al., 2020	PoolSim: A Discrete-Event Mining Pool Simulation Framework
Xing-Hong et al., 2022	Selection Strategy of Mining Pool under Various Different Payment Mechanisms
Xu et al., 2020	Dynamic Selection of Mining Pool with Different Reward Sharing Strategy in Blockchain Networks
Yaish et al., 2023	Correct Cryptocurrency ASIC Pricing: Are Miners Overpaying?
Zhang et al., 2023	Modeling & analysis of block generation process of the mining pool in blockchain system
Zhang et al., 2023	Achieving optimal rewards in cryptocurrency stubborn mining with state transition analysis
Zheng et al., 2020	Difficulty Prediction for Proof-of-Work Based Blockchains
Zolotavkin et al., 2019	Incentives for stable mining in pay per last n shares pools
B. Zur et al., 2020	Efficient MDP Analysis for Selfish-Mining in Blockchains

.2 Database Schema Implementation

Listing 1: SQLAlchemy database schema for the proposed framework

```
# sqlalchemy psycopg2-binary
from __future__ import annotations
from datetime import datetime
from typing import List, Optional

from sqlalchemy import (
   create_engine, MetaData, CheckConstraint, ForeignKeyConstraint,
   PrimaryKeyConstraint, UniqueConstraint, Index
)
from sqlalchemy.orm import (
   DeclarativeBase, Mapped, mapped_column, relationship, Session
)
from sqlalchemy import Integer, Text, Numeric, DateTime
```

```
DATABASE_URL = "sqlite:///pow.db"
engine = create_engine(DATABASE_URL, echo=False, future=True)
naming_convention = {
    "pk": "pk_%(table_name)s",
    "fk": "fk_%(table_name)s__%(column_0_name)s__%(referred_table_name)s",
    "ix": "ix_%(table_name)s__%(column_0_name)s",
    "uq": "uq_%(table_name)s__%(column_0_name)s",
    "ck": "ck_%(table_name)s__%(constraint_name)s",
}

class Base(DeclarativeBase):
    metadata = MetaData(naming_convention=naming_convention)

class Device(Base):
    __tablename__ = "device"

    id_device: Mapped[int] = mapped_column(Integer, primary_key=True,
        autoincrement=True)
    name: Mapped[str] = mapped_column(Text, unique=True, nullable=False)

    algorithms: Mapped[List["DeviceAlgorithm"]] = relationship(
        back_populates="device", cascade="all, delete-orphan"
    )

class Algorithm(Base):
    __tablename__ = "algorithm"

    id_algorithm: Mapped[int] = mapped_column(Integer, primary_key=True,
        autoincrement=True)
    name: Mapped[str] = mapped_column(Text, unique=True, nullable=False)

    devices: Mapped[List["DeviceAlgorithm"]] = relationship(
        back_populates="algorithm", cascade="all, delete-orphan"
    )
    compatibilities: Mapped[List["Compatibility"]] = relationship(
        back_populates="algorithm", cascade="all, delete-orphan"
    )

class Cryptocurrency(Base):
    __tablename__ = "cryptocurrency"

    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True,
```

```
        autoincrement=True)
    name: Mapped[str] = mapped_column(Text, unique=True, nullable=False)

    parameters: Mapped[Optional["Parameters"]] = relationship(
        back_populates="cryptocurrency", uselist=False, cascade="all,␣delete-orphan"
    )
    networks: Mapped[List["Network"]] = relationship(
        back_populates="cryptocurrency", cascade="all,␣delete-orphan"
    )

class Pool(Base):
    __tablename__ = "pool"

    id_pool: Mapped[int] = mapped_column(Integer, primary_key=True,
        autoincrement=True)
    name: Mapped[str] = mapped_column(Text, unique=True, nullable=False)

class RewardSystem(Base):
    __tablename__ = "reward_system"

    id_rewardsystem: Mapped[int] = mapped_column(Integer, primary_key=True,
        autoincrement=True)
    name: Mapped[str] = mapped_column(Text, unique=True, nullable=False)

class DeviceAlgorithm(Base):
    __tablename__ = "device_algorithm"

    id_device: Mapped[int] = mapped_column(Integer)
    id_algorithm: Mapped[int] = mapped_column(Integer)
    hashrate: Mapped[float] = mapped_column(Numeric(20, 8), nullable=False) # H/s
    energy_consumption: Mapped[float] = mapped_column(Numeric(12, 4),
        nullable=False) # W

    __table_args__ = (
        PrimaryKeyConstraint("id_device", "id_algorithm"),
        ForeignKeyConstraint(["id_device"], ["device.id_device"], ondelete="CASCADE"),
        ForeignKeyConstraint(["id_algorithm"], ["algorithm.id_algorithm"],
            ondelete="CASCADE"),
        CheckConstraint("hashrate␣>=␣0", name="hashrate_nonneg"),
        CheckConstraint("energy_consumption␣>=␣0", name="energy_nonneg"),
        Index("ix_device_algorithm__id_algorithm", "id_algorithm"),
    )
```

```
    device: Mapped["Device"] = relationship(back_populates="algorithms")
    algorithm: Mapped["Algorithm"] = relationship(back_populates="devices")

class Compatibility(Base):
    __tablename__ = "compatibility"

    id_algorithm: Mapped[int] = mapped_column(Integer)
    id_cryptocurrency: Mapped[int] = mapped_column(Integer)

    __table_args__ = (
        PrimaryKeyConstraint("id_algorithm", "id_cryptocurrency"),
        ForeignKeyConstraint(["id_algorithm"], ["algorithm.id_algorithm"],
            ondelete="CASCADE"),
        ForeignKeyConstraint(["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"], ondelete="CASCADE"),
        Index("ix_compatibility__id_cryptocurrency", "id_cryptocurrency"),
    )

    algorithm: Mapped["Algorithm"] = relationship(back_populates="compatibilities")

class PoolCryptoReward(Base):
    __tablename__ = "pool_crypto_reward"

    id_pool: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_rewardsystem: Mapped[int] = mapped_column(Integer, primary_key=True)
    pool_fee: Mapped[float] = mapped_column(Numeric(6, 4), nullable=False,
        default=0) # 0.02 = 2%
    pool_hashrate: Mapped[int] =mapped_column(Integer, nullable=False)

    __table_args__ = (
        PrimaryKeyConstraint("id_pool", "id_cryptocurrency", "id_rewardsystem"),
        ForeignKeyConstraint(["id_pool"], ["pool.id_pool"], ondelete="CASCADE"),
        ForeignKeyConstraint(["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"], ondelete="CASCADE"),
        ForeignKeyConstraint(["id_rewardsystem"], ["reward_system.id_rewardsystem"],
            ondelete="CASCADE"),
        CheckConstraint("pool_fee >= 0", name="pool_fee_nonneg"),
        Index("ix_pool_crypto_reward__id_cryptocurrency", "id_cryptocurrency"), ,
    )
```

```
class Parameters(Base):
    __tablename__ = "parameters"

    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    constant_c: Mapped[Optional[float]] = mapped_column(Numeric(10, 6))
    block_reward: Mapped[float] = mapped_column(Numeric(18, 8), nullable=False)
    Tb: Mapped[int] = mapped_column(Integer,default=600) # iba en cryptocurrency, la
        cambie porque es mejor que todo este aqui.

    __table_args__ = (
        ForeignKeyConstraint(["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"], ondelete="CASCADE"),
    )

    cryptocurrency: Mapped["Cryptocurrency"] =
        relationship(back_populates="parameters")

class Network(Base):
    __tablename__ = "network"

    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    #ts: Mapped[datetime] = mapped_column(DateTime(timezone=True), primary_key=True)
    difficulty: Mapped[float] = mapped_column(Numeric(30, 10), nullable=False)
    global_hashrate: Mapped[float] = mapped_column(Numeric(30, 10), nullable=False)
    price: Mapped[float] = mapped_column(Numeric(28, 10), nullable=False)
    avg_tx_block_fee: Mapped[float] = mapped_column(Numeric(20, 10), nullable=True)

    __table_args__ = (
        ForeignKeyConstraint(["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"], ondelete="CASCADE"),
        CheckConstraint("difficulty >= 0", name="difficulty_nonneg"),
        CheckConstraint("global_hashrate >= 0", name="ghr_nonneg"),
        CheckConstraint("price >= 0", name="price_nonneg"),
    )

    cryptocurrency: Mapped["Cryptocurrency"] =
        relationship(back_populates="networks")

# =================
# 4) Estimaciones
# =================
```

```
class Estimations(Base):
    __tablename__ = "estimations"

    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_pool: Mapped[int] = mapped_column(Integer, primary_key=True)
    blocks_solved: Mapped[float] = mapped_column(Numeric(20, 4), nullable=False,
        default=0)
    total_sharerate: Mapped[float] = mapped_column(Numeric(30, 10), nullable=False,
        default=0)

    __table_args__ = (
        ForeignKeyConstraint(["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"], ondelete="CASCADE"),
        ForeignKeyConstraint(["id_pool"], ["pool.id_pool"], ondelete="CASCADE"),
        CheckConstraint("blocks_solved␣>=␣0", name="blocks_nonneg"),
        CheckConstraint("total_sharerate␣>=␣0", name="tsr_nonneg"),
    )

##12 nov 25
class Snapshot(Base):
    __tablename__ = "snapshot"

    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_pool: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_rewardsystem: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_device: Mapped[int] = mapped_column(Integer, primary_key=True)

    payout: Mapped[Optional[float]] = mapped_column(Numeric(30, 12), nullable=True,
        default=None)
    fee: Mapped[Optional[float]] = mapped_column(Numeric(10, 6), nullable=True,
        default=None)
    solved_blocks: Mapped[Optional[float]] = mapped_column(Numeric(20, 6),
        nullable=True, default=None)
    B: Mapped[Optional[float]] = mapped_column(Numeric(20, 8), nullable=True,
        default=None) # Block reward
    pool_hashrate: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    global_hashrate: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    price: Mapped[Optional[float]] = mapped_column(Numeric(28, 10), nullable=True,
        default=None)
    hashrate_device: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
```

```
        nullable=True, default=None)
    difficulty_share: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    difficulty_network: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    constant_c: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    energy_consumption: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    cost: Mapped[Optional[float]] = mapped_column(Numeric(30, 10), nullable=True,
        default=None)
    profit: Mapped[Optional[float]] = mapped_column(Numeric(30, 10), nullable=True,
        default=None)
    __table_args__ = (
        ForeignKeyConstraint(
            ["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_pool"],
            ["pool.id_pool"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_rewardsystem"],
            ["reward_system.id_rewardsystem"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_device"],
            ["device.id_device"],
            ondelete="CASCADE"
        ),

        CheckConstraint("payout␣>=␣0␣OR␣payout␣IS␣NULL",
            name="snapshot_payout_nonneg"),
        CheckConstraint("fee␣>=␣0␣OR␣fee␣IS␣NULL", name="snapshot_fee_nonneg"),
        CheckConstraint("solved_blocks␣>=␣0␣OR␣solved_blocks␣IS␣NULL",
            name="snapshot_blocks_nonneg"),
        CheckConstraint("pool_hashrate␣>=␣0␣OR␣pool_hashrate␣IS␣NULL",
            name="snapshot_poolhash_nonneg"),
```

```
        CheckConstraint("global_hashrate␣>=␣0␣OR␣global_hashrate␣IS␣NULL",
            name="snapshot_globalhash_nonneg"),
        CheckConstraint("hashrate_device␣>=␣0␣OR␣hashrate_device␣IS␣NULL",
            name="snapshot_devicehash_nonneg"),
        CheckConstraint("difficulty_share␣>=␣0␣OR␣difficulty_share␣IS␣NULL",
            name="snapshot_diffshare_nonneg"),
        CheckConstraint("difficulty_network␣>=␣0␣OR␣difficulty_network␣IS␣NULL",
            name="snapshot_diffnet_nonneg"),

        Index("ix_snapshot_crypto_pool", "id_cryptocurrency", "id_pool"),
    )

class Simulacion1(Base):
    __tablename__ = "Simulacion1"

    # PK compuesta
    id_cryptocurrency: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_pool: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_rewardsystem: Mapped[int] = mapped_column(Integer, primary_key=True)
    id_device: Mapped[int] = mapped_column(Integer, primary_key=True)

    payout: Mapped[Optional[float]] = mapped_column(Numeric(30, 12), nullable=True,
        default=None)
    fee: Mapped[Optional[float]] = mapped_column(Numeric(10, 6), nullable=True,
        default=None)
    solved_blocks: Mapped[Optional[float]] = mapped_column(Numeric(20, 6),
        nullable=True, default=None)
    B: Mapped[Optional[float]] = mapped_column(Numeric(20, 8), nullable=True,
        default=None) # Block reward
    pool_hashrate: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    global_hashrate: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    price: Mapped[Optional[float]] = mapped_column(Numeric(28, 10), nullable=True,
        default=None)
    hashrate_device: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    difficulty_share: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    difficulty_network: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    constant_c: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
```

```
        nullable=True, default=None)
    energy_consumption: Mapped[Optional[float]] = mapped_column(Numeric(30, 10),
        nullable=True, default=None)
    cost: Mapped[Optional[float]] = mapped_column(Numeric(30, 10), nullable=True,
        default=None)
    profit: Mapped[Optional[float]] = mapped_column(Numeric(30, 10), nullable=True,
        default=None)
    __table_args__ = (
        ForeignKeyConstraint(
            ["id_cryptocurrency"],
            ["cryptocurrency.id_cryptocurrency"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_pool"],
            ["pool.id_pool"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_rewardsystem"],
            ["reward_system.id_rewardsystem"],
            ondelete="CASCADE"
        ),
        ForeignKeyConstraint(
            ["id_device"],
            ["device.id_device"],
            ondelete="CASCADE"
        ),
    )

Base.metadata.drop_all(engine)
Base.metadata.create_all(engine)
```

.3 Linear Programming Model

Listing 2: Linear Programming Model

```
import sqlite3
import pandas as pd
import pulp as lp
import time
```

```
DB_PATH = "pow.db"
conn = sqlite3.connect(DB_PATH)
query = """
SELECT
␣␣␣␣s.id_device,
␣␣␣␣d.name␣AS␣device_name,
␣␣␣␣s.id_cryptocurrency,
␣␣␣␣c.name␣AS␣cryptocurrency_name,
␣␣␣␣s.id_pool,
␣␣␣␣p.name␣AS␣pool_name,
␣␣␣␣s.id_rewardsystem,
␣␣␣␣s.global_hashrate,
␣␣␣␣s.pool_hashrate,
␣␣␣␣r.name␣AS␣rewardsystem_name,
␣␣␣␣s.fee,
␣␣␣␣s.payout,
␣␣␣␣s.cost,
␣␣␣␣s.profit
FROM␣snapshot␣AS␣s
JOIN␣device␣AS␣d
␣␣␣␣ON␣d.id_device␣=␣s.id_device
JOIN␣cryptocurrency␣AS␣c
␣␣␣␣ON␣c.id_cryptocurrency␣=␣s.id_cryptocurrency
JOIN␣pool␣AS␣p
␣␣␣␣ON␣p.id_pool␣=␣s.id_pool
JOIN␣reward_system␣AS␣r
␣␣␣␣ON␣r.id_rewardsystem␣=␣s.id_rewardsystem;
"""
df = pd.read_sql_query(query, conn)
conn.close()
df = df.dropna(subset=["payout", "cost", "profit"], how="any")
df = df[~((df["id_rewardsystem"] == 1) & (df["global_hashrate"] >
    df["pool_hashrate"]))]
print(df.head())
if "profit" not in df.columns:
   df["profit"] = df["payout"] - df["cost"]
else:
   df["profit"] = df["profit"].fillna(df["payout"] - df["cost"])
df["id_row"] = df.index

S = df["id_row"].tolist()
devices = df["id_device"].unique().tolist()
```

```
profit = dict(zip(df["id_row"], df["profit"]))
device_of = dict(zip(df["id_row"], df["id_device"]))

print(f"Number␣of␣combinations:␣{len(df)}")
print(f"Number␣of␣distinct␣devices:␣{len(devices)}")
print("Combinations␣per␣device:")
print(df["id_device"].value_counts())

model = lp.LpProblem("MiningOptimization", lp.LpMaximize)

x = lp.LpVariable.dicts(
   "x",
   S,
   lowBound=0,
   upBound=1,
   cat=lp.LpBinary
)

model += lp.lpSum(profit[s] * x[s] for s in S)

# Constraints
for k in devices:
   model += lp.lpSum(x[s] for s in S if device_of[s] == k) == 1,
       f"device_{k}_one_choice"

# 4) Solve
solver = lp.PULP_CBC_CMD(msg=True)
start = time.perf_counter()
model.solve(solver)
end = time.perf_counter()
solve_time = end - start

optimal_profit = lp.value(model.objective)
#print(f"Optimal␣profit:␣{optimal_profit:.2f}␣USD.")


solution_rows = []

for s in S:
   if lp.value(x[s]) > 0.5:
      row = df[df["id_row"] == s].iloc[0]
```

```
        solution_rows.append(row)

solution_df = pd.DataFrame(solution_rows)

solution_df = solution_df.sort_values(by="profit", ascending=False)

print("\nOptimal␣configuration␣per␣device")
print(
    solution_df[
        [
            "id_device",
            "device_name",
            "id_cryptocurrency",
            "cryptocurrency_name",
            "id_pool",
            "pool_name",
            "id_rewardsystem",
            "rewardsystem_name",
            "fee",
            "payout",
            "cost",
            "profit",
        ]
    ]
)



print('Output:␣')
print('Device␣n␣|␣device␣name␣|␣cryptocurrency␣|␣pool␣|␣reward␣system␣|␣payout␣|␣
     cost␣|␣profit_24H')
for _, row in solution_df.iterrows():
    print(
        f"Device␣{row['id_device']}␣|␣'{row['device_name']}␣|␣
            {row['cryptocurrency_name']}␣|"

        f"␣{row['pool_name']}␣|␣{row['rewardsystem_name']}␣|␣{row['payout']:.3f}␣|␣
            {row['cost']:.3f}␣|␣{row['profit']:.3f}␣USD.␣"
    )
print(f"Optimal␣profit:␣{optimal_profit:.2f}␣USD.")
print(f"Solution␣time:␣{solve_time:.2f}␣seconds")
```