Your Money or Your Privacy: A Systematic Approach to Coin Selection

Svetlana Abramova
University of Innsbruck

Rainer Böhme
University of Innsbruck

Abstract

Coin selection, a real-world variant of the NP-complete subset sum problem, refers to the selection of a set of unspent transaction outputs (UTXOs) in a cryptocurrency wallet. The total value of the set must match or exceed the required transaction amount. This decision represents an intertemporal multi-objective optimization problem which concerns individual and collective interests of ordinary users in a decentralized network. From the individual's perspective, coin selection entails a trade-off over time between transaction fees and privacy. In particular, uninformed decisions may link several transactions made by the same person and thereby compromise one's financial privacy. We formally define this complex problem in a two-period model and analyze it for two types of users, myopic and strategic. The combinatorial approach proposed to generically solve the model relies on exhaustive search and leverages the capabilities of *computer algebra* and *satisfiability checking* tools. We demonstrate our approach on two computationally tractable instances of the model and compare the results for myopic, strategic as well as heuristic-based coin selection.

1 Introduction

Public distributed ledgers, the technology behind most decentralized cryptocurrencies, have turned into a readily accessible source of registered user payments. In contrast to other logs of user activity collected aside for later analysis, the ledger of all confirmed transactions – the *blockchain* – is required by the protocol itself to achieve a consistent system-wide state in the decentralized network. However, many public distributed ledgers unwittingly expose users to privacy risks. Since systems like Bitcoin store all transactions unencrypted on the blockchain, they reveal potentially sensitive information, such as the digital identities of transaction partners or the transacted values.

Though some privacy-enhancing techniques are regularly proposed, many cryptocurrencies remain vulnerable to blockchain-based analyses examining users' behavioral patterns and modes of use [2,18,20,23,27,32–34]. In fact, privacy concerns usually go beyond the scope of a single transaction, a wallet or even a single blockchain. Blockchain data may be exploited to link multiple transactions made by the same person, both within or across ledgers [36]. This information, combined with other digital traces left by users on the web [11, 17], may lead to an accumulation of individual's preferences, purchase histories, and extensive user profiling.

The problem of user privacy and linkability of individual transactions is rooted in the design of UTXO-based cryptocurrency systems. To be precise, it is closely related to a decision making process termed as *coin selection*. Whenever a payment need arises, the user must decide which coins (i. e., unspent transaction outputs) in a cryptocurrency wallet to spend in that transaction. Coin selections are made on a regular basis, reaching (at the time of writing) around 300 000 decisions per day [7]. We argue that they are pivotal to ordinary users who are interested in concealing their payment history from other transaction counterparties.

At its core, coin selection presents an *intertemporal* optimization problem, which affects both the individual user and the entire network. This has several reasons. First, a sequence of coin selections may link multiple transactions to the same user and thus compromise financial privacy. An illustrative example is a transaction which spends a UTXO left over as a "change" from an earlier payment. Such a transaction is linked to the preceding transaction where the change originates from. Second, coin selection impacts transaction fees, which the user must pay to miners for their efforts in validating the transaction and extending the blockchain. In general, the fewer UTXOs are spent in a transaction, the lower are the transaction fees. Thus, it is in user's best interests to keep transactions compact. Lastly, UTXOs may be of arbitrary value, which gives users much flexibility in partitioning or aggregating UTXOs, again involving coin selection. However, the blockchain is a shared resource, and the network must store every unspent output at each point in time.[1] Therefore,

[1] Note that the actual implementation is more complicated than this de-

it is socially desirable to have fewer UTXOs of large denominations than many UTXOs of small values. This limits the growth of the blockchain and reduces the storage requirements imposed on all full nodes.

A systematic approach to coin selection requires an intertemporal perspective because local decisions made by the user for individual transactions may be suboptimal over a longer time horizon. In practice, users rely (often blindly) on cryptocurrency wallet software whose hard-coded logic automates the coin selection process. Ordinary users are often poorly equipped in terms of resources, knowledge, and skills to comprehend or objectively evaluate this logic. What is even more challenging is to predict potential implications of the coin selection decision over time. Against this backdrop, this paper aims to break ground for research on this understudied topic and examines coin selection as an intertemporal trade-off between transaction linkability and fees.

A common simplification in many studies on intertemporal choices in the economic literature involves two-period decision making models that account for two extremes of behavior, *myopic* and *strategic* [1, 10]. Myopic behavior is dominated by short-term needs and immediate rewards, whereas strategic behavior is formed by forward-looking concerns about the impact of present decisions on the future well-being of agents. We adopt this convention and model coin selection as a two-period multi-objective optimization problem, where each period refers to a single transaction. We argue that a two-period model is enough for our purpose because failing to make two unlinkable transactions rules out privacy in a larger transaction graph.

Besides the formal modeling, we propose a novel combinatorial solution approach and demonstrate its application. A unique technical feature of our approach is that, in an effort to find a generic solution, it combines symbolic computer algebra with the efficient search capabilities of a satisfiability modulo theories (SMT) solver.

Contributions. In short, this paper makes the following contributions:

1. to our best knowledge, this is the first study that reasons about coin selection formally;
2. it leverages computer algebra to solve the intertemporal problem generically;
3. it validates the proposed approach on two computationally tractable instances of the model and compares the results for different modes of coin selection.

The rest of the paper is organized as follows. We recall basic principles of the transaction logic of cryptocurrencies in Section 2. To provide the necessary context, we discuss coin selection through theoretical and practical lenses in Sections 3 and 4, respectively. In Section 5, we model the coin selection problem formally. Section 6 presents the solution approach, its application, and the results obtained. We put our effort in context of related work in Section 7 before we conclude in Section 8.

2 Preliminaries

We give a succinct introduction to the most relevant concepts and principles of the transaction logic on typical blockchains. (Please refer to, e. g., [28] for a more general and complete reference.) In the paper, we use Bitcoin as a reference, though our model and analysis generalizes to other UTXO-based cryptocurrencies sharing similar transaction principles.

There are two models of managing funds on the blockchain: *account-based*, as implemented by Ethereum, and *UTXO-based*, as implemented by Bitcoin. As its name suggests, the account-based model maintains a list of accounts and their balances as part of the shared state. The UTXO-based system, on the contrary, uses a special data structure of *unspent transaction outputs*. In the paper, we limit our analysis to Pay-to-Public-Key-Hash (P2PKH) outputs, making up the bulk of the entire UTXO set. Each UTXO of this type has a value and a cryptographic locking script requiring a pair of the public and private keys in order to spend that UTXO. A *public address* is generated from the public key and interpreted as a technical identifier of the owner of the UTXO. There is a one-to-many relationship between an individual address and UTXOs: i. e., multiple UTXOs may share the same address. Users transfer values between addresses via transactions, which consume existing UTXOs and generate new ones for future payments. Utxos that are spent and generated in a transaction are called *transaction inputs* and *outputs*, respectively. Following this terminology, coin selection is the process of choosing UTXOs contained in a cryptocurrency wallet as transaction inputs.

A UTXO can be of any positive nominal value. However, it must be spent entirely when referenced as a transaction input. If the value of a UTXO exceeds the required transaction amount, a new UTXO (*change output*) may be created to return the excess amount to the sender. This output may reuse one of the input addresses or be assigned to a new address under the sender's control.

While public addresses act as digital pseudonyms, the transaction history is completely transparent and traceable. The design principle that transaction inputs reference valid outputs of past transactions allows a passive observer of the blockchain to construct a transaction graph and track money flows. While it is straightforward to find a balance of an individual address on the blockchain, the pseudonymity property makes it harder to identify all addresses controlled by the same owner. In practice, however, linking transactions, UTXOs, and ad-

scription. For example, spent coins in Bitcoin are often held persistent without a technical need. This may change in the future. We aim here at a generalizable presentation.

dresses which are likely to be made or controlled by the same person, still remains possible. Specifically, *multi-input* and *change heuristics* are the two most common methods used to this end [2, 23, 33, 34]. The multi-input heuristic assumes that the addresses of all inputs of a given transaction belong to the same user. The change heuristic tries to identify among (potentially many) transaction outputs the change output. If it exists, the heuristic assumes that the change address and all input addresses of that transaction belong to the same user. These two heuristics, along with some special treatment of corner cases, are common tools used by commercial services to form clusters of addresses that likely belong to the same user.

3 Coin Selection in Theory

For a new transaction, the user must select some UTXOs from the set of all available UTXOs with the minimum total value of a transaction need plus transaction fees. So, coin selection can be interpreted as a variant of a well-known combinatorial optimization problem – the subset sum problem [22]. In fact, it is a multi-objective discrete optimization problem, which represents a trade-off between *user privacy*, *transaction fees*, and *the maintenance overhead of the blockchain*. These goals partly conflict with each other and can be classified as individual (fees and privacy) and collective (maintenance). In practice, coin selection is often left to the discretion of wallet software developers. Almost all software prioritizes the maintenance goal and attempts to limit the ever-growing blockchain data [15]. Below, we briefly discuss each aspect of the trade-off.

The blockchain size (or at the least, the total set of all UTXOs present in the system at each point in time) determines storage requirements of a single full node in the network. Thus, the smaller the global UTXO set is, the better it is for the entire network. One example of the undesirable state of the UTXO pool is when it contains many unprofitable change outputs known as "dust". Although the values of such UTXOs are lower than the marginal fee needed to spend them, they cannot be ignored or automatically pruned.

Privacy is another objective the coin selection algorithm may optimize for. In a narrow sense, privacy can be improved for a single transaction by avoiding the selection of UTXOs which are associated with different addresses. A more advanced and desirable strategy is to look both backwards and forward in order to account for the risk of linking multiple transactions (and the addresses therein) due to suboptimal coin selections. This interpretation of user privacy adds a temporal dimension to coin selection. If privacy matters, users should consider prospective payments and be more deliberate in their choices. For example, the user may better refrain from spending a particular UTXO in the current transaction because it is a better candidate for a future transaction which spends other UTXOs belonging to this UTXO's address.

Miners prioritize unconfirmed transactions by fees and either exclude or delay the confirmation of less profitable payments. In an ideal world, the coin selection problem can be reduced to the trade-off between transaction fees and privacy only, since fees are meant to internalize the externalities of provisioning the blockchain, the cost of proof-of-work computations, and the cost of storing a transaction record [24]. However, real systems are not ideal. In contrast to one-off costs of proof-of-work incurred by miners, storage costs are imposed on all full nodes and depend on the transaction size, the lifetime of all transaction outputs, and the size of the network. Hence, it is impossible to predict the total cost at the time a new output is generated. This imperfection explains why in practice fees are approximated by the transaction size. We review coin selection and fee estimation algorithms implemented in popular cryptocurrency wallets in the next section.

4 Coin Selection in Practice

In practice, coin selection decisions are almost always made by algorithms built into the existing cryptocurrency wallets. To date, wallets come in many forms, with different features and for different platforms. We take an in-depth look at Bitcoin Core, which is the reference client and arguably the most popular wallet among Bitcoin users. In addition, we briefly review coin selection policies adopted in some alternative Bitcoin wallets. To complete the context, we also document other events which might have affected the coin selection practices in Bitcoin in the past.

4.1 Coin Selection in Cryptocurrency Wallets

Coin selection in the Bitcoin Core software client has undergone minor changes over the past several years. In early implementations before the **release 0.17.0 (October 3, 2018)**, a multi-step algorithm tries to find a set of UTXOs whose total value exactly matches the required amount. Depending on the algorithm's stage, this value is determined by the transaction need, a minimum transaction fee, and, if necessary, a minimum change output (set to 0.01 BTC by default). The fee is re-estimated after each selection round and passed as an input parameter to the next round. If no exact match is found, the algorithm runs a knapsack solver which iteratively selects from a descendingly sorted set of outputs smaller than the target value and attempts to find either an exact match or a subset with the lowest excess over the target. Outputs are chosen randomly with a 50 % chance and the smallest valid set is returned as the best result. Finally, the algorithm compares this candidate set against the minimum UTXO which is larger than the required amount and selects the option that produces a lower change.

The implementation also allows users to customize and partly influence coin selection. In **version 0.9.0 (March 19, 2014)**, users were given the option to manually preselect

UTXOs in their wallets. Bitcoin Core first checks whether the preselected UTXOs are enough to fund a transaction. Otherwise, it reduces the target value by the sum of preselected UTXOs and continues the coin selection for the remaining amount in an ordinary manner. This version also introduced an option for users to specify a custom change address instead of adopting a newly generated one. It should be noted, however, that there is no user study exploring the uptake or presenting empirical evidence on the use of these features.

The seminal work of [15] on coin selection has demonstrated several drawbacks of Bitcoin Core's algorithm. First, the algorithm does not adjust transaction fees in each selection round, but re-calculates them in-between. Secondly, the algorithm is inefficient and computationally expensive as it repeatedly starts the knapsack solver and runs over the same space of combinations. To overcome these limitations, Erhardt [15] proposed in his thesis a branch and bound algorithm. The algorithm calculates a contribution of each UTXO to the target value, builds a binary tree of available UTXOs and then searches it in a depth-first-search. This proposal has been implemented and deployed in the **release 0.17.0 (October 3, 2018)**.

Summing up, coin selection in Bitcoin Core is largely oriented by the policy of minimizing the global UTXO set. While it ensures that neither too high nor too little transaction fees are paid out to miners, it largely neglects user privacy and the risk of linking multiple transactions.

In contrast to Bitcoin Core, the desktop wallet Electrum attempts to address privacy constraints and favors UTXOs of the same address, while de-prioritizing addresses that appear in unconfirmed transactions. The *bitcoinj* library selects UTXOs based on a priority metric defined by the age and value of UTXOs. The *bitcoinjs* library has several coin selection algorithms following either a Highest Value First (HVF) or Lowest Value First (LVF) rule. HVF sorts all UTXOs contained in a wallet in a descending order of value and selects UTXOs from the top of this list until the target value is reached. The LVF rule differs in that UTXOs are sorted in an ascending order. These examples illustrate the diversity of coin selection algorithms. The high complexity and a lack of the one-size-fits-all solution highlight a need for a systematic approach in studying this problem.

4.2 Other Influencing Factors

Though the coin selection algorithm of Bitcoin Core has remained fairly stable over many years, the growth rate[2] of the global UTXO set in Bitcoin (see Figure 1) shows large temporal variations. In fact, there are other *endogenous* and *exogenous* factors that influence coin selection decisions.

The main endogenous factor is a highly dynamic market of transaction fees. The coin selection algorithm in Bitcoin Core

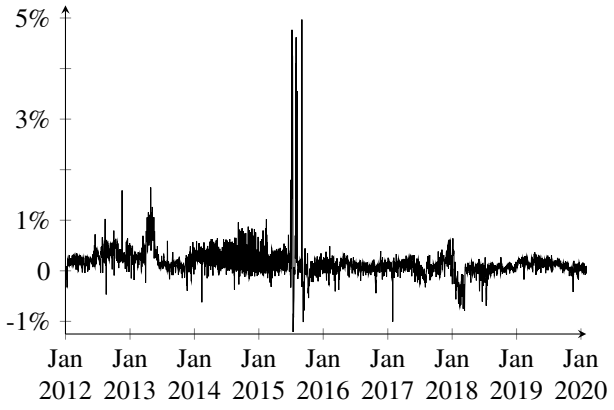

Figure 1: Daily growth rate of the Bitcoin UTXO set. Source: own analysis of blockchain data

closely interacts with a fee estimation tool, which has undergone repeated changes. Starting from optional and default absolute levels, it has progressed to a more complicated calculation. As of today, fees in Bitcoin are calculated based on satoshi[3] per kilobyte. Adding an extra transaction input or output is costly due to an increase of the transaction size. Moreover, the fees depend on the utilization of the blockchain's capacity. In Bitcoin Core, the fee estimation tool evaluates the most recent transactions and their confirmation times in order to calculate the lowest fee for several confidence levels that a transaction will be confirmed in a given time frame. We refer the interested reader to [35] for further details.

Stress tests and spam attacks against the system are exogenous factors with a "ripple effect" that can be observed in the second half of 2015. Those campaigns flooded the network with high fan-in and fan-out transactions in an attempt to promote the raise of the block size limit, a contentious issue at the time [4]. The apparent reduction of the UTXO set in early 2018 is likely caused by the concurrent drop of transaction fees and a general trend among expert users towards consolidating many small, unprofitable UTXOs. Also, from **version 0.16.0 (February 26, 2018)**, the client allows users to generate SegWit-compatible addresses. In a nutshell, a SegWit address format requires less transaction data to be stored on the blockchain, thereby freeing up some additional block space. The transition to such addresses may have stimulated some Bitcoin users to "sweep" wallets, i. e., to consolidate many dust outputs into a UTXO of higher value.

5 Modeling Coin Selection

Coin selection belongs to a class of problems of high combinatorial complexity. Therefore, we restrict its scope by examining the trade-off between the individual objectives (i. e., transaction fees and privacy) only. We analyze two variants

[2] approximated by the first difference of logarithms

[3] The smallest unit in Bitcoin: 1 satoshi $= 10^{-8}$ bitcoin.

of the optimization model. In the first, the user is *myopic* and minimizes the fee of a one-off transaction. In the second, the user is *strategic* and anticipates future payments. Thus, the strategic user considers the implications of coin selection decisions on her financial privacy. The ultimate objective of strategic coin selection is therefore to make several unlinkable payments while minimizing total transaction fees.

Since an intertemporal optimization is generally non-trivial and computationally demanding, we limit our model to two periods. As mentioned in the introduction, such models are a common simplification for many intertemporal problems in the economics literature (e. g., when modeling strategic consumer behavior [10] or merchants' conditioning price discrimination [1]). We consider an abstract Bitcoin-style cryptocurrency and allow that a transaction may have an arbitrary number of inputs; however, it has exactly two outputs: the payment itself and the change output [4]. For simplicity, we ignore external incoming payments between the two periods and approximate fees as a linear function of the number of transaction inputs.[5]

5.1 Terminology

The authors of [3] propose an abstract model to enable formal reasoning on Bitcoin transactions. We refrain from adopting this model here due to its focus on technical details, which are not relevant in our context. Instead, we formally define only those terms and concepts that are required for modeling coin selection and presenting a general solution approach.

An *address a* serves as an account number and is technically generated from a public key. Let $\mathbb{A}$ denote a set of all addresses present in a cryptocurrency system.

Definition 1 (Unspent transaction output) *An unspent transaction output (shorthand,* UTXO*) is a tuple of the form* (a, v)*, where the address* $a \in \mathbb{A}$ *defines the owner of the UTXO and* $v \in \mathbb{R}^+$ *denotes the value of the UTXO in units of cryptocurrency.*

This way, the balance of any address a is the sum of values of all UTXOs sharing the same address a. Each UTXO can be spent independently from all other UTXOs.

Definition 2 (Wallet) *A wallet* $W = \{a_1, \ldots, a_d\}$ *is a set of addresses, which have a non-zero balance and are controlled by the same real-world owner. We say that the wallet W* ***contains*** *an UTXO* (a, v) *if the address a is an element of W.*

In our model, the real-world owner acts as a decision maker in the coin selection problem. By convention, we call the number of addresses $d \in \mathbb{N}$ the *cardinality* of W and use the term *size of the wallet* to refer to the number of UTXOs contained in a given wallet W. Observe that $1 \leq d \leq m$, where m denotes the initial size of W (not counting the change output). To rule out symmetries in the analysis part, we assume that UTXOs contained in the wallet W are ordered by the ascending values. In the coin selection problem, addresses of UTXOs matter for privacy, whereas the values of UTXOs constrain the decision space. Therefore, the combinatorial part of our approach disregards addresses and operates solely on the values of UTXOs contained in W. The addresses are required for the calculation of a cost function and hence, are considered later at the evaluation stage.

Transactions move values between different addresses. A transaction emerges from an exogenous transaction need n, i. e., some amount to be paid to another party. The exact composition of a transaction is the result of the user's choice subject to a number of constraints. We distinguish two types of constraints: *combinatorial* and *funding* constraints. A combinatorial constraint states that any UTXO can be spent only once (as enforced by the foundational principle of the UTXO model [28]). This constraint is independent of the exact values of neither UTXOs nor transaction needs. A funding constraint states that the sum of transaction inputs must be no less than the transacted amount. Not every combinatorially possible selection of UTXOs is enough to satisfy a particular transaction need. Hence, in contrast to combinatorial constraints, funding constraints depend on transaction needs and the values of the selected UTXOs. Every funding constraint can be expressed as a linear inequality.

Definition 3 (Elementary choice) *An elementary choice s is a mapping of all UTXOs contained in a given wallet W to the ternary set* $\{t_1, t_2, \perp\}$*, and of a* change output (a_h, v_h) *generated in the first transaction to the binary set* $\{t_2, \perp\}$*, where* t_1 *denotes that the UTXO is spent in the first transaction,* t_2 *– in the second transaction, and* $\perp$ *denotes that the UTXO is not spent at all.*

Note that this definition of elementary choice already respects the combinatorial constraints. The evaluation of the funding constraints determines whether the elementary choice is feasible.

Lemma 1 *For a wallet of size m, the total number of elementary choices q is given by:*

$$q(m) = \sum_{i=1}^{m} \binom{m}{i} (2^{m-i+1} - 1). \tag{1}$$

Proof 1 *Disregarding the funding constraints, the decision maker has* $\binom{m}{i}$ *possible combinations to spend i UTXOs in the first transaction. Recall that the change output is added to the wallet W after the first transaction. Hence, for each combination selected in* t_1*, the decision maker has* $(m-i+1)$ *UTXOs left in the wallet for the second transaction, which*

[4] We do not handle cases without the change output specifically because they are so rare in practice.

[5] Strictly speaking, in Bitcoin, the block space required for both inputs and outputs must be accounted for in transaction fees. Fixing the number of outputs, however, does not change the preference order.

result in the power set of $(2^{m-i+1}-1)$ *possible combinations (without the empty set).*

The set of elementary choices can be visualized in tree or matrix form. Each path of the tree corresponds to a sequence of the coin selection decisions made in periods t_1 and t_2. A corresponding funding constraint can be assigned to each edge. Alternatively, all possible coin selections in t_1 and t_2 can be arranged along columns and rows of a two-dimensional matrix. In this representation, we need to account for the combinatorial constraints and exclude all cross-product combinations in the matrix which contain the same UTXO in both periods. The funding constraints can be annotated to each column and row of the matrix. As a result, each elementary choice is constrained by a system of two linear inequalities. Note that it is possible to establish a complete order of all elementary choices in order to identify a certain choice by its index.

Example 1 *For simplicity, consider the wallet* $W = \{a_1, a_2\}$ *that contains two UTXOs* (a_1, v_1) *and* (a_2, v_2)*, where* $0 < v_1 \leq v_2$*. Recall that* (a_h, v_h) *denotes the change output of the first transaction. The decision tree of all elementary choices for this example is shown in Figure 2. The corresponding matrix form is shown in Figure 5 in Appendix A.*

Definition 4 (Choice profile) *A choice profile* $\vec{p} \in \{0,1\}^q$ *is a binary vector of length q (given by Eq. 1) indicating which elementary choices are feasible alternatives under the set of funding constraints.*

A choice profile is said to be *active* if there exists a region in the $(m+2)$-dimensional parameter space where all elementary choices indicated by 1 in the profile are feasible and all elementary choices indicated by 0 in the profile are not feasible. The region is defined by a logical conjunction of the funding constraints for each set bit and negated conjunction of the funding constraints for each zero bit in the choice profile. For each realization of the parameters, the decision can be in at most one active choice profile.

Example 2 *We stick to Example 1 and specify a logical expression*[6] *of the constraints for one arbitrary instance of the choice profile* $\vec{p} = (0110111)$*. Let us establish the complete order of all elementary choices for this example as they are shown in Figure 2 from left to right. The constraints for each elementary choice can be derived from Figure 2 (or from Table 5 in Appendix B).*

$$
\begin{aligned}
&\neg(n_1 \leq v_1 \wedge n_2 \leq v_2) \wedge && \text{(\# 1, } m=2 \text{ in Table 5)}\\
&\quad (n_1 \leq v_1 \wedge n_2 \leq v_2 - n_1) \wedge && \text{(\# 2, } m=2 \text{ in Table 5)}\\
&\quad (n_1 \leq v_1 \wedge n_2 \leq v_1 + v_2 - n_1) \wedge && \text{(\# 3, } m=2 \text{ in Table 5)}\\
&\neg(n_1 \leq v_2 \wedge n_2 \leq v_1) \wedge && \text{(\# 4, } m=2 \text{ in Table 5)}\\
&\quad (n_1 \leq v_2 \wedge n_2 \leq v_2 - n_1) \wedge && \text{(\# 5, } m=2 \text{ in Table 5)}\\
&\quad (n_1 \leq v_2 \wedge n_2 \leq v_1 + v_2 - n_1) \wedge && \text{(\# 6, } m=2 \text{ in Table 5)}\\
&\quad (n_1 \leq v_1 + v_2 \wedge n_2 \leq v_1 + v_2 - n_1). && \text{(\# 7, } m=2 \text{ in Table 5)}
\end{aligned}
$$

A symbolic expression of this form can be generated and evaluated for every choice profile.

Observe that values of UTXOs appear as variables in the symbolic expressions, but what matters for privacy is their assignment to addresses. To facilitate notation, we use a helper function $\theta_W(i) : \mathbb{N} \to \mathbb{A}$, which takes an integer index i as argument and returns the address $a \in \mathbb{A}$ of the i-th smallest UTXO contained in the wallet W.

Definition 5 (Cost) *Given* $\lambda \in [0,1]$ *as the user's preference for privacy over lower transaction fees, a cost function c maps an elementary choice s to a real value:*

$$c(s, \theta_W, \lambda) : S \times \Theta \times \mathbb{R} \to \mathbb{R}^+$$

Following the notion of *Pareto optimality*, we call a feasible elementary choice s in the choice profile $\vec{p}$ *Pareto-optimal* if there is no other feasible elementary choice s' in $\vec{p}$ such that $c(s', \theta_W, \lambda) \leq c(s, \theta_W, \lambda)$ for any active choice profile $\vec{p}$ and fixed preference λ. If the inequality is strict, i. e., $c(s', \theta_W, \lambda) < c(s, \theta_W, \lambda)$, the elementary choice s is *weakly Pareto-optimal*.

5.2 Model

In each time period t_i, $i \in \{1,2\}$, a transaction need $n_i > 0$ arrives, where $n_1 + n_2 \leq \sum_{j=1}^{m} v_j$ (i. e., the user must have sufficient balance in the wallet W to satisfy both needs). The set S_i denotes a set of UTXOs selected in period t_i:

$$S_i = \{\underbrace{(a,v)}_{\text{UTXO}} \mid s\big((a,v)\big) = t_i\}. \tag{2}$$

The set $A_i \subseteq W$ denotes a set of addresses associated with S_i:

$$A_i = \{a \mid (a^*, v) \in S_i \wedge a = a^*\}. \tag{3}$$

Figure 3 illustrates the decision timeline for the two-period coin selection problem. Given the wallet W, the user learns n_1 and selects the set of UTXOs S_1 to spend in the first transaction. In period t_2, the user learns n_2 and selects the set of UTXOs S_2 from the remaining UTXOs plus the change output. Accounting for user types and whether the strategic user has partial or complete knowledge of n_2 when making the decision in period t_1, leads to the following list of conceivable scenarios:

[6] The expression is intentionally not simplified in order to make the provenance of terms explicit.

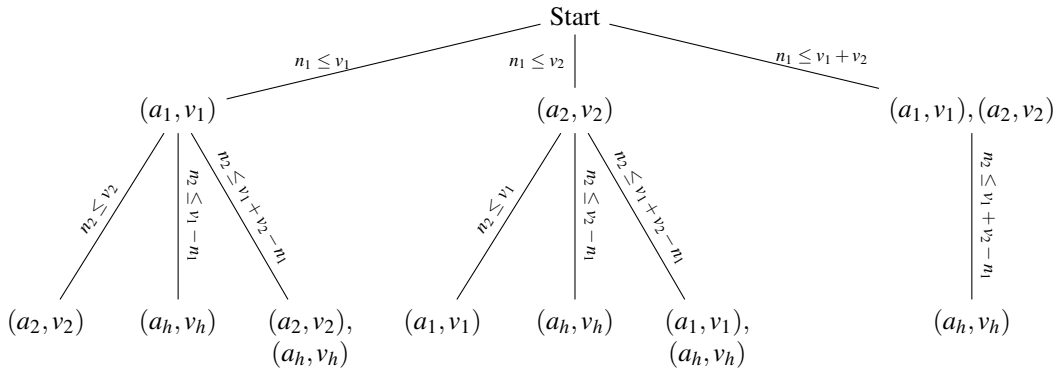

Figure 2: Decision tree of elementary choices. Note that the value of the change output v_h can be specified as a difference between the sum of values of the UTXOs spent in t_1 and the transaction need n_2. We replace v_h in the funding constraints by a corresponding symbolic expression.

1. *myopic optimization*, in which the coin selection decisions are made "locally" in each period for a given state of the wallet;
2. strategic optimization with *complete information* about n_2 in period t_1;
3. strategic optimization with *incomplete information* about n_2 in period t_1. The user forms expectations about n_2 in order to make an optimal choice in t_1.

Here we study the first two scenarios and comment on the third in the discussion of future work.

Recall that the parameter $\lambda \in [0,1]$ indicates the user's preference for privacy over lower transaction fees. It is introduced to make those objectives comparable in the cost function. We instantiate the cost function c as a weighted sum of transaction fees and a penalty of one unit if the coin selection decisions *link* both transactions. Transaction fees are modeled as the number of UTXOs (transaction inputs) spent in a transaction. We call two individual transactions *linked* if their inputs refer to the same address(es) either by spending UTXOs from an identical address or by spending the change output. Note that our conceptualization of privacy exactly follows the state-of-the-art multi-input and change address heuristics proposed in [23] and used to de-anonymize users. With this simplistic cost function, we can formulate the optimization model for myopic, strategic as well as for heuristic coin selection.

Definition 6 (Myopic coin selection) *In each period t_i, the myopic user solves the following optimization problem:*

$$\begin{aligned} \min_{S_i} \quad & |S_i| \\ s.t. \quad & n_i \le \sum_{j \in S_i} v_j \qquad \forall i \in \{1,2\}. \end{aligned} \tag{4}$$

So, the myopic user does not explicitly care about privacy and optimizes for lower fees in each transaction. The strategic user, on the contrary, optimizes concurrently for transaction fees and financial privacy based on the complete knowledge about both transaction needs.

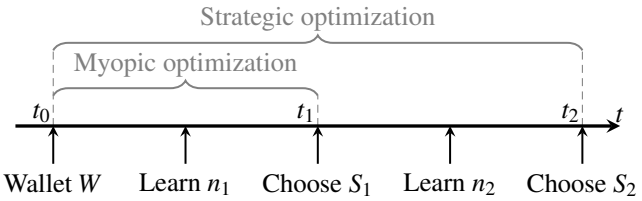

Figure 3: Decision timeline of the coin selection process

Definition 7 (Strategic coin selection) *The strategic user solves the following optimization problem:*

$$\begin{aligned} \min_{S_1,S_2} \quad & (1-\lambda)\cdot|S_1 \cup S_2| + \lambda \cdot \delta\big(A_1 \cap A_2 \neq \emptyset \vee s\big(\underbrace{(a_h, v_h)}_{change}\big) = t_2\big) \\ s.t. \quad & n_i \le \sum_{j \in S_i} v_j \quad \forall i \in \{1,2\} \end{aligned} \tag{5}$$

where $\delta(x)$ is the Kronecker delta function that returns 1 if the condition x holds or 0, otherwise.

As it follows from Eq. (5), privacy is defined in our model in a narrow sense as both transactions being unlinked to each other. In the analysis below, we use the cost function of strategic coin selection to compare and measure the disadvantages of myopic decisions.

Besides myopic and strategic coin selection, we give a general definition of heuristic-based coin selection which applies to a single transaction only.

Definition 8 (Heuristic coin selection) *Let $\big((a_1, v_1), \ldots, (a_m, v_m)\big)$ be the ordered set of all UTXOs contained in the wallet W. Heuristic coin selection solves the following opti-*

mization problem:

$$\min_{k_i \in \{1,\ldots,m\}} k_i$$
$$s.t. \quad n_i \leq \sum_{j=1}^{j=k} v_j \qquad \forall i \in \{1,2\}. \tag{6}$$

The LVF heuristic requires UTXOs to be ordered ascendingly by their values (i. e., $v_1 \leq \ldots \leq v_m$), whereas the HVF heuristic assumes UTXOs to be ordered in a descending manner (i. e., $v_1 \geq \ldots \geq v_m$).

6 The Combinatorial Approach

Many combinatorial optimization problems exhibit special difficulties and often cannot be solved by standard analytical methods. The coin selection problem suffers from these limitations, too, due to its large search space of choices. In this paper, we propose a specific combinatorial approach to solving the optimization model. Since we are interested in finding general results, our approach heavily relies on computer algebra tools and the z3 SMT solver [12] (via the API in Python). We symbolically encode the transaction needs and values of the UTXOs and perform symbolic computations and checks over the model's parameters. Being limited by the exponential complexity of our algorithm, we present results for two computationally tractable instances of the model. More specifically, we analyze and compare myopic and strategic coin selection for the two wallet configurations of the size $m = 2$ and $m = 3$. Section 6.1 presents the algorithm at the basis of our approach, whereas Section 6.2 reports on the results obtained.

6.1 Algorithm

In a nutshell, the approach is based on an exhaustive grid search over the parameter space, determined by the relevant constraints. Geometrically, a funding constraints for each elementary choice defines a polyhedron in $(m+2)$-dimensional space. Under the common convention that the sum of all UTXOs contained in the wallet can be normalized to one, we can view these bounded polyhedrons as polytopes. The polytopes may overlap with each other and generate volumes in the space in which two or more elementary choices are feasible. Our task is to detect all such volumes in the space and evaluate all possible elementary choices within each region in order to select the optimal one(s). Apart from the optimality criterion, we can also check whether the user can choose from several (optimal) alternatives, or whether there are such choice(s) which do not link both transactions, thereby preserving user privacy. So, the idea behind the method is to dissect the integral polytope defined by the funding constraints into smaller non-intersecting volumes, and for each identified partition and each elementary choice feasible in it, evaluate the cost function.

Figure 4 depicts the workflow of the combinatorial approach. For the sake of obtaining general results, it leverages computer algebra and satisfiability checking tools in order to work on the symbolic level rather than on numerical realizations.

Note that coin selection is a combinatorial problem in itself, meaning that the decision depends on the concrete values of UTXOs and transaction needs. Addresses, which act as attributes of UTXOs, are relevant in the analysis of privacy implications of coin selection decisions. Therefore, our approach first solves the combinatorial part of the problem (in the *upper* part of Figure 4). Then, it proceeds with the evaluation of results for each possible assignment of UTXOs to addresses (in the *lower* part of Fugure 4).

Algorithm 1 shows pseudocode for finding the set of active choice profiles. First, it creates an array of m symbolic variables, referring to the values of UTXOs. Then, it proceeds with the generation of a set of the elementary choices (in *lines 9–17*). Each combination of the elementary choices is extended with the two funding constraints expressed as symbolic linear inequalities. The second part of Algorithm 1 adopts a top-down approach to the exhaustive search by generating all possible choice profiles (array $profiles$) and checking whether each profile is feasible in the parameter space. The key task is to identify which choice profiles have support in the parameter space given the respective funding constraints of the elementary choices. To this end, we take advantage of efficient search capabilities of a satisfiability modulo theories (SMT) solver.

In general, SMT solvers are used to check a satisfiability of a first-order logic formula with respect to some logical theory [5]. Therefore, the main prerequisite for the use of a SMT solver is to formalize the problem using first-order logic formulas and linear arithmetic. In our case, the input formula is constructed as a conjunction of q boolean terms (refer to *lines 20–27*). For each set bit in the choice profile p, the term is a conjunction of two funding constraints defined for a respective elementary choice. For each zero bit in the choice profile, the term is a negated conjunction of the funding constraints. In addition, we extend the SMT model with some basic assumptions which impose an ascending order relation on the m symbolic variables and state that all variables must be strictly positive (in *line 28*). Once the SMT model is specified, we call the z3 SMT solver to check for satisfiability. If the solver returns a successful result (*sat*), then we know that there is at least one region in the parameter space, in which the tested choice profile has support. The algorithm adds this binary vector to the array $ActiveProfiles$ which collects the identified active choice profiles.

In principle, we could now determine the volume of each region (given by its *H-representation*) in the parameter space in order to calculate a probability weight for each satisfiable choice profile. This is necessary for the scenario with incomplete information where users form expectations about the

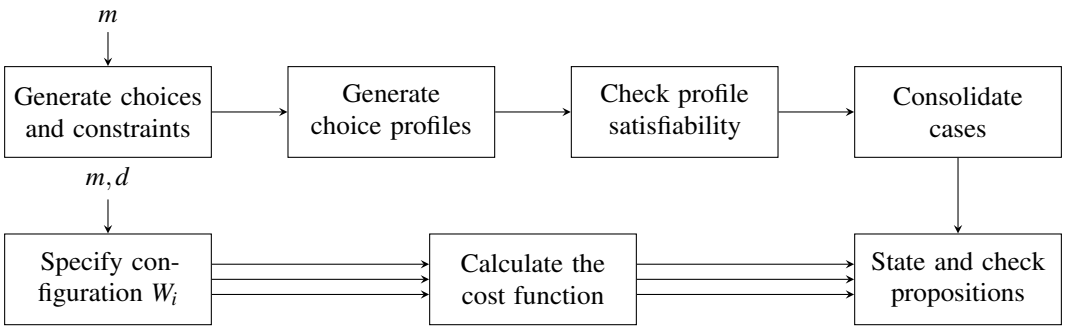

Figure 4: Workflow of the combinatorial approach

transaction need in the second period n_2. Recall that we excluded this scenario from the analysis in this work, hence we instead count the number of connected satisfiable regions for our evaluation of the scenario with complete information. Since a choice profile is defined by a logical expression of linear constraints, it may refer to one or more non-convex or unconnected regions in the multidimensional parameter space. To differentiate between these volumes, we introduce the notion of a *case*, defined as follows:

Definition 9 (Case) *A* case *is a connected region in the parameter space where for every fixed preference* λ*, the cost function as defined in Equation 5 is a mapping from a fixed set of feasible elementary choices to constants.*

The number of distinct cases for each satisfiable choice profile can be obtained by transforming the input formula of the SMT model to its disjunctive normal form (DNF). Each component of this boolean expression bounds a certain region in the multidimensional space and falls under our definition of the case. As a final step, we can calculate the cost function for all feasible elementary choices in a profile for every possible assignment of m UTXOs to d addresses.

6.2 Analysis

We demonstrate the application of the proposed approach on two wallets of the size $m = 2$ and $m = 3$. As it follows from Equation 1, these two configurations lead in total to $(2^7 - 1)$ and $(2^{31} - 1)$ possible choice profiles. By running the z3 SMT solver on a high-performance computing cluster, we managed to find 11 and 93 active choice profiles. The representation of the corresponding logic formulas in DNF reveals that each active choice profile corresponds to exactly one case. Tables 5, 6 and **??** in Appendix B list the elements of *choices* and $ActiveProfiles$ for both scenarios.

We discuss the results of our analysis by formulating a number of propositions that compare privacy-preserving and non-privacy-preserving decisions as well as strategic and myopic coin selection. We also demonstrate that our analysis is useful to evaluate common heuristics by comparing the performance of the LVF and HVF rules (see Sect. 4.1). The evaluation of active choice profiles and the search for optimal solutions are executed for every possible assignment of m UTXOs to d addresses. Therefore, Configuration 1 ($m = 2$) has two sub-configurations, referring to $W_1^1 = \{a_1, a_2\} \Rightarrow utxos = \{(a_1, v_1), (a_2, v_2)\}$ and $W_2^1 = \{a_1\} \Rightarrow utxos = \{(a_1, v_1), (a_1, v_2)\}$. Configuration 2 ($m = 3$) has the following five sub-configurations:

1. $W_1^2 = \{a_1, a_2, a_3\} \Rightarrow utxos = \{(a_1, v_1), (a_2, v_2), (a_3, v_3)\}$
2. $W_2^2 = \{a_1, a_2\} \Rightarrow utxos = \{(a_1, v_1), (a_1, v_2), (a_2, v_3)\}$
3. $W_3^2 = \{a_1, a_2\} \Rightarrow utxos = \{(a_1, v_1), (a_2, v_2), (a_1, v_3)\}$
4. $W_4^2 = \{a_1, a_2\} \Rightarrow utxos = \{(a_1, v_1), (a_2, v_2), (a_2, v_3)\}$
5. $W_5^2 = \{a_1\} \Rightarrow utxos = \{(a_1, v_1), (a_1, v_2), (a_1, v_3)\}$.

Note that while the wallets in sub-configurations 2–4 contain two addresses, they differ from each other in the way the UTXOs of different value order are assigned to these addresses.

Proposition 1 (Privacy at the Core) *There exist cases where strategic users can preserve their privacy, regardless of the fee minimization constraint.*

This proposition (and some following ones) does not apply to the wallets W_2^1 and W_5^2, which have one address only and therefore do not allow users to preserve their privacy. Evidently, the highest number of privacy-preserving cases (5 out of 11 and 80 out of 93) exist for the wallets W_1^1 and W_1^2. As far as the wallets W_2^2–W_4^2 are concerned, the higher level of flexibility in terms of the number of available options is achieved in W_4^2. Thus, holding UTXOs of moderately small, rather than large, values in the same address is recommended from the privacy-oriented standpoint.

Proposition 2 (Privacy Over Fees) *There exist cases where, for* $\lambda \neq 0$*, the privacy-preserving choice is Pareto-optimal.*

Algorithm 1 Find Active Profiles

m: numerical value of the size of the wallet
n_1, n_2: symbolic values of the transaction needs
v_h: symbolic value of the change output
procedure FINDACTIVEPROFILES(m, n_1, n_2, v_h)
    $choices \leftarrow \varnothing$ ▷ an empty array of quadruples
    $ActiveProfiles \leftarrow \varnothing$ ▷ an empty array of binary vectors
    $utxos \leftarrow symbols(m)$ ▷ generate an array of symbolic variables $(v_1, \ldots, v_m)$
    $definitions \leftarrow v_1 \leq v_2 \leq \ldots \leq v_m \wedge v_i > 0 \; \forall i \in \{1 \ldots m\} \wedge n_j > 0 \; \forall j \in \{1,2\}$
    $option_1 \leftarrow \texttt{PowerSet}(utxos) \setminus \varnothing$ ▷ generate all coin selection options in t_1
    for $(x \in option_1)$ **do**
        $unspent \leftarrow utxos \setminus x$ ▷ apply the combinatorial constraints
        $option_2 \leftarrow \texttt{PowerSet}(unspent \cup v_h) \setminus \varnothing$ ▷ generate all coin selection options in t_2
        for $(y \in option_2)$ **do**
            $constraint_1 \leftarrow n_1 \leq \sum x$ ▷ the sum operator generates a symbolic expression summing over the elements in x
            $v_h \leftarrow \sum x - n_1$ ▷ replace v_h with its symbolic expression
            $constraint_2 \leftarrow n_2 \leq \sum y$ ▷ the sum operator generates a symbolic expression summing over the elements in y
            $choices \leftarrow choices \cup (x, y, constraint_1, constraint_2)$
    $profiles \leftarrow \{0,1\}^{|choices|}$ ▷ generate a set of all choice profiles
    for $(p \in profiles)$ **do**
        $formula \leftarrow true$ ▷ initialize the logical formula
        for $i \in \{1, \ldots, |choices|\}$ **do**
            $constraint_1 \leftarrow choices_i[constraint_1]$ ▷ get the linear inequality for t_1
            $constraint_2 \leftarrow choices_i[constraint_2]$ ▷ get the linear inequality for t_2
            if $(p_i == 1)$ **then**
                $formula \leftarrow formula \wedge (constraint_1 \wedge constraint_2)$
            else
                $formula \leftarrow formula \wedge \neg(constraint_1 \wedge constraint_2)$
        $status \leftarrow \texttt{CallSMTSolver}(\texttt{Assert}(definitions \wedge formula))$ ▷ define the model and call the SMT solver
        if $(status == sat)$ **then**
            $ActiveProfiles \leftarrow ActiveProfiles \cup p$
    return $ActiveProfiles$

For $\lambda = 0$, strategic users have two weakly Pareto-optimal choices, one of which preserves privacy and the other minimizes transaction fees.

This proposition implies that strategic users make the optimal choice and optimize for both privacy and transaction fees. Table 1 reports the number of such cases for each wallet sub-configuration.

Table 1: Number of cases for Proposition 2

W_1^1	W_2^1	W_1^2	W_2^2	W_3^2	W_4^2	W_5^2
4	na	42	41	49	38	na

Proposition 3 (Privacy vs. Fees) *There exist cases where strategic users have a choice of optimizing either for privacy or for transaction fees (depending on their preference $\lambda \in [0,1]$).*

There are three such cases in each of the wallets W_4^2 and W_5^2. These cases refer to the choice profiles indexed by 5EB7, 7C37 and 7EB7 in Table 6 in Appendix B. Depending on the preference λ, the strategic user decides either in favor of privacy by selecting more UTXOs in both transactions or, alternatively, in favor of less UTXOs of the same address and, consequently, lower fees. We specify the parameter space referring to these profiles in the following lemma.

Lemma 2 *If $\lambda > \frac{1}{2}$, the strategic user chooses the privacy-preserving Pareto-optimal solution and pays higher transaction fees. If $\lambda < \frac{1}{2}$, the strategic user chooses the fee-minimizing Pareto-optimal solution, but links both transactions. If $\lambda = \frac{1}{2}$, the strategic user has two weakly Pareto-optimal solutions. These propositions apply to the following*

intervals of transaction needs (where $v_1 + v_2 \leq v_3$):

$$v_2 \leq t_1 \leq v_1 + v_2, \quad v_1 + v_2 \leq t_2 \leq v_3 - v_2$$
$$v_2 \leq t_1 \leq v_1 + v_2, \quad v_2 \leq t_2 \leq \min\{v_1 + v_2, v_3 - (v_1 + v_2)\}$$
$$v_1 + v_2 \leq t_1 \leq v_3, \quad v_2 \leq t_1 \leq \min\{v_1 + v_2, v_3 - v_2\}$$

The proof (omitted for brevity) follows from the logical union and subsequent simplification of the constraints. The lemma illustrates that the characterization of the feasible regions can be obtained directly from the constraints of the active choice profiles (and possibly be simplified).

Proposition 4 (Effects of Short-Term Horizon) *There exist cases where myopic users are worse off than strategic users optimizing for privacy ($\lambda = 1$).*

Myopic coin selection generally results in a higher variability of the choices available to the user, since the choice made in the first period is oftentimes not deterministic (e. g., in some cases, the user may spend in the first period any of the individual UTXO contained in the wallet). In Table 2, we report in how many cases the myopic user ***always*** makes a non-privacy-preserving decision (i. e., ***all*** myopic alternative choices do not preserve privacy, though there exists at least one privacy-preserving choice feasible in that choice profile). As opposed to this outcome, we also specify in how many cases the myopic user ***always*** manages to preserve her privacy.

Table 2: Privacy effects of myopic coin selection

Criterion	W_1^1	W_2^1	W_1^2	W_2^2	W_3^2	W_4^2	W_5^2
No privacy	0	0	6	6	0	0	0
Privacy	1	na	21	12	7	1	na

Our approach also allows us to compare the performance of the LVF and HVF heuristics by using strategic optimization as a benchmark. To this end, we state first two lemmas concerning limitations of each heuristic in comparison to the optimal solution and, based on these results, infer a proposition on the evaluation of the LVF versus HVF rule.

Lemma 3 (Lowest Value First Rule) *There exist cases where users following the LVF rule are worse off than strategic users optimizing either for privacy ($\lambda = 1$) or for transaction fees ($\lambda = 0$).*

First, we report the results for Configuration 1. With respect to the wallet W_1^1, the LVF heuristic links both transactions in all cases, as it always selects the change output (a_h, v_h) for the second-period transaction. In the cases indexed by 7, B, and F, this heuristic spends both UTXOs and the change output, although there is a cheaper choice available that spends only one UTXO in each transaction. This is valid for both wallets W_1^1 and W_2^1, whereas the cheaper choice in the cases B and F for W_1^1 would allow the user to preserve privacy, too.

Table 3 presents the performance results of the LVF heuristic compared to privacy-aware coin selection ($\lambda = 1$) for Configuration 2. Recall that under our definition of privacy, the wallet W_5^2 does not allow for any privacy-preserving decision. The first row of Table 3 reports the number of cases in which the LVF rule links both transactions, although there is at least one privacy-preserving choice available in the choice profile. In the second row, we report the number of cases in which the choice dictated by the LVF rule coincides with the choice of privacy-aware coin selection. Note that this choice may be privacy-preserving as well as not privacy-preserving, depending on a particular wallet and a profile. The special case refers to the choice profile, in which the LVF rule dictates either a privacy-preserving (when $v_3 \leq \frac{(v_1+v_2)}{2}$) or a not privacy-preserving (when $v_3 \geq \frac{(v_1+v_2)}{2}$) selection of UTXOs. In terms of transaction fees, the application of the LVF heuristics results in 78 cases for each wallet, in which the user needs to pay higher fees than optimal.

Table 3: Evaluation of LVF in terms of privacy

Criterion	W_1^2	W_2^2	W_3^2	W_4^2	W_5^2
LVF worse off than $\lambda = 1$.	67	65	52	65	na
LVF coincides with $\lambda = 1$.	25	27	41	28	na
Special case	1	1	0	0	na

Lemma 4 (Highest Value First Rule) *There exist cases where users following the HVF rule are worse off than strategic users optimizing for privacy ($\lambda = 1$) or transaction fees ($\lambda = 0$).*

We start with the results for Configuration 1. With respect to the wallet W_1^1, the HVF rule coincides with privacy-oriented coin selection in the cases B and 3B. In the cases F, 3F, and 7F, this heuristic may compromise user privacy, since it dictates either a privacy-preserving (when $v_1 \geq \frac{v_2}{2}$) or a non-privacy-preserving (when $v_1 \leq \frac{v_2}{2}$) selection of UTXOs. In terms of transaction fees, the HVF heuristic is worse off in the single case 33.

Table 4 reports the performance results of the HVF heuristic compared to privacy-oriented coin selection ($\lambda = 1$) for Configuration 2. Note that there are more special cases due to a larger number of different order relations of the values v_2, v_1, and v_h. In terms of transaction fees, the HVF rule is worse than fee-aware coin selection ($\lambda = 0$) in 14 cases for each wallet.

Proposition 5 *Based on the performance results reported in Lemmas 3 and 4, the HVF heuristic is better in terms of transaction fees and user privacy than the LVF heuristic.*

The LVF rule often selects the change output and thereby compromises user privacy. However, our model does not consider the blockchain size, the other relevant aspect of the coin

Table 4: Evaluation of HVF in terms of privacy

Criterion	W_1^2	W_2^2	W_3^2	W_4^2	W_5^2
HVF worse off than $\lambda = 1$.	26	21	34	31	na
HVF coincides with $\lambda = 1$.	34	45	44	56	na
Special cases	33	27	15	6	na

selection trade-off. From a collective perspective, the LVF rule would be more favorable for the network, as it reduces the size of the UTXO set and the associated maintenance overhead.

7 Related Work

We organize our review of related work on cryptocurrencies along the following lines of research: coin selection, transaction management, and privacy. The closest papers on coin selection to our own are [15] and [29]. In his seminal work, Erhardt [15] compares several policies implemented in some popular wallets and evaluates their effects on transaction fees and the size of the UTXO pool. Drawing on numerical simulation results, the author suggests some improvements to the Bitcoin Core's algorithm, which were endorsed and adopted by the community of developers. The work [29] models coin selection as an optimization of either transaction fees or the UTXO set. The authors evaluate their models on real transactions collected for a short time span from the Bitcoin network. In comparison to these studies, our work looks at the coin selection problem through theoretical lenses and elaborates on its formal modeling. Furthermore, our work stands out by its intertemporal view on coin selection, its explicit consideration of the coin selection implications on user privacy, and the use of computer algebra tools and symbolic computations instead of numerical simulation and observations.

Research on transaction management in cryptocurrencies covers a broad spectrum of topics. Based upon economic models or empirical evidences, a number of papers studies transaction fees and pricing mechanisms for cryptocurrencies [14, 21, 24]. The work [3] defines a formal model of Bitcoin transactions and proves basic transaction properties. More recent works [13, 31] analyze the evolution of the UTXO set of Bitcoin and its closest clones in terms of the profitability of spending a single UTXO. Drawing on the insightful results, the authors of [31] emphasize the need for designing proper coin selection strategies in order to avoid potential flooding of the UTXO set with many unprofitable UTXOs. Our work seeks to fill this identified gap.

In terms of privacy, there is an extensive body of literature on the address clustering and money flow tracking in cryptocurrencies ([2,23,33,34] in Bitcoin, [20,27] in Monero, [18] in ZCash). Most of these empirical analyses follow a similar research design: based on certain technical features or behavioral patterns, a set of meaningful heuristics are applied to de-anonymize vulnerable transactions or addresses. Besides, some recent studies on privacy-centric coins (e. g., [18, 27]) have revealed that cryptographic privacy features may be in vain because wallets make bad trade-offs and configurations for users. Our work relates to these findings and emphasizes that modern coin selection heuristics should be revised in users' best interests. Other studies look closer at anonymity techniques of mixing services [25, 26] or analyze anonymity vulnerabilities and countermeasures at the peer-to-peer layer that break a link between users' IP addresses and their public keys and transactions [6, 16, 19]. Our work is fundamental to the latter approaches in the sense that privacy-oriented techniques at the network level may simply be futile if coin selection is done blindfold in the first place.

Finally, our method is related to works that employ computer algebra for studying large combinatorial and number theoretic problems. The idea of coupling state-of-the-art symbolic computation tools with SMT solvers is relatively new, being first introduced in 2017 in [37]. The recent work [8] leverages the capabilities of this combined method for enumerating Williamson matrices of even order. In a related stream of research, the work [9] uses a Bayes–Nash equilibrium search algorithm for finding optimal core-selecting payment rules for combinatorial auctions.

8 Conclusion

At the core of this paper is a new model to systematically reason about coin selection in cryptocurrencies. In comparison to other related works, the model captures transactions fees and, as a novel feature, user privacy. It considers two periods, which allows us to compare the outcome for myopic and strategic users, for now in the complete information regime. To deal with the complexity of coin selection, we proposed an uncommon solution approach leveraging symbolic computation and SMT solvers. The validity of this approach is demonstrated by solving two tractable instances of the problem, each with various wallet configurations. This allowed us to support relevant propositions with evidence. For example, to the best of our knowledge, we are the first to evaluate the performance of the LVF and HVF heuristics, which are both used in practice. Large parts of our work generalize to all UTXO-based cryptocurrencies; specifically, to the ones that refer to unique UTXOs (e. g., Bitcoin, its forks, and unshielded transactions in ZCash). Conceivably, the model may also fit to the operation of Monero by adapting the conceptualization of privacy in the cost function.

There remain a number of limitations in the present model and approach that deserve attention in further research. First, our analysis relies on the assumption of complete information about future transaction needs. This can be relaxed by extending the model with expectations formed by the user over future payments. Second, the cost function is simplistic and

can be criticized for its disparity with reality. In that sense, our results are approximations where arguably the approximation error of the privacy component exceeds the error of the simplified fee rule. While the latter can be refined, we are less optimistic that it will be possible to measure heterogeneous users' privacy preferences much more precisely than in the current form. Third, our results apply to small wallets only. This limitation is less severe than it may sound. The unique ground-truth data of about 37 585 Bitcoin wallets (obtained from studying the Bloom filter leak [30]) allows us to conjecture that the analyzed configurations are, in fact, representative. In particular, around 70 % of the wallets analyzed in [30] have had the cardinality no more than two addresses. Our own analysis of the current set of standard UTXOs in Bitcoin (50 217 821 UTXOs in total[7]) lends further support: 86 % and 6 % of all addresses have one and two UTXOs, respectively. Nevertheless, future work should explore ways to improve the search strategy in order to apply the approach to larger instances of the coin selection problem. Perhaps, it is possible to find new domain-specific features of the problem definition that lead to a reduction of the search space.

While it appears tempting to suggest that future cryptocurrency wallet software should implement our approach (or a refined version thereof), we do not expect this in the short run. This not only hinges on the runtime complexity even for small wallets (we decided more than two billion SMT problems), but on the difficulty of collecting the users' expectations about future transaction needs, or predicting them from data. However, what wallet software could and should do is to allow the user to choose the objective as simple as by selecting between lower fees or better privacy. This will ensure that wallets work more often in their user's interest rather than against it. For both objectives, we expect that practical coin selection will still remain a heuristic process. This work paves the ground to evaluate and improve these heuristics.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 740558.

References

[1] Alessandro Acquisti and Hal R. Varian. Conditioning Prices on Purchase History. *Marketing Science*, 24(3):367–381, 2005.

[2] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating User Privacy in Bitcoin. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 34–51. Springer, Berlin Heidelberg, 2013.

[3] Nicola Atzei, Massimo Bartoletti, Stefano Lande, and Roberto Zunino. A formal model of Bitcoin transactions. In Sarah Meiklejohn and Kazue Sako, editors, *Financial Cryptography and Data Security*, pages 1–19, 2018.

[4] Khaled Baqer, Danny Yuxing Huang, Damon McCoy, and Nicholas Weaver. Stressing Out: Bitcoin "Stress Testing". In Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security*, pages 3–18, Berlin Heidelberg, 2016. Springer.

[5] Clark Barrett and Cesare Tinelli. Satisfiability modulo theories. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 305–343. Springer International Publishing, Cham, 2018.

[6] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of Clients in Bitcoin P2P Network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 15–29, New York, NY, USA, 2014. ACM.

[7] Blockchain. Confirmed Transactions Per day – Blockchain, 2019. Retrieved February, 4 2020 from https://www.blockchain.com/charts/n-transactions?timespan=all.

[8] Curtis Bright, Ilias Kotsireas, and Vijay Ganesh. A SAT+CAS Method for Enumerating Williamson Matrices of Even Order. In *32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018.

[9] Benedikt Bünz, Benjamin Lubin, and Sven Seuken. Designing Core-selecting Payment Rules: A Computational Search Approach. In *19th ACM Conference on Economics and Computation*, pages 109–147, Ithaca, NY, 2018.

[10] Gerard P. Cachon and Robert Swinney. Purchasing, Pricing, and Quick Response in the Presence of Strategic Consumers. *Management Science*, 55(3):497–511, 2009.

[11] David Chaum. Achieving Electronic Privacy. *Scientific American*, pages 96–101, 1992.

[12] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] We have excluded from the analysis multi-signature, non-standard, and data storage (OP_RETURN) UTXOs.

[13] Sergi Delgado-Segura, Cristina Pérez-Solà, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. Analysis of the Bitcoin UTXO set. In *Financial Cryptography and Data Security, 5th Workshop on Bitcoin and Blockchain Research*, Lecture Notes in Computer Science, 2018.

[14] David Easley, Maureen O'Hara, and Soumya Basu. From Mining to Markets: The Evolution of Bitcoin Transaction Fees. SSRN, Report 2018/3055380, 2018. Retrieved February, 4 2020 from https://ssrn.com/abstract=3055380.

[15] Mark Erhardt. An Evaluation of Coin Selection Strategies, 2016. Master thesis, retrieved February, 4 2020 from http://murch.one/wp-content/uploads/2016/11/erhardt2016coinselection.pdf.

[16] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(2):29:1–29:35, June 2018.

[17] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies, 2017. Retrieved February 4, 2020 from https://arxiv.org/abs/1708.04748.

[18] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. An Empirical Analysis of Anonymity in Zcash. In *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, MD, 2018. Retrieved February, 4 2020 from https://www.usenix.org/conference/usenixsecurity18/presentation/kappos.

[19] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 469–485, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[20] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. A Traceability Analysis of Monero's Blockchain. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 153–173, Cham, 2017. Springer International Publishing.

[21] Ron Lavi, Or Sattath, and Aviv Zohar. Redesigning Bitcoin's Fee Market. In *The World Wide Web Conference*, WWW '19, pages 2950–2956, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Silvano Martello and Paolo Toth. A mixture of dynamic programming and branch-and-bound for the subset-sum problem. *Management Science*, 30(6):765–771, 1984.

[23] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, pages 127–140, New York, NY, USA, 2013. ACM.

[24] Malte Möser and Rainer Böhme. Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees. In Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff, editors, *Financial Cryptography and Data Security, 2nd Workshop on BITCOIN Research*, volume 8976 of *Lecture Notes in Computer Science*, pages 19–33, Berlin Heidelberg, 2015. Springer.

[25] Malte Möser and Rainer Böhme. Join Me on a Market for Anonymity. In *Proceedings of the 15th Annual Workshop on the Economics of Information Security*, Berkeley, CA, USA, 2016.

[26] Malte Möser and Rainer Böhme. Anonymous Alone? Measuring Bitcoin's Second-Generation Anonymization Techniques. In *IEEE Security & Privacy on the Blockchain (IEEE S&B)*, Paris, France, 2017.

[27] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. An Empirical Analysis of Traceability in the Monero Blockchain. In *Proceedings on Privacy Enhancing Technologies*, volume 3, pages 143–163, 2018.

[28] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA, 2016.

[29] Van-Huy Nguyen, Hong-Son Trang, Quoc-Thinh Nguyen, Nguyen Huynh-Tuong, and Thanh-Van Le. Building mathematical models applied to UTXOs selection for objective transactions. In *5th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 160–164, 2018.

[30] Jonas David Nick. Data-driven de-anonymization in bitcoin, 2015. Master thesis, retrieved February 4, 2020 from https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/155286/eth-48205-01.pdf.

[31] Cristina Pérez-Solà, Sergi Delgado-Segura, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. Another coin bites the dust: An analysis of dust in UTXO based cryptocurrencies. Cryptology ePrint Archive, Report 2018/513, 2018. Retrieved February 4, 2020 from https://eprint.iacr.org/2018/513.

[32] Jeffrey Quesnelle. On the linkability of Zcash transactions. *arXiv preprint:1712.01210*, 2017. Retrieved February, 4 2020 from https://arxiv.org/abs/1712.01210.

[33] Fergal Reid and Martin Harrigan. An Analysis of Anonymity in the Bitcoin System. In Yaniv Altshuler, Yuval Elovici, B. Armin Cremers, Nadav Aharony, and Alex Pentland, editors, *Security and Privacy in Social Networks*, pages 197–223. Springer, New York, 2013.

[34] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 6–24. Springer, Berlin Heidelberg, 2013.

[35] Bitcoin Wiki. Transaction Fees – Bitcoin Wiki, 2018. Retrieved February, 4 2020 from https://en.bitcoin.it/wiki/Transaction_fees.

[36] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. Tracing Transactions across Cryptocurrency Ledgers. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 837–850, USA, 2019. USENIX Association.

[37] Edward Zulkoski, Curtis Bright, Albert Heinle, Ilias Kotsireas, Krzysztof Czarnecki, and Vijay Ganesh. Combining SAT Solvers with Computer Algebra Systems to Verify Combinatorial Conjectures. *Journal of Automated Reasoning*, 58(3):313–339, 2017.

A Elementary Choices in a Matrix Form

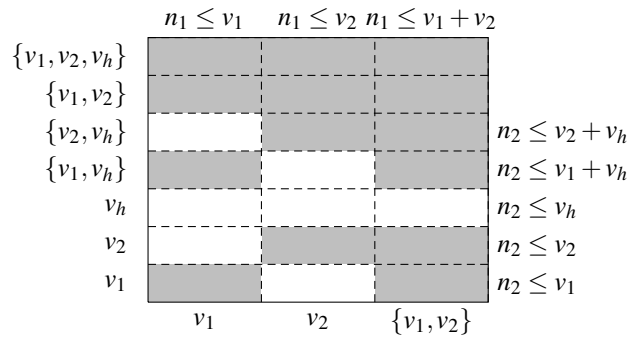

Figure 5: Constrained space of elementary choices in a matrix form. Gray areas are infeasible due to the application of the combinatorial constraints. The funding constraints are shown for each coin selection option (in columns for t_1 and in rows for t_2).

B Elementary Choices and Choice Profiles

Table 5: Elementary choices and funding constraints for the analyzed configurations

m	#	$option_1$	$option_2$	$constraint_1$	$constraint_2$
	1	$\{v_1\}$	$\{v_2\}$	$n_1 \leq v_1$	$n_2 \leq v_2$
	2	$\{v_1\}$	$\{v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 - n_1$
	3	$\{v_1\}$	$\{v_2, v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 + v_2 - n_1$
$m = 2$	4	$\{v_2\}$	$\{v_1\}$	$n_1 \leq v_2$	$n_2 \leq v_1$
	5	$\{v_2\}$	$\{v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_2 - n_1$
	6	$\{v_2\}$	$\{v_1, v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_1 + v_2 - n_1$
	7	$\{v_1, v_2\}$	$\{v_h\}$	$n_1 \leq v_1 + v_2$	$n_2 \leq v_1 + v_2 - n_1$
	1	$\{v_1\}$	$\{v_2\}$	$n_1 \leq v_1$	$n_2 \leq v_2$
	2	$\{v_1\}$	$\{v_3\}$	$n_1 \leq v_1$	$n_2 \leq v_3$
	3	$\{v_1\}$	$\{v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 - n_1$
	4	$\{v_1\}$	$\{v_2, v_3\}$	$n_1 \leq v_1$	$n_2 \leq v_2 + v_3$
	5	$\{v_1\}$	$\{v_2, v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 + v_2 - n_1$
	6	$\{v_1\}$	$\{v_3, v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 + v_3 - n_1$
	7	$\{v_1\}$	$\{v_2, v_3, v_h\}$	$n_1 \leq v_1$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	8	$\{v_2\}$	$\{v_1\}$	$n_1 \leq v_2$	$n_2 \leq v_1 + v_2 - n_1$
	9	$\{v_2\}$	$\{v_3\}$	$n_1 \leq v_2$	$n_2 \leq v_2 + v_3 - n_1$
	10	$\{v_2\}$	$\{v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_2 - n_1$
	11	$\{v_2\}$	$\{v_1, v_3\}$	$n_1 \leq v_2$	$n_2 \leq v_1 + v_3$
	12	$\{v_2\}$	$\{v_1, v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_1 + v_2 - n_1$
	13	$\{v_2\}$	$\{v_3, v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_2 + v_3 - n_1$
	14	$\{v_2\}$	$\{v_1, v_3, v_h\}$	$n_1 \leq v_2$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	15	$\{v_3\}$	$\{v_1\}$	$n_1 \leq v_3$	$n_2 \leq v_1$
$m = 3$	16	$\{v_3\}$	$\{v_2\}$	$n_1 \leq v_3$	$n_2 \leq v_2$
	17	$\{v_3\}$	$\{v_h\}$	$n_1 \leq v_3$	$n_2 \leq v_3 - n_1$
	18	$\{v_3\}$	$\{v_1, v_2\}$	$n_1 \leq v_3$	$n_2 \leq v_1 + v_2$
	19	$\{v_3\}$	$\{v_1, v_h\}$	$n_1 \leq v_3$	$n_2 \leq v_1 + v_3 - n_1$
	20	$\{v_3\}$	$\{v_2, v_h\}$	$n_1 \leq v_3$	$n_2 \leq v_2 + v_3 - n_1$
	21	$\{v_3\}$	$\{v_1, v_2, v_h\}$	$n_1 \leq v_3$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	22	$\{v_1, v_2\}$	$\{v_3\}$	$n_1 \leq v_1 + v_2$	$n_2 \leq v_3$
	23	$\{v_1, v_2\}$	$\{v_h\}$	$n_1 \leq v_1 + v_2$	$n_2 \leq v_1 + v_2 - n_1$
	24	$\{v_1, v_2\}$	$\{v_3, v_h\}$	$n_1 \leq v_1 + v_2$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	25	$\{v_1, v_3\}$	$\{v_2\}$	$n_1 \leq v_1 + v_3$	$n_2 \leq v_2$
	26	$\{v_1, v_3\}$	$\{v_h\}$	$n_1 \leq v_1 + v_3$	$n_2 \leq v_1 + v_3 - n_1$
	27	$\{v_1, v_3\}$	$\{v_2, v_h\}$	$n_1 \leq v_1 + v_3$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	28	$\{v_2, v_3\}$	$\{v_1\}$	$n_1 \leq v_2 + v_3$	$n_2 \leq v_1$
	29	$\{v_2, v_3\}$	$\{v_h\}$	$n_1 \leq v_2 + v_3$	$n_2 \leq v_2 + v_3 - n_1$
	30	$\{v_2, v_3\}$	$\{v_1, v_h\}$	$n_1 \leq v_2 + v_3$	$n_2 \leq v_1 + v_2 + v_3 - n_1$
	31	$\{v_1, v_2, v_3\}$	$\{v_h\}$	$n_1 \leq v_1 + v_2 + v_3$	$n_2 \leq v_1 + v_2 + v_3 - n_1$

Table 6: List of active choice profiles obtained by running the SMT solver

$ActiveProfiles$	Hex value	$ActiveProfiles$	Hex value
Configuration 1 $(m=2, q=7)$			
(0000001)	1	(0110011)	33
(0000011)	3	(0110111)	37
(0000111)	7	(0111011)	3B
(0001011)	B	(0111111)	3F
(0001111)	F	(1111111)	7F
(0010011)	13		
Configuration 2 $(m=3, q=31)$			
(00000000000000000000000000000001)	1	(00000000000000111111111011111111)	1FEFF
(00000000000000000000000000000011)	3	(00000000000000111111111111111111)	1FFFF
(00000000000000000000000000000111)	7	(00000000000001000000010010010011)	20493
(00000000000000000000000000001011)	B	(00000000000011000000110010010111)	60C97
(00000000000000000000000000001111)	F	(00000000001001000000010010010011)	120493
(00000000000000000000000000010011)	13	(00000000001001000100100100100011)	122493
(00000000000000000000000001010011)	53	(00000000001011000001100100100111)	160C97
(00000000000000000000000001010111)	57	(00000000001011000101100100100111)	162C97
(00000000000000000000000001011011)	5B	(00000000101011000001110100100111)	560E97
(00000000000000000000000001011111)	5F	(00000000101011000011110101101111)	561EB7
(00000000000000000000000001111111)	7F	(00000000101011000101110100100111)	562E97
(00000000000000000000000010010011)	93	(00000000101011000111110101101111)	563EB7
(00000000000000000000001010010011)	293	(00000000101011001011110101101111)	565EB7
(00000000000000000000001011010011)	2D3	(00000000101011001111110101101111)	567EB7
(00000000000000000000001011010111)	2D7	(00000000101011010101110110101111)	56AED7
(00000000000000000000001011011011)	2DB	(00000000101011010111110111101111)	56BEF7
(00000000000000000000001011011111)	2DF	(00000000101011011111110111101111)	56FEF7
(00000000000000000000001011111111)	2FF	(00000000101111010111111111101111)	5EBFF7
(00000000000000000000001111111111)	3FF	(00000000101111011111111111101111)	5EFFF7
(00000000000000000000010000010011)	413	(00000000111111011111111111101111)	7EFFF7
(00000000000000000000100100100011)	493	(00000001101111110111111111111111)	DFBFFF
(00000000000000000000011010010011)	693	(00000001101111111111111111111111)	DFFFFF
(00000000000000000000110000010111)	C17	(00000001111111111111111111111111)	FFFFFF
(00000000000000000000111010010111)	E97	(00000010000001000000010010010011)	1020493
(00000000000000000001110000110111)	1C37	(00010010000001000000010010010011)	9020493
(00000000000000000001111010110111)	1EB7	(00010010000011000000110010010111)	9060C97
(00000000000000000010010000010011)	2413	(00010010001001000000010010010011)	9120493
(00000000000000000010010010010011)	2493	(00010010001001000100100100100011)	9122493
(00000000000000000010011010010011)	2693	(00010010001011000001100100100111)	9160C97
(00000000000000000010110000010111)	2C17	(00010010001011000101100100100111)	9162C97
(00000000000000000010111010010111)	2E97	(00010110001011000011100101101111)	B161CB7
(00000000000000000011110000110111)	3C37	(00010110001011000111100101101111)	B163CB7
(00000000000000000011111010110111)	3EB7	(00011110001111000111101101101111)	F1E3DB7
(00000000000000000101110000110111)	5C37	(01010110101011000011110101101111)	2B561EB7
(00000000000000000101111010110111)	5EB7	(01010110101011000111110101101111)	2B563EB7
(00000000000000000111110000110111)	7C37	(01010110101011001011110101101111)	2B565EB7
(00000000000000000111111010110111)	7EB7	(01010110101011001111110101101111)	2B567EB7
(00000000000000001010110001010111)	AC57	(01011110101111000111111101101111)	2F5E3FB7
(00000000000000001010111011010111)	AED7	(01011110101111001111111101101111)	2F5E7FB7
(00000000000000001011110001110111)	BC77	(11011110101111010111111111101111)	6F5EBFF7
(00000000000000001011111011110111)	BEF7	(11011110101111011111111111101111)	6F5EFFF7
(00000000000000001111110001110111)	FC77	(11011110111111011111111111101111)	6F7EFFF7

Table 6: List of active choice profiles obtained by running the SMT solver

$ActiveProfiles$	Hex value	$ActiveProfiles$	Hex value
(00000000000000011111111011110111)	FEF7	(11011111101111110111111111111111)	6FDFBFFF
(00000000000000011011110001111111)	1BC7F	(11011111101111111111111111111111)	6FDFFFFF
(00000000000000011011110111111111)	1BEFF	(11011111111111111111111111111111)	6FFFFFFF
(00000000000000011011111111111111)	1BFFF	(11111111111111111111111111111111)	7FFFFFFF
(00000000000000011111110001111111)	1FC7F		