

An analysis of dust in UTXO based cryptocurrencies

Cristina Pérez-Solà, **Sergi Delgado-Segura**, Guillermo
Navarro-Arribas, Jordi Herrera-Joancomartí

Departament d'Enginyeria de la Informació i les Comunicacions
Universitat Autònoma de Barcelona

October 6th, 2018

1 Introduction

2 Definitions

3 Results

4 Conclusions

Introduction

A **UTXO** is a transaction output that has not been spent yet.

When we talk about bitcoins we are actually referring to UTXOs.

The **UTXO set** is where all UTXOs are stored. We can see it as wallet that includes all unspent bitcoins. No matter their type, *owner* nor value.



Properties of the UTXO set

- It is part of every full node.
- The Bitcoin value of a UTXO does not affect its size (bigger value != bigger size).
- In general, the larger the output script of a UTXO, the more space it occupies in the set.

Goals and Motivation

Goals

- How many unspent outputs are actually worth spending?
- How much space devoted to not-worth-spending outputs is every full node storing?

1 Introduction

2 Definitions

- Dust
- Unprofitable

3 Results

4 Conclusions

Outputs worth spending

How do we know if an output is worth spending?

Outputs worth spending

It depends on two factors:

- How much data such output contributes to a new transaction
- Which is the fee rate we need (or want) to pay

Bitcoin Core dust definition

Dust

Bitcoin Core defines **dust** as an output that costs more in fees to spend than the value of the output.

To compute the cost of spending an output, both its size and the size of the input are considered.

$$\text{is_dust}(\text{out}) = \begin{cases} 1, & \text{out}_v < f * (41 + 107/\alpha + \text{out}_s) \\ 0, & \text{otherwise} \end{cases}$$

where α is 1 for non-segwit outputs and 4 otherwise.

Unprofitable

Our definition: unprofitable outputs

Unprofitable

We define an **unprofitable** output as the output of a transaction that holds less value than the fee necessary to be spent, taking into account **only the size of the input** that will be needed to spend it.

$$\text{is_unprofitable}(\text{out}) = \begin{cases} 1, & \text{out}_v < f * \text{pred_in}_s / \alpha \\ 0, & \text{otherwise} \end{cases}$$

where:

pred_in_s is the predicted size of the input that will spend output out .

α is 1 for non-segwit outputs and 4 otherwise.

Unprofitable

Our definition: unprofitable outputs

...but how do we know the size of an input before we see it?

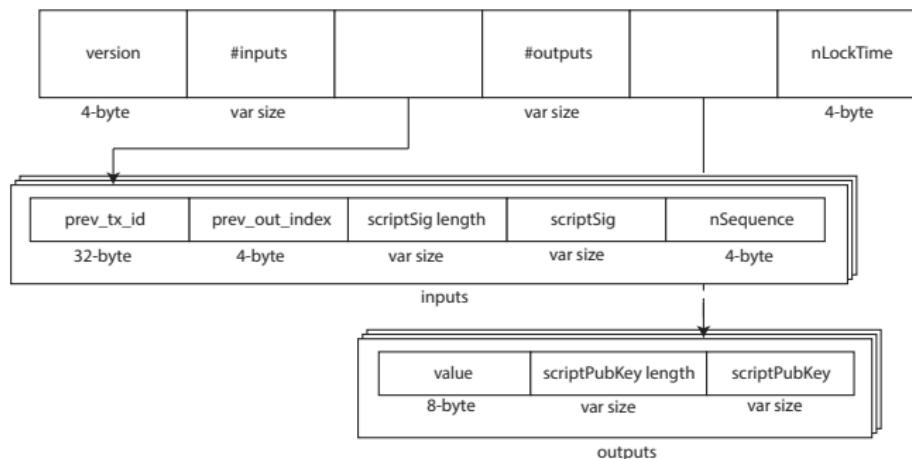


Figure: Generic transaction structure

Unprofitable

Our definition: computing the minimum size

$$\text{min_size} = \text{fixed_size} + \text{variable_size}$$

$$\text{fixed_size} = \text{outpoint} + n\text{Sequence} = \boxed{40 \text{ bytes}}$$

$$\text{variable_size} = \underline{\text{scriptSig_len} + \text{scriptSig}}$$



depends on the UTXO type

Unprofitable

Our definition: unprofitable outputs

Two different metrics for unprofitability:

- A **lower bound** on unprofitability, that will take into account the minimum size of the input;
- An **estimation** of unprofitability, that tries to estimate the real unprofitable rates taking into account data available in the blockchain.

Unprofitable

Variable size: non-SegWit

Pay-to-PubKey (P2PK) outputs:

PUSH sig (1 byte) + sig (71 bytes)

Pay-to-PubkeyHash (P2PKH) outputs:

PUSH sig (1) + sig (71) + PUSH pk (1) + pk (33-65)

Pay-to-multisig (P2MS) outputs:

OP_0 (1) + (PUSH sig (1) + sig (71)) * req_sigs (1-20)

Pay-to-ScriptHash (P2SH) outputs:

∅

Unprofitable

Variable size: SegWit

Pay-to-Witness-Public-Key-Hash (P2WPKH) outputs:

PUSH sig (1) + sig (72) + PUSH pk (1) + pk (33)

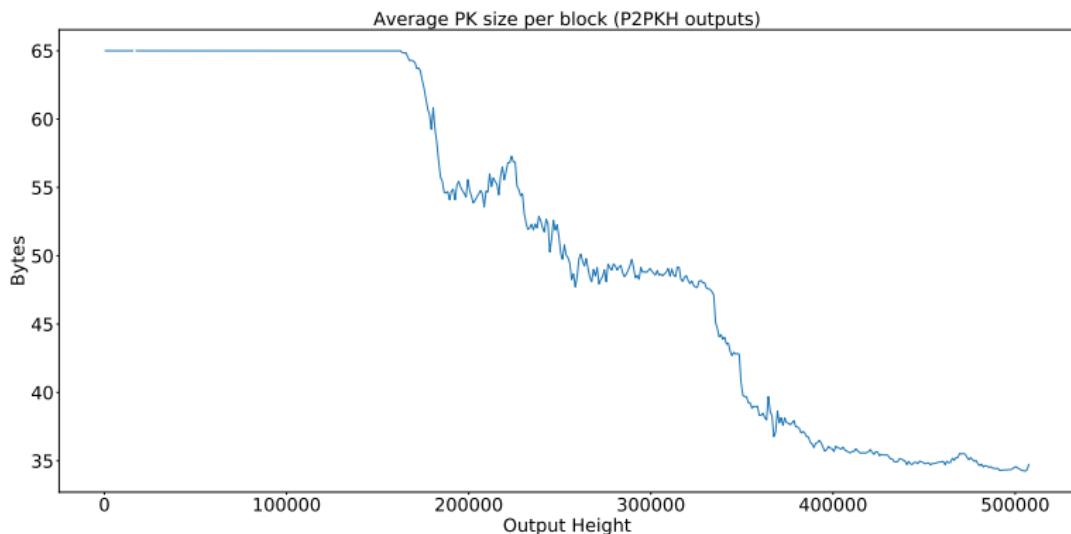
Pay-to-Witness-Script-Hash (P2WSH) outputs:

∅

Witness scripts discounted $\alpha = 1/4$

Unprofitable

Public key sizes in the Bitcoin blockchain



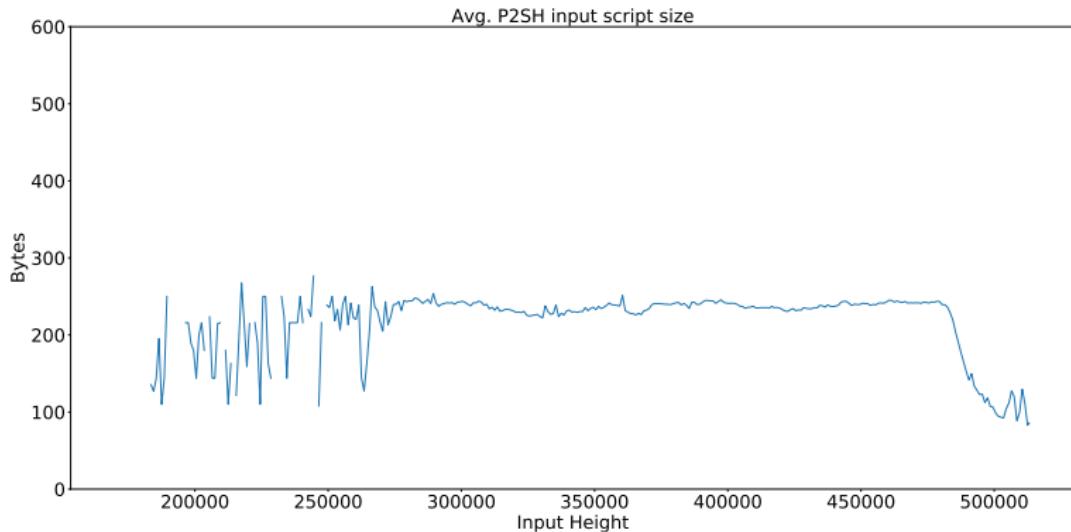
Unprofitable

P2SH redeem scripts in the Bitcoin blockchain

Redeem script	Number of inputs	Average input size
Multisig	80,839,329	241.6
P2WPKH	7,961,073	23
P2WSH	5,544,793	35
Nonstd	112,354	169.98
P2PK	23,557	108.01
P2PKH	448	132
P2SH (Hash puzzle)	82	28.73
Total	94,481,636	210.93

Unprofitable

P2SH sizes in the Bitcoin blockchain



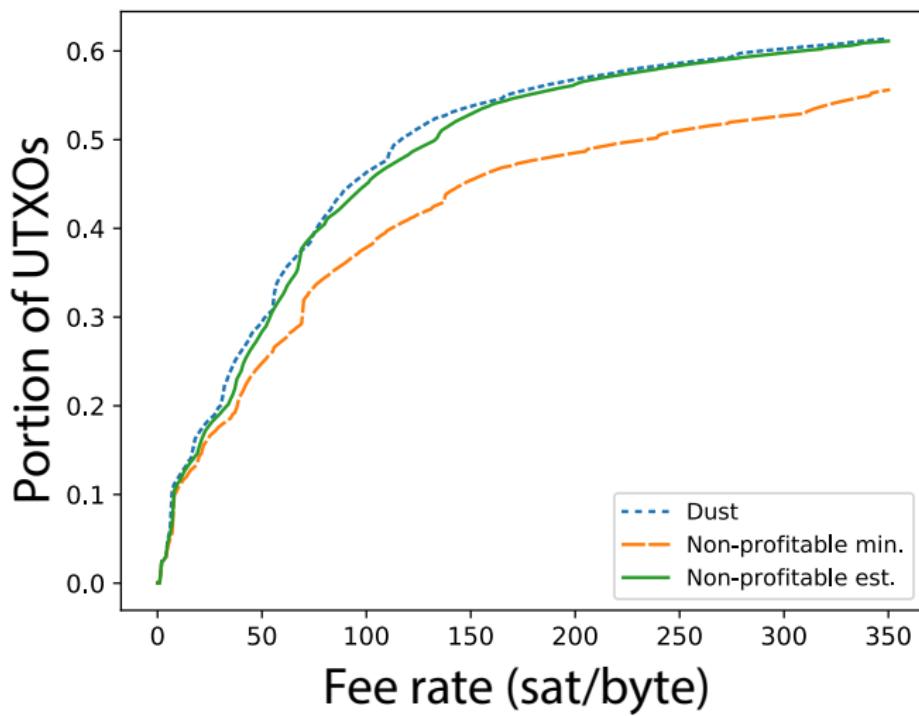
1 Introduction

2 Definitions

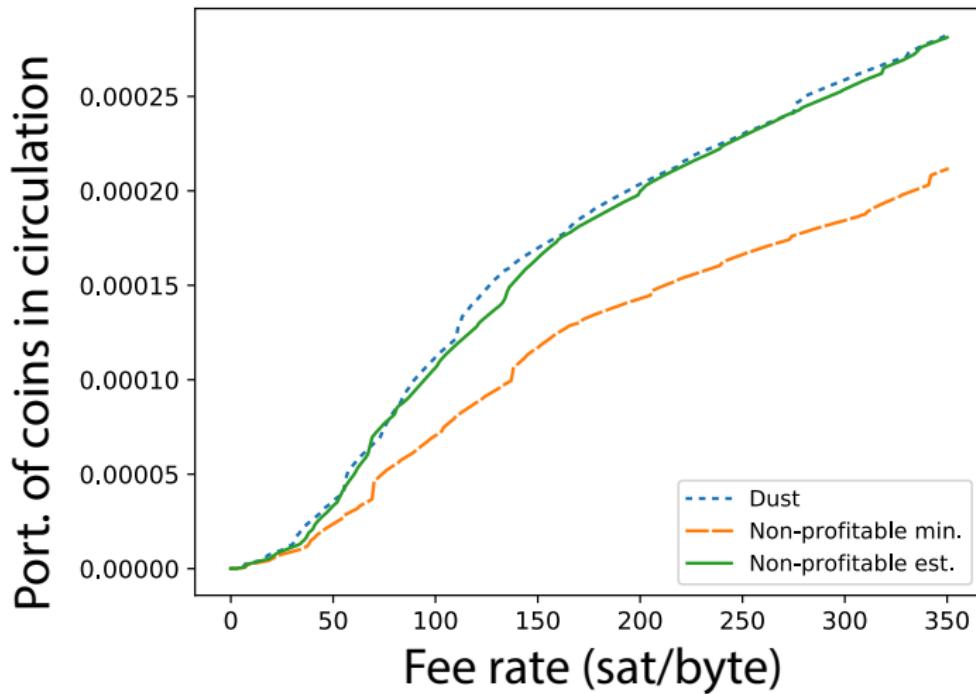
3 Results

4 Conclusions

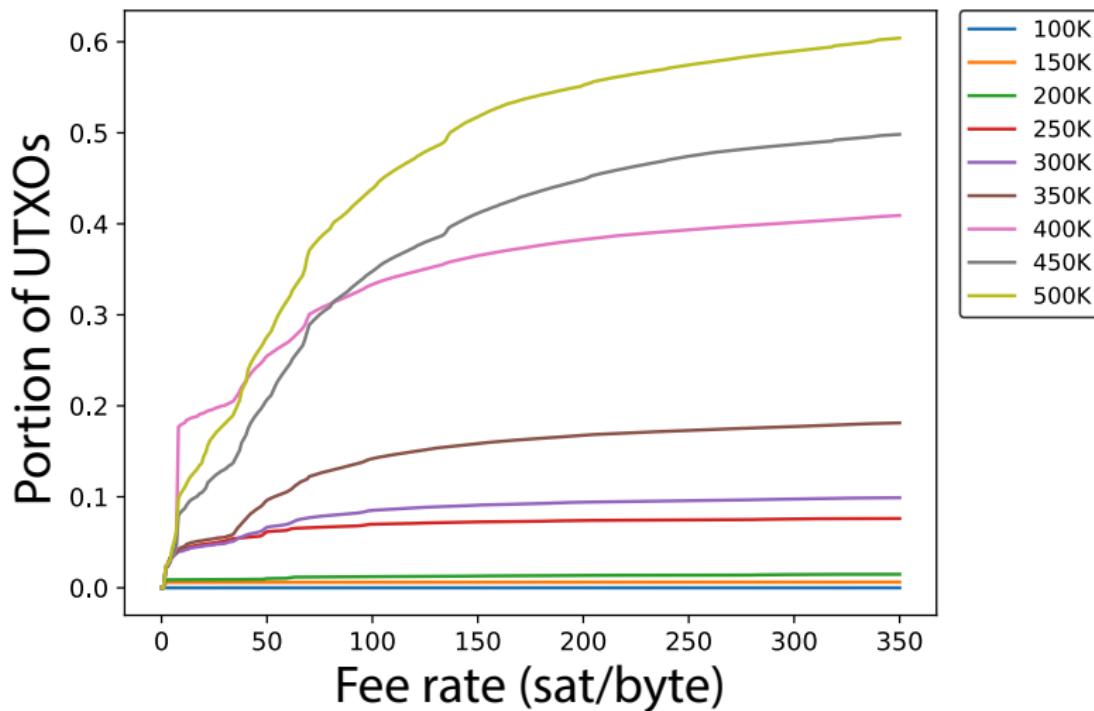
Dust UTXOs



Dust value



Unprofitability evolution (Bitcoin)



Introduction
ooo

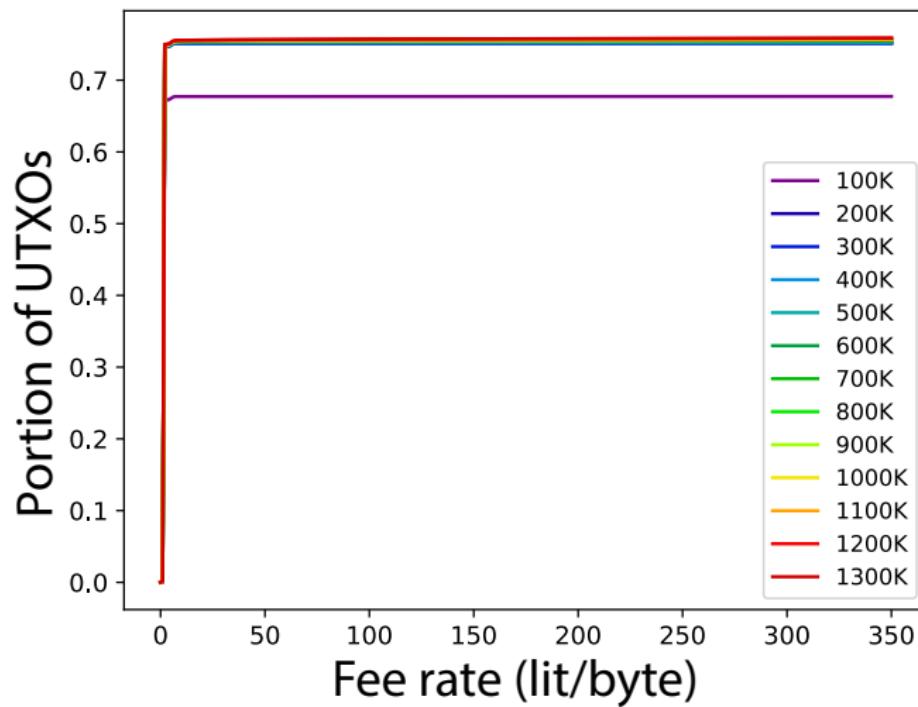
Definitions
oooooooooooo

Results
oooo●○

Conclusions
ooo

Is this really that bad?

Unprofitability evolution (Litecoin)



1 Introduction

2 Definitions

3 Results

4 Conclusions

Conclusions

- There is a fairly big percentage of dust in the UTXO set
- The current implementation of the UTXO set can grow unbounded
- The bigger the set gets, the less suitable it is to run a full node in low resource devices
- Dust attacks can be performed to make the set grow

Solutions?

- There has been proposals to mitigate this (TXO commitments by Peter Todd)
- Output consolidation when fees are low
- A good coin selection algorithm is important, specially for exchanges

An analysis of dust in UTXO based cryptocurrencies

Cristina Pérez-Solà, **Sergi Delgado-Segura**, Guillermo
Navarro-Arribas, Jordi Herrera-Joancomartí

Departament d'Enginyeria de la Informació i les Comunicacions
Universitat Autònoma de Barcelona

October 6th, 2018