ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KĨ THUẬT MÁY TÍNH

MÔ HÌNH HÓA TOÁN HỌC

Assignment

Mathematical model for UTXO selection

GVHD: TS. Huỳnh Tường Nguyên

Nhóm: L01, Nhóm 1
DSSV: Phạm Phước Hoài - 1711371
Nguyễn Công Anh - 1710477
Phạm Thành Công - 1710706
Nguyễn Quang Vương - 1714037
Nguyễn Lê Quốc Cường - 1710721
Nguyễn Trần Minh Đăng - 1711014

Thành phố Hồ Chí Minh, Ngày 7 tháng 5 năm 2019

Mục lục

Danh sách hình vẽ

1 Giới thiệu

1.1 Công nghệ Blockchain

1.2 Tiền điện tử

Tiền điện tử (hay tiền ảo) là một tài sản kỹ thuật số ứng dụng công nghệ blockchain, sử dụng các hệ thống mật mã để bảo đảm các giao dịch và tính toàn vẹn mà không cần sự can thiệp của bên thứ ba nào khác. Tất cả các giao dịch tiền điện tử trong hệ thống được đăng ký trên một cuốn sổ cái (ledger) cấu thành từ một chuỗi các khối, mỗi khối chứa một số lượng giao dịch không cố định cùng với mã băm của khối trước đó để tất cả các giao dịch trong chuỗi khối là bất biến và hợp lệ (công nghệ blockchain).
Một ví dụ điển hình của loại tiền điện tử này là Bitcoin, được giới thiệu vào năm 2008, Bitcoin hiện có hơn 141 tỷ đô la trên thị trường tiền điện tử, với trung bình 229,000 giao dịch mỗi ngày và khoảng 183.89 GB dung lượng lưu trữ.

1.3 Đề tài

Trong các mạng blockchain, số dư tài khoản tiền điện tử được quản lý theo hai mô hình khác nhau: "mô hình dựa trên tài khoản" (account-based models) và "mô hình dựa trên giao dịch" (transaction-based models - UTXO models).

Trong mô hình dựa trên giao dịch, người ta sử dụng các đầu ra giao dịch (transaction outputs) để chi trả cho các giao dịch mới với vai trò là đầu vào giao dịch (transaction inputs). Bất cứ đầu ra giao dịch nào chưa trở thành đầu vào giao dịch, tức là chưa được sử dụng để chi trả trong một giao dịch bất kỳ, được gọi là Unspent Transaction Output (UTXO). Mỗi số dư tài khoản tiền điện tử được thể hiện bằng một tập các UTXO (set of UTXOs). Khi một người dùng sử dụng tiền của mình để giao dịch với một người dùng khác, một giao dịch được tạo ra bằng cách chọn các UTXO trong tập UTXOs của người đó làm các đầu vào giao dịch đồng thời tạo ra các UTXO mới làm đầu ra giao dịch và gửi cho người nhận.

Chiến lược lựa chọn các UTXO cho giao dịch đóng vai trò quan trọng trong việc quản lý số dư tài khoản tiền điện tử. Một chiến lược lựa chọn tối ưu phải đáp ứng các ràng buộc cứng và các mục tiêu thiết yếu của ba nhóm chính là người dùng (users), cộng đồng (community) và thợ đào tiền ảo (miners). Đối với người dùng, họ muốn tạo ra một giao dịch mà giảm thiểu phí giao dịch và đồng thời muốn được đảm bảo quyền riêng tư trong các hành vi của mình. Ngược lại, các thợ đào tập trung vào việc khai thác các giao dịch có mức phí càng cao càng tốt. Đối với cộng đồng, kích thước "UTXO pool" lớn trở thành một vấn đề nghiêm trọng vì nó làm giảm hiệu suất xử lý giao dịch và cũng tạo ra chi phí tiêu thụ bộ nhớ cao.

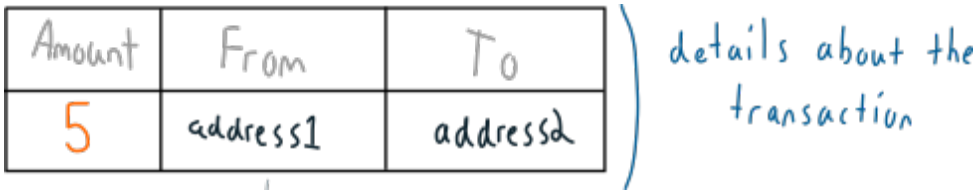
Trong bài tập lớn này, nhóm đã đề xuất một chiến lược hiệu quả để lựa chọn các UTXO cho một giao dịch nhất định mà sẽ tốn một khoản phí tối thiểu cho các thợ đào tiền ảo hoặc thu thập càng nhiều UTXO nhỏ càng tốt để giảm kích thước "UTXO pool".

2 Cơ sở lí thuyết

2.1 Tổng quan

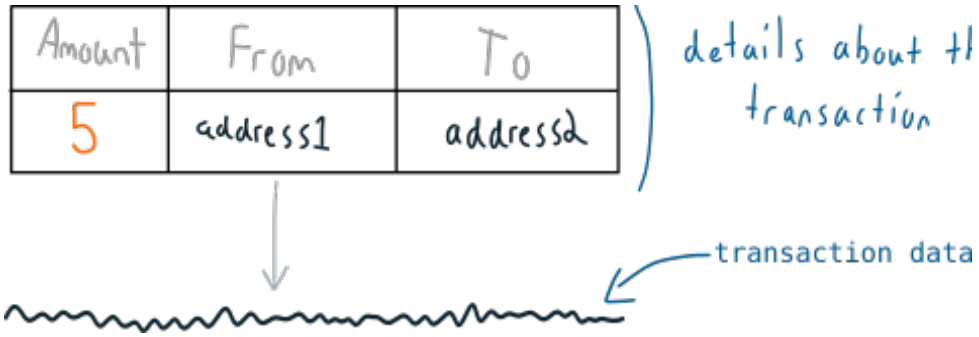
Giao dịch:
Một giao dịch Bitcoin là một loạt các dữ liệu, chứa thông tin về số tiền được gửi, tài khoản của người gửi và tài khoản mà nó đang được gửi đến. Chẳng hạn:

Hình 1: Góc nhìn trừu tượng của giao dịch



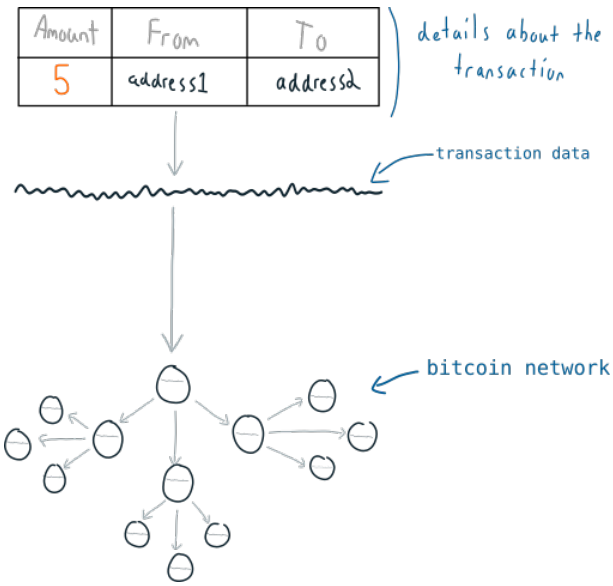
Đây là thông tin cơ bản, vì vậy nó có thể được trình bày dễ dàng trong một dòng dữ liệu:

Hình 2: Góc nhìn trừu tượng của dữ liệu giao dịch



Khi thực hiện một giao dịch, ta chỉ cần gửi dữ liệu giao dịch này vào mạng Bitcoin:

Hình 3: Đưa dữ liệu giao dịch vào mạng Bitcoin

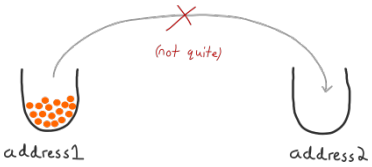


Cuối cùng, một trong các nút (node) trên mạng sẽ "đào" giao dịch này vào một khối, và khối này sẽ được nối vào blockchain.

Hình 4: Thêm giao dịch vào khối

Cách một giao dịch hoạt động:
Một địa chỉ bitcoin (bitcoin address) giống như một số tài khoản (account number) giữ bitcoins. Nhưng giao dịch không hoạt động bằng cách lấy một lượng bitcoin chính xác từ một cái túi chuyển sang cái túi khác.

Hình 5: Giao dịch bitcoins không giống với ngân hàng truyền thống

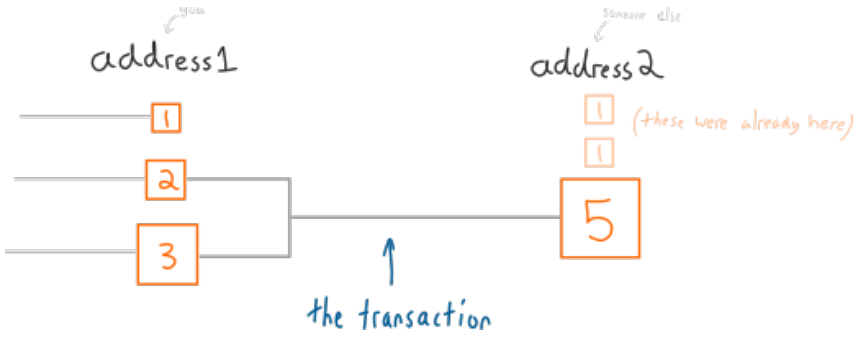


Thay vào đó, mỗi địa chỉ bitcoin theo dõi từng giao dịch riêng lẻ mà nó nhận được

Hình 6: Có 3 coins nhận được từ 3 giao dịch trước

Vì vậy, khi muốn gửi bitcoins cho một người khác, chúng ta phải lấy toàn bộ số bitcoins đã nhận trong một giao dịch để gửi.

Hình 7: 2 coins được chuyển từ địa chỉ 1 sang địa chỉ 2

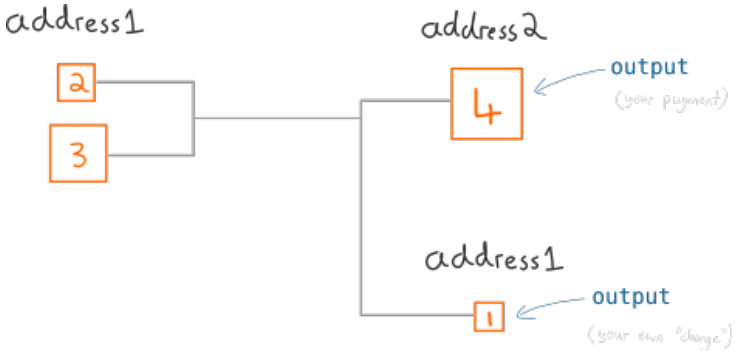


Và người nhận được bitcoins có thể sử dụng số bitcoins đó để giao dịch với một người khác nữa

Hình 8: 3 coins tiếp tục được sử dụng để giao dịch

Nếu tổng số bitcoins lớn hơn số tiền cần gửi thì người gửi chỉ cần gửi cho chính mình số tiền thừa, điều này có thể hơi rắc rối nhưng nó sẽ dễ dàng cho việc lập trình.

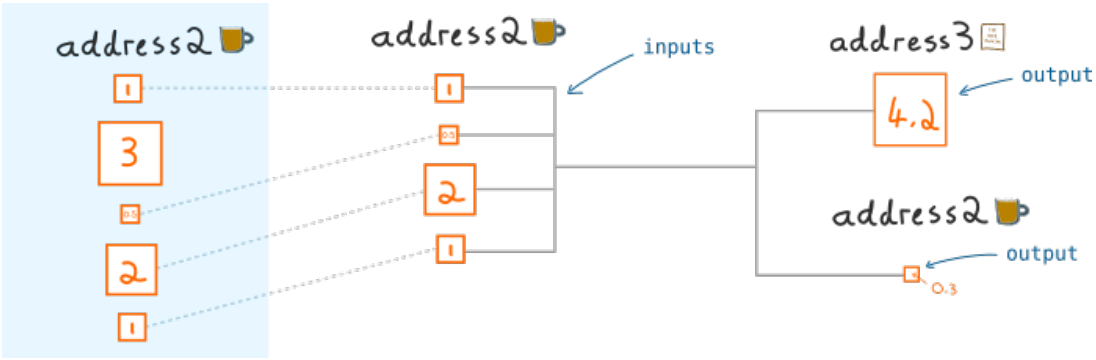
Hình 9: Người gửi tự gửi cho mình 1 coin tiền thừa



Tóm lại:
1. Người dùng có một địa chỉ bitcoin. Bitcoins đến địa chỉ này theo đợt, được gọi là các đầu ra (outputs).
2. Giao dịch bitcoin là quá trình sử dụng các đầu ra này để tạo các đầu ra mới (và gửi chúng đến địa chỉ mới).

Các đầu ra:
Trong giao dịch bitcoin, ta sử dụng đầu ra của các giao dịch cũ để làm các đầu vào cho giao dịch mới.
Trong một ví dụ thực tế, giả sử chúng ta mua bia ở một cái máy bán bia và phải trả tiền cho nơi sản xuất thông qua giao dịch bằng tiền ảo:

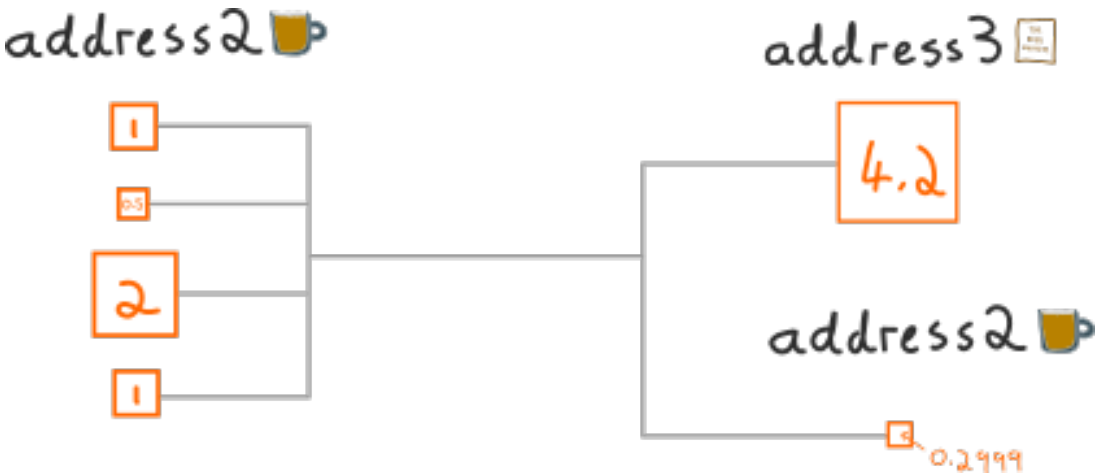
Hình 10: Thanh toán



Trong giao dịch, ta không có sẵn một đầu ra đủ để trả tiền bia nhưng chúng ta có thể sử dụng nhiều đầu ra có tổng giá trị lớn hơn số tiền cần trả (4.2), nếu chúng ta sử dụng các đầu ra như hình trên, đầu ra có giá trị 3 chưa được sử dụng và có thể sử dụng được sau này.
Những đầu ra chưa được sử dụng như vậy được gọi là các Unspent Transaction Output (UTXOs)

Phí giao dịch:
Và cuối cùng, để giao dịch được thực hiện thì các thợ đào phải đưa giao dịch vào trong một khối (block). Để các thợ đào thực hiện điều đó, đương nhiên việc trả một khoản phí cho họ là cần thiết. Điều này khiến cho tổng giá trị đầu ra của một giao dịch luôn nhỏ hơn tổng giá trị đầu vào:

Hình 11: Giá trị đầu ra nhỏ hơn giá trị đầu vào



2.2 Wallets

Ví tiền (wallet) là phần mềm chịu trách nhiệm chuyển tiền giữa các bên. Mỗi người dùng có thể sử dụng phần mềm ví của bất kỳ nhà cung cấp nào để quản lý khóa riêng tư (private key) và số dư tài khoản tiền điện tử của mình. Thông qua ví, người dùng giao dịch với những người dùng khác bằng cách tạo giao dịch.

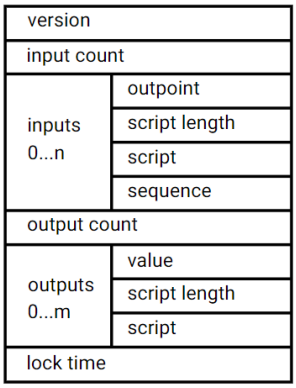
2.3 Unspent Transaction Output (UTXOs)

Đầu ra giao dịch chưa được sử dụng (UTXO) là cơ chế nội bộ được sử dụng trong nhiều loại tiền điện tử để đại diện cho các đồng xu (coins).

2.4 Transaction

Giao dịch là một thông điệp chuyển quyền sở hữu một hoặc nhiều đồng tiền kỹ thuật số (digital coins) giữa những người giao dịch. Mỗi coin được chi tiêu xuất hiện như một đầu vào (input) giao dịch, mỗi đầu vào tham chiếu đến coin duy nhất mà nó chi tiêu. Một coin mới được tạo bởi một giao dịch xuất hiện dưới dạng đầu ra (output).

Hình 12: Định dạng giao dịch



Hình 13: Các trường cơ sở của một giao dịch

Name	Length (bytes)	Description
Version	4	The transaction format version. Currently 2.
Input Count	1-9	The number of inputs.
Outpoint	36	The parent transaction's ID concatenated with the output's zero-based index.
Input Script Length	1-9	The length of the input script, in bytes.
Input Script	variable	The authentication response for the output to be spent.
Sequence	4	Used with time locks.
Output Count	1-9	The number of outputs. Larger counts require more bytes.
Value	8	The face value of the output, in satoshis.
Output Script Length	1-9	The length of the output script, in bytes.
Output Script	variable	The authentication challenge for the new coin.
Lock Time	4	A timestamp or block index past which the transaction becomes valid.

Cách thức ước lượng kích thước giao dịch:
Loại giao dịch phổ biến nhất sử dụng mẫu ủy quyền được gọi là pay-to-public-key-hash (P2PKH).

Hình 14: P2PKH: Tập lệnh Input và Tập lệnh Output

A P2PKH input script contains two values that require on average 107 bytes. Variability results from the method used to encode signatures.

P2PKH Input Script. Values require an additional variable length integer as a length specifier. Signature lengths vary due to DER encoding. The total length for this input script is 107±1 bytes.

Name	Length (bytes)	Description
`<signature>`	72±1 + 1	Transaction signature.
`<pubkey>`	33 + 1	An uncompressed public key would require 65 + 1 bytes.

A P2PKH output script encodes four opcodes (one byte each) and a 20-byte value (requiring 21 bytes total). The result requires 25 bytes.

P2PKH Output Script. Opcodes require one byte and values require. The total length is 25 bytes.

Name	Length (bytes)	Description
`OP_DUP`	1	Duplicate the top stack item.
`OP_HASH160`	1	Pop value, push its hash value.
`<pubkeyhash>`	20 + 1	Push the hash value of a public key.
`OP_EQUALVERIFY`	1	Pop two items, verify their equality.
`OP_CHECKSIG`	1	Check transaction signature using the contents of the stack.

Kết hợp các trường yêu cầu độ dài này mang lại một công thức có thể tính toán kích thước của bất kỳ giao dịch P2PKH nào:
$s = 10 + 148 * n + 34 * t \pm n$, với n là số inputs, t là số outputs.
148 = outpoint(36 bytes) + script length (1 byte) + script (107 bytes) + sequence (4 bytes).
34 = value field(8 bytes) + script length (1 byte) + and script (25 bytes).

3 Thành lập bài toán

3.1 Yêu cầu

Yêu cầu của bài toán: *Đưa ra một chiến lược lựa chọn hiệu quả các UTXO trong tập UTXOs sao cho thỏa mãn nhu cầu thiết yếu của 3 nhóm chính:*
1. Người dùng: người dùng muốn giao dịch của mình phải trả một khoản phí thấp nhất có thể.
2. Cộng đồng: Cộng đồng muốn tăng tốc độ truy vấn thông tin trong mạng blockchain, đồng thời giảm lượng dữ liệu cần phải lưu trữ.
3. Thợ đào tiền ảo: Thợ đào muốn đào được những giao dịch có mức phí cao để thu về nhiều tiền ảo cho mình.

Để giải quyết yêu cầu của người dùng, chiến lược lựa chọn phải tối thiểu hóa phí giao dịch. Để giải quyết yêu cầu của cộng đồng, chiến lược lược lựa chọn phải tối đa hóa số lượng UTXOs được chọn làm đầu vào giao dịch và tối thiểu hóa số lượng UTXO mới được sinh ra bởi giao dịch. Để giải quyết yêu cầu của thợ đào, cần tăng phí giao dịch.

Rõ ràng, chúng ta không thể làm thỏa mãn tối đa yêu cầu của cả 3 nhóm người này cùng một lúc, vì họ luôn quan tâm đến lợi ích của bản thân đầu tiên. Do đó, chiến lược lựa chọn phải đảm bảo hướng đến những mục tiêu trên ở mức phù hợp, sao cho người dùng phải trả một khoản phí chấp nhận được đồng thời đảm bảo giao dịch của họ có lượng phí thu hút các thợ đào nhằm tăng tốc độ xác thực giao dịch. Cuối cùng, phải duy trì lượng UTXOs trong "UTXO pool" ở mức mà các hệ thống hiện nay có thể đáp ứng được về tốc độ truy vấn và khả năng lưu trữ.

Trong bài tập lớn này, nhóm đã đề xuất một mô hình giúp giảm thiểu tối đa phí giao dịch và một mô hình khác nhằm duy trì lượng phí giao dịch ở mức thấp nhưng đồng thời tăng thêm lượng UTXO được chọn vào làm đầu vào cho các giao dịch.

3.2 Ý tưởng

Mô hình 1: Tối thiểu phí giao dịch
Công thức tính phí giao dịch:

$$fee = fee\_rate * transaction\_size$$

Trong đó, "fee_rate" là một đại lượng cố định,... Vì vậy chiến lược lựa chọn UTXOs hướng đến mục tiêu là tối thiểu kích thước giao dịch (transaction size).
Công thức tính kích thước giao dịch (P2PKH):

$$transaction\_size = all\_Inputs\_Size + all\_Outputs\_Size + 10$$

Trong đó, "all_Inputs_Size" là tổng kích thước các đầu vào của giao dịch, "all_Outputs_Size" là tổng kích thước các đầu ra của giao dịch. Vậy, để tối thiểu được kích thước của giao dịch, chúng ta cần phải tối thiểu kích thước tập đầu vào hoặc tối thiểu kích thước tập

đầu ra của giao dịch, hoặc tốt nhất là cả hai.

a) Đối với việc tối thiểu kích thước tập đầu vào của giao dịch:
Khi một giao dịch được người dùng tạo ra, một số UTXO từ tập UTXOs (set of UTXOs) của người dùng đó sẽ được chọn làm các đầu vào để chi trả cho giao dịch. Giá trị cần phải chi trả cho giao dịch là tổng giá trị các đầu ra và phí giao dịch:

$$target = all\_Outputs\_Value + Transaction\_Fee$$

Dĩ nhiên, các UTXO được chọn phải có tổng giá trị lớn hơn hoặc bằng "target" để giao dịch là hợp lệ, đây là ràng buộc cứng cho mọi lựa chọn:

$$all\_Inputs\_Value \geq target$$

Tuy nhiên, có thể có rất nhiều cách lựa chọn các UTXO làm đầu vào cho giao dịch thỏa mãn ràng buộc cứng. Công việc mà chúng ta phải làm là tìm ra trong đó cách lựa chọn nào làm cho tổng kích thước tập đầu vào là nhỏ nhất.
b) Đối với việc tối thiểu kích thước tập đầu ra của giao dịch:
Ta có thể chia tập đầu ra thành 2 phần:
Phần 1: những đầu ra gửi cho những người khác
Phần 2: một đầu ra gửi phần dư lại cho chính người gửi.
Dễ thấy, tổng giá trị và tổng kích thước của phần 1 là cố định, nếu muốn tối thiểu kích thước của tập đầu ra, ta chỉ có thể tối thiểu kích thước của phần thứ 2. Công việc mà chúng ta phải làm là tìm ra cách lựa chọn các UTXO sao cho tổng giá trị của chúng đúng bằng "target" để không sinh ra phần dư.

Thực tế cho thấy, để kích thước giao dịch là nhỏ nhất, ta phải tìm ra tổ hợp các UTXO sao cho kích thước tập đầu vào là nhỏ nhất trước, sau đó mới xét xem tổ hợp đó có làm tối thiểu được kích thước tập đầu ra hay không, việc tăng kích thước tập đầu vào (chọn nhiều UTXO hơn) để tìm ra một tổ hợp có tổng giá trị đúng bằng "target" luôn làm kích thước giao dịch tăng lên so với khi đã tối thiểu được kích thước tập đầu vào. Vì vậy, khi tìm cách tối thiểu kích thước một giao dịch, ta chỉ xét các trường hợp sau đây:
TH1: Tập đầu vào có kích thước nhỏ nhất, giao dịch không dư
Đây là trường hợp tốt nhất có thể xảy ra:

$$all\_Inputs\_Value = target$$

• transaction_size đạt min khi thỏa các điều kiện sau:

$$all\_Inputs\_Value = target$$
$$all\_Inputs\_Size \text{ đạt } min$$
$$all\_Outputs\_Size \text{ đạt } min$$

TH2: Tập đầu vào có kích thước nhỏ nhất, giao dịch có dư

$$all\_Inputs\_Value > target$$
$$change\_output = all\_Inputs\_Value - target$$

Ta xét phần dư "change_output" so với ngưỡng bụi ("dust_threshold"):
Nếu $change\_output > dust\_threshold$, ta tạo thêm 1 UTXO để trả phần dư lại cho người dùng.
Nếu $0 < change\_output \leq dust\_threshold$, ta không thể tạo thêm 1 UTXO để trả lại phần dư, phần dư này sẽ được cộng thêm vào số tiền mà thợ đào tìm thấy ở giao dịch, hoặc sẽ trở thành rác hệ thống dưới hình thức là một đầu ra không thể giao dịch được (unspendable ouput), dù sao đi nữa, chúng ta cũng sẽ không chấp nhận trường hợp này.
• transaction_size đạt min khi thỏa các điều kiện sau:

$$all\_Inputs\_Value > target$$
$$all\_Inputs\_Size \text{ đạt min}$$
$$change\_output > dust\_threshold$$

Vậy, giải thuật tối ưu sẽ đi tìm một tổ hợp các UTXO thỏa mãn "target" và "all_Inputs_Size" đạt min, sau đó xét xem giao dịch có dư hay không, nếu giao dịch không dư thì tổ hợp được chấp nhận, nếu giao dịch có dư thì phần dư này phải lớn hơn ngưỡng bụi mới chấp nhận tổ hợp.

Mô hình 2: Tối đa số UTXO được chọn ở mức phí cho phép
Từ kết quả của mô hình 1, ta đưa thêm vào một đại lượng "gamma", sau đó tính ra phần kích thước giao dịch cho phép tăng lên so với mô hình 1, từ đó chọn lại một tổ hợp UTXO khác sao cho số UTXO được chọn là lớn nhất.

4 Mô hình đề xuất

Chiến lược lựa chọn các UTXO (Proposed UTXO Selection Strategy)

Mục tiêu: Xác định một tập con UTXOs phù hợp sao cho thỏa mãn nhiều ràng buộc bao gồm các ràng buộc cứng H1 và các ràng buộc mềm S1.

Input: Tất cả những tham số đầu vào cho chiến lược

Tham số đầu vào	Mô tả
$U = \{u_1, ..., u_n\}$	tập UTXOs
$O = \{o_1, ..., o_m\}$	tập đầu ra giao dịch
$V^u = \{V_1^u, ..., V_n^u\}$	Tập giá trị các UTXO
$V^o = \{V_1^o, ..., V_m^o\}$	Tập giá trị các đầu ra giao dịch
$S^u = \{S_1^u, ..., S_n^u\}$	Tập kích thước các đầu vào giao dịch, với đầu vào được chọn từ UTXO u_i
$S^o = \{S_1^o, ..., S_m^o\}$	Tập kích thước các đầu ra giao dịch
M	Kích thước tối đa của giao dịch (kích thước block)
α	Tỉ lệ phí
T	Ngưỡng bụi
ε	Giá trị phần dư tối thiểu được cho phép
γ	Tỉ lệ tăng thêm của kích thước giao dịch

Output: Kết quả của chiến lược
- Tập con UTXOs được chọn thỏa mãn các ràng buộc
- Phần dư (nếu có)

Các ràng buộc cứng H1:
1) Tổng giá trị các UTXO được chọn phải đủ để chi trả cho giao dịch.
2) Kích thước giao dịch không vượt quá kích thước block.
3) Tất cả các đầu ra giao dịch phải lớn hơn ngưỡng bụi để chắc chắn rằng giao dịch này được chuyển tiếp lên mạng và được xác nhận.

Các ràng buộc mềm S1:
1) Kích thước giao dịch được tối thiểu hóa.
2) Số lượng UTXO được chọn được tối đa hóa nhằm thu hẹp kích thước "UTXO pool".

4.1 Mô hình 1: Tối thiểu kích thước giao dịch

Các biến:
1.a) Biến quyết định:

$$x_i = \begin{cases} 1, & \text{nếu UTXO } u_i \text{ được chọn} \\ 0, & \text{nếu không} \end{cases}$$

1.b) Biến trung gian:
- y: kích thước giao dịch
- z_v: giá trị của phần dư sao giao dịch
- z_s: kích thước của phần dư sau giao dịch

$$z_s = \begin{cases} 0, & 0 \leq z_v \leq \varepsilon \\ \beta, & z_v > \varepsilon \end{cases}$$

Các ràng buộc:

2.a) Kích thước giao dịch không được lớn hơn kích thước block

$$y = \sum_{i|u_i \in U} (s_i^u * x_i) + \sum_{j|o_j \in O} (s_j^o) + 10 + z_s \leq M$$

2.b) Tổng giá trị các UTXO được chọn phải chi trả được cho giao dịch

$$\sum_{i|u_i \in U} (v_i^u * x_i) \geq \sum_{j|o_j \in O} (v_j^o) + \alpha * y + z_v$$

2.c) Tất cả các giá trị đầu ra giao dịch phải cao hơn ngưỡng bụi để chắc chắn rằng giao dịch này được chuyển tiếp lên mạng và được xác nhận.

$$T \leq v_j^o, \forall j | o_j \in O$$

2.d) Mối quan giữa giá trị phần dư z_v và kích thước phần dư z_s như sau:

$$z_s \leq \left\lfloor \frac{z_s}{\varepsilon} \right\rfloor * \beta$$

Nếu $z_v \leq \varepsilon$, z_s sẽ bằng 0; nếu không z_s sẽ bằng β

2.e) x_i là giá trị nhị phân

$$\forall i | u_i \in U : x_i \in \{0, 1\}$$

Hàm mục tiêu: Tối thiểu kích thước giao dịch

$$minimize\ y$$

4.2 Mô hình 2: Tối đa số lượng UTXO được chọn

Các biến:

Bao gồm tất cả các biến của Mô hình 1

Các ràng buộc:

Bao gồm tất cả các ràng buộc của Mô hình 1 và thêm một ràng buộc sau:

$$y < (1 + \gamma) * Y$$

trong đó,
- Y: là kích thước tối thiểu của giao dịch thu được bởi Mô hình 1
- γ: là tỉ lệ kích thước tăng thêm của giao dịch $(0 < \gamma < 1)$

Hàm mục tiêu: Tối đa số lượng UTXO được chọn

$$maximize \left(\sum_{i|u_i \in U} (x_i) - \frac{z_s}{\beta} \right)$$

5 Giải thuật tối ưu

Giải thuật tối ưu được viết bằng **Python** thuần và source code được gửi kèm với báo cáo. Các biểu đồ dữ liệu được vẽ bằng package **Matplotlib** của Python.
Mỗi file dữ liệu ban đầu được tách thành 3 file chứa trong cùng 1 thư mục, bao gồm:
"input_set.csv": Chứa tập UTXOs ban đầu
"output_set.csv": Chứa tập đầu ra của giao dịch
"const_set.csv": Chứa các tham số cần thiết

Trong giải thuật có định nghĩa các hàm sau đây:
Hàm 1: Kiểm tra tất cả các output có thỏa điều kiện lớn hơn ngưỡng bụi hay không

```
# Function 1:
def valid_output(outs_value, f_dust):
```

Hàm 2: Kiểm tra kích thước một giao dịch có có thỏa điều kiện nhỏ hơn kích thước block hay không

```
# Function 2:
def check_block_size(data, outs_size, f_beta, f_block_size):
```

Hàm 3: Tính tổ hợp chập k của n, nhằm chặn time_out khi số lượng tổ hợp quá lớn

```
# Function 3:
def nck(n, k):
```

Hàm 4: Giải thuật "Highest Value First" với k UXTOs có giá trị lớn nhất được chọn

```
# Function 4:
def highest_value_first(sort_values, k, outs_value, outs_size, optimal_hvf,
    f_beta, f_dust, f_fee_rate, f_block_size):
```

Hàm 5: Tìm các tổ hợp k UTXOs thỏa mãn mọi ràng buộc cứng

```
# Function 5:
def combinations(sort_sizes, outs_value, outs_size, data, start, end, index,
    r, optimal_a, optimal_b, f_beta, f_dust, f_fee_rate, f_block_size):
```

Hàm 6: Tìm ra k (số UTXO) nhỏ nhất để tìm tổ hợp k UTXOs trong n UTXOs ban đầu, được một tập các tổ hợp tối ưu nhất

```
# Optimal Function:
def constraints(input_num, sort_sizes, sort_values, outs_value, outs_size,
    f_block_size, f_dust, f_fee_rate, f_beta, optimal_a, optimal_b,
    optimal_hvf):
```

Hàm 7: Từ tập các tổ hợp tối ưu nhất, chọn ra 1 tổ hợp tối ưu nhất trong tập đó

```
def best_solution(in_optimal, num):
```

Hàm 8: Truyền gamma vào để giải mô hình 2

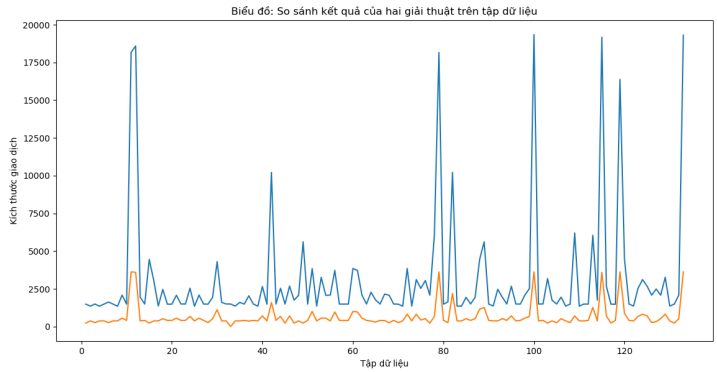
```
# Maximize UTXOs Function:
def maximize_choice(opcode, sort_values, sort_sizes, outs_value, outs_size,
    f_gamma, f_size, combination_num, input_num, optimal_c, optimal_d,
    optimal_hvf_new, f_beta, f_dust, f_fee_rate, f_block_size):
```

Giải thuật cụ thể được trình trong source code, kết quả chạy giải thuật được ghi ra 2 file: **"model1_result.csv"** và **"model2_result.csv"** được gửi kèm với báo cáo.

6 Thử nghiệm và đánh giá kết quả

Kết quả chạy mô hình 1 cho thấy kích thước các giao dịch đã giảm so với trong tập dữ liệu mẫu:

Hình 15: Kết quả chạy mô hình 1: Kích thước từng giao dịch

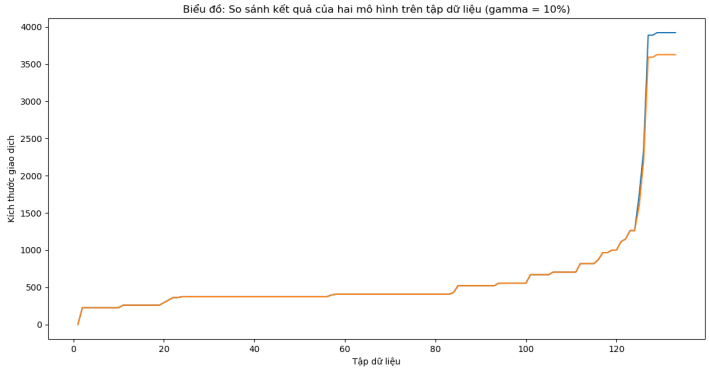


Hình 16: Kết quả chạy mô hình 1: Kích thước toàn bộ giao dịch

Tổng kích thước cũ (bytes)	Tổng kích thước mới (bytes)
416804	87620

Kết quả chạy mô hình 2 với $\gamma = 10\%$ cho phép tăng số lượng UTXO được chọn lên ở một mức phí phù hợp:

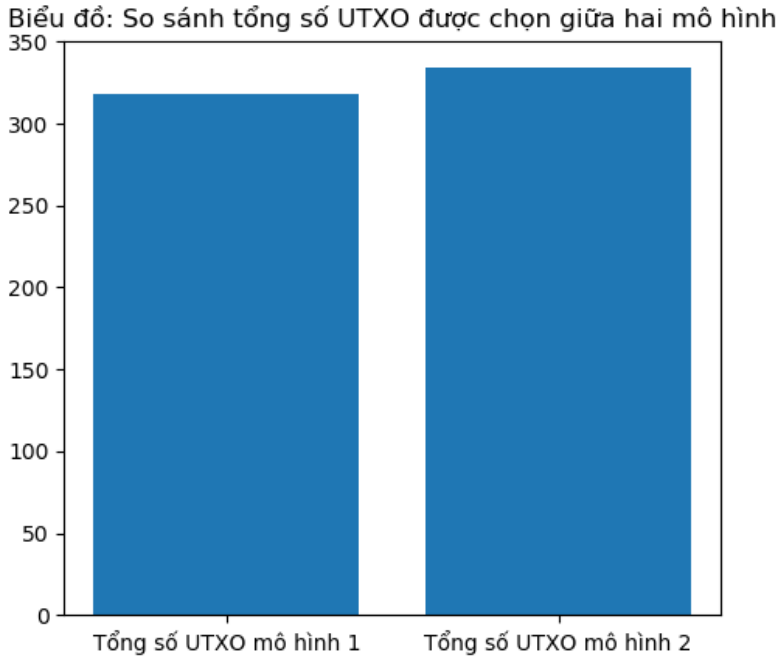
Hình 17: Kết quả chạy mô hình 2: Kích thước từng giao dịch so với mô hình 1



Hình 18: Kết quả chạy mô hình 2: Kích thước toàn bộ giao dịch so với mô hình 1

Tổng kích thước giao dịch (MH1, bytes)	Tổng kích thước giao dịch (MH2, bytes)
87620	89988

Hình 19: Kết quả chạy mô hình 2: Tổng số UTXO được chọn so với mô hình 1



Tổng số UTXO (MH1)	Tổng số UTXO (MH2)
318	334

7 Kết luận

Như vậy, bằng việc đề xuất 2 mô hình, một mô hình dùng để tối thiểu phí giao dịch và một mô hình dùng để tối ưu số tượng UTXO được chọn trong một mức phí tăng thêm cho phép. Nhóm đã viết giải thuật tối ưu để giải hai mô hình này và thu được những kết quả nhất định. Nếu so với kết quả trong tập dữ liệu test thì kết quả thu được tốt hơn nhiều, tuy nhiên vì chưa giải trực tiếp trên các tập dữ liệu khổng lồ ngoài đời thật nên giải thuật sẽ còn nhiều hạn chế chưa lường trước được. Vì vậy để giải thuật có thể sử dụng được ngoài thực tế cần thực hiện thêm nhiều lần thí nghiệm, kiểm thử với các tập dữ liệu trong thực tế cũng như sửa đổi và phát triển thêm cho giải thuật.

8 Tài liệu tham khảo

[1]: Transactions, lần truy cập cuối: 6/5/2019
[2]: Outputs, lần truy cập cuối: 6/5/2019
[3]: Making Sense of Bitcoin Transaction Fees, lần truy cập cuối: 6/5/2019
[4]: An analysis of dust in UTXO based cryptocurrencies, lần truy cập cuối: 6/5/2019
[5]: Bitcoin Fee Calculator & Estimator, lần truy cập cuối: 6/5/2019
[6]: UTXO-based cryptocurrencies, lần truy cập cuối: 6/5/2019
[7]: Bitcoin Transaction Fees, lần truy cập cuối: 6/5/2019
[8]: How do I calculate my transaction fee?, lần truy cập cuối: 6/5/2019
[9]: PREDICTING BITCOIN FEES FOR TRANSACTIONS., lần truy cập cuối: 6/5/2019

9 Phụ lục

Quá trình tìm hiểu các lý thuyết cũng như quá trình mô hình hóa hai bài toán tối ưu không thể tránh khỏi những sai sót, vì vậy, nhóm mong muốn sẽ nhận được những góp ý về bài tập này để có thể hoàn thiện hơn những gì đã làm.