



MATHEMATICAL MODELING

Assignment

Mathematical model for UTXO selection

Tutor: Huỳnh Tường Nguyên	(htnguyen@hcmut.edu.vn)
Class: L03-B,	Group: 12
Student members:	Nguyễn Linh Đăng Minh - 1712177
	Cao Thành Nhân - 1710214
	Lương Thiện Chí - 1610304
	Trần Minh Tú - 1713850
	Trịnh Đức Thọ - 1713343
	Nguyễn Đình Thịnh - 1713325
	Phạm Văn Việt - 1713955



Contents

1	Giới thiệu	2
2	Problem formulation	2
3	Proposed model	2
4	Experimental evaluation	3
5	Conclusion	3

1 Giới thiệu

Tiền mã hóa phi tập trung (decentralized cryptocurrency) là một tài sản kỹ thuật số sử dụng hệ thống mã hóa chọn lọc để đảm bảo an toàn cho giao dịch và tính minh bạch mà không cần sự tham gia can thiệp của bên thứ ba. Tất cả những giao dịch trong hệ thống đều được ghi lại trên một cuốn sổ cái (ledger) được gọi là blockchain - được cấu thành từ hàng loạt các khối khác nhau (block). Mỗi block chứa một con số không cố định của các giao dịch và đồng thời có một mã hash của block trước đó do đó tất cả các giao dịch trong blockchain rất khó để có thể thay đổi và trở nên hợp lệ vững chắc. Một ví dụ điển hình cho kiểu tiền mã hóa này chính là Bitcoin được giới thiệu vào năm 2008 mà hiện nay đang có hơn 141 tỉ USD trong thị trường tiền ảo, với 229 ngàn giao dịch và 183.89 GB lưu trữ trung bình trong một ngày.

Trong mỗi giao dịch dùng blockchains, thì chiến lược chọn UTXOs (Unspent Transaction Output set-Tập hợp các đầu ra giao dịch chưa chi tiêu) cho một giao dịch đóng một vai trò rất là quan trọng trong quản lý số dư tiền mã hóa của bất kỳ ví nào. Một chiến lược lựa chọn số tiền đã được tối ưu hóa thì phải thỏa mãn những ràng buộc cứng và những mục tiêu thiết yếu của ba nhóm chủ yếu như là users (người dùng), community (cộng đồng), và miners (người đào Bitcoin). Đứng từ phía người sử dụng, họ luôn muốn có thể tạo ra những giao dịch sẽ làm giảm thiểu phí giao dịch và đồng thời giữ được sự riêng tư của những giao dịch đó. Ngược lại, những miners lại tập trung vào khai thác những giao dịch có phí cao hơn càng nhiều càng tốt. Còn đối với cộng đồng, những UTXO cỡ lớn trở thành một vấn đề nan giải vì nó sẽ làm giảm thiểu hiệu suất xử lý các giao dịch và cũng sẽ gây ra những tiêu tốn chi phí lớn cho việc tiêu thụ bộ nhớ.

Trong phần nghiên cứu này, chúng tôi cân nhắc suy nghĩ về vấn đề chiến lược nghiên cứu để có thể chọn từ một tập các UTXOs đã được cho sẵn trong các giao dịch sao cho giảm được thấp nhất chi phí cho các miners hoặc tập hợp được càng nhiều các UTXO nhỏ để có thể giảm thiểu được kích thước của các UTXO.

Phần còn lại của báo cáo này được xây dựng theo các mục như sau: Mục 2. Xây dựng công thức cho vấn đề trình bày ngắn gọn nội dung và những yêu cầu của vấn đề đang được cân nhắc. Sau đó, chúng tôi trình bày những kết quả đạt được trong Mục 3. Đề xuất mô hình toán và Mục 4. Những suy luận đã được hình thành từ việc thử nghiệm mô hình, và bàn luận về kết quả của việc thử nghiệm này. Cuối cùng, chúng tôi tổng hợp và đưa ra kết luận những kết quả của nhóm ở mục cuối cùng.

2 Problem formulation

Describe clearly problem statement or requirement of the problem that needs to be modeled.

3 Proposed model

Sử dụng mô hình 2 để đưa ra chiến lược chọn UTXO dựa trên mô hình 1 như dưới đây:

a) Mô hình 1:

- Các biến:

+ Biến quyết định:

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

+ Biến trung gian:

- y : kích thước giao dịch
- z_v : giá trị của đầu ra thay đổi
- z_s : kích thước của đầu ra thay đổi

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

- Các ràng buộc:

- Kích thước giao dịch không được vượt quá kích thước dữ liệu khối tối đa.

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

- Một giao dịch phải có đủ giá trị để tiêu thụ.

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

- Tất cả các đầu ra giao dịch phải cao hơn ngưỡng bụi để chắc chắn rằng giao dịch này được chuyển tiếp đến mạng và được xác nhận.

$$T \leq \sum_{j|o_j \in O} v_j^o$$

- Mối quan hệ giữa giá trị đầu ra thay đổi z_v và kích thước của nó z_s được định nghĩa như sau:

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

- x_i là biến nhị phân.

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

- **Hàm mục tiêu:** Giảm thiểu kích thước giao dịch

minimize y

b) **Mô hình 2:** được xây dựng dựa trên kết quả thu được của mô hình 1

- **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 bổ sung thêm ràng buộc sau:

$$y < (1 + \gamma) \times Y, \text{ trong đó}$$

- Y là kích thước giao dịch tối thiểu thu được từ Mô hình 1.
- γ là hằng số ($0 < \gamma < 1$).

Nếu γ gần về 0, ta giữ kích thước giao dịch tối thiểu thu được từ Mô hình 1. Mặt khác, giao dịch có kích thước phù hợp được tạo bởi một số UTXO càng lớn càng tốt.

- **Hàm mục tiêu:** Tối đa hóa số lượng UTXO.

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

4 Experimental evaluation

Input format:

Tập dữ liệu bao gồm 133 file data định dạng .dat tương ứng với 133 trường hợp địa chỉ chứa các UTXOs đang khảo sát. Quan sát tập dữ liệu sẽ giúp ta hiểu rõ hơn về việc các UTXOs được chọn trong mạng lưới như thế nào. Tập dữ liệu bao gồm 133 trường hợp, trong đó không có địa chỉ nào có 1 hoặc không có UTXOs; có 99 địa chỉ chứa từ 2 đến 10 UTXOs trong tổng số 133 địa chỉ, chiếm tỉ lệ cao nhất (75%); có 19 trường hợp có từ 11 đến 100 UTXOs (chiếm khoảng 14.39% tổng thể) và 14 trường hợp có từ 101 đến 100000 UTXOs (khoảng 10.61% tổng thể). Figure 1 thể hiện rõ phân bố tần suất của các UTXOs trong các địa chỉ.

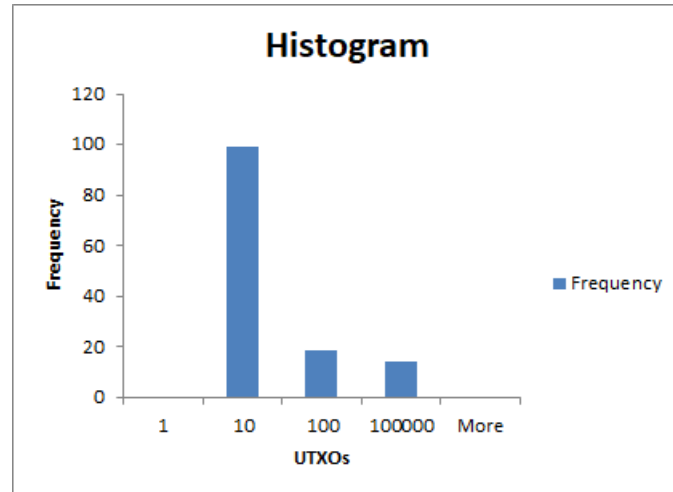


Figure 1: Tần suất UTXOs trong các địa chỉ

```
<Cấu trúc file "dat.dat"> [frame=single]
/***** OPL 12.9.0.0 Data * Author: Tanaka Kai * Creation
Date: May 5, 2019 at 2:39:14 PM *****/ datFiles = "5ad448a959678302d59e6f7
"5ad44bfdce94cf05c955f862.dat", "5ad44e0ece94cf05c955f864.dat", "5ad44e1bce94cf05c955f865.dat", .....
..... "5ad4c6024c372215dd13d6bd.dat", "5ad4cb594c372215dd13d6be.dat", "5ad4cb834c372215dd13d6bf.dat";
<Cấu trúc 1 file dữ liệu "[ID].dat"> [frame=single]
input = < 2, 1, 61480, 1048576, 1.38313609467456, 756, 756, 34, 1352, 330, 0, 0, >;
UTXOs = <1, 148, 30234>, <2, 148, 33116>, ;
output = <1, 34, 61480>, ;
```

Output format:

Sau khi xử lý bằng IBM CPLEX Optimization Studio 12.9.0, kết quả được đưa về dạng file csv để thuận tiện cho việc xử lý sau này.

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
ID	NumUTX	OTS_MD1	NumUTX	Time_MD	NumUTX	TS_25%	Time_25%	NumUTX	TS_50%	Time_50%	NumUTX	TS_75%	Time_75%						
1	5ad448a9	99	216	1	0.093	1	216	0.016	1	216	0	2	364	0.109					
2	5ad44bfd	2	364	2	0	2	364	0	2	364	0	2	364	0					
3	5ad44e0e	3	250	1	0.031	1	250	0	1	250	0.016	2	398	0.015					
4	5ad44e1b	2	364	2	0	2	364	0	2	364	0.015	2	364	0					
5	5ad4503e	5	364	2	0.015	2	364	0	3	512	0	3	512	0					
6	5ad4517c	16	250	1	0.047	1	250	0.031	1	250	0.031	2	398	0.032					
7	5ad4519c	6	364	2	0	2	364	0	3	512	0	3	512	0.016					
8	5ad45307	2	364	2	0	2	364	0.016	2	364	0	2	364	0					
9	5ad45353	3	546	3	0	3	546	0.015	3	546	0	3	546	0					
10	5ad4537e	2	398	2	0	2	398	0.016	2	398	0.016	2	398	0					
11	5ad45471	19863	3616	1	1.11	7	4470	15.328	13	5358	4.235	19	6246	4.688					
12	5ad45a85	19880	3582	1	2.75	7	4436	4.922	13	5324	6.171	19	6212	3.078					
13	5ad45aac	7	330	2	0.047	2	330	0	3	478	0.016	3	478	0.031					
14	5ad45bc6	3	398	2	0.016	2	398	0	3	546	0	3	546	0					
15	5ad45f55	8	216	1	0.047	1	216	0	1	216	0.062	2	364	0.047					
16	5ad46006	8	364	2	0.015	2	364	0	3	512	0	3	512	0					
17	5ad4601c	2	364	2	0	2	364	0	2	364	0	2	364	0					
18	5ad46027	10	512	3	0	3	512	0.063	4	660	0	5	808	0.031					
19	5ad462d6	2	398	2	0	2	398	0	2	398	0	2	398	0					
20	5ad46427	4	364	2	0.078	2	364	0.015	2	364	0.015	2	364	0.031					
21	5ad46607	6	546	3	0.031	3	546	0.031	4	694	0.062	5	842	0.015					
22	5ad46696	2	398	2	0	2	398	0.016	2	398	0	2	398	0					
23	5ad466b8	3	398	2	0.016	2	398	0	3	546	0	3	546	0					
24	5ad46769	4	660	4	0	4	660	0	4	660	0	4	660	0					
25	5ad46823	2	364	2	0.015	2	364	0	2	364	0	2	364	0.016					

Figure 2: Output format

Implementation in GLPK/AMPL: IBM CPLEX Optimization Studio 12.9.0

```
<File "sub.mod"> [frame=single]
/***** OPL 12.9.0.0 Model * Author: Tanaka Kai * Creation
Date: May 5, 2019 at 2:41:48 PM *****/ tuple vin int vid; int
vszie; int vValue;
```

```

tuple vout int vid; int vsize; int vValue;
tuple inputSet int n; int m; float outValue; float M; float alpha; float T; float epsilon; int beta; int txsize;
int iosize; float cout; float coutValue;
inputSet input = ...;
vin UTXOs = ...;
vout output = ...;
int outputSize = sum(e in output) e.vsize;
dvar boolean a[UTXOs]; dvar int+ zsize; /*range n = 1..input.n;*/ //dexpr float minv=min(forall(e in
UTXOs, a[e] diff 0)) a[e]*e.vsize; dexpr int inputSize = sum(e in UTXOs) a[e]*e.vsize; dexpr int inputValue =
sum(e in UTXOs) a[e]*e.vValue;
dexpr int transactionSize = inputSize + outputSize + zsize; dexpr float zvalue = inputValue - (input.outValue
+ input.alpha*transactionSize); minimize transactionSize;
subject to cons0: (zvalue <= input.epsilon - 0.0001) => (zsize == 0); (zvalue >= input.epsilon) => (zsize
== input.beta); cons1: inputSize + outputSize + zsize <= input.M; cons2: forall (out in output) out.vValue
>= input.T; cons3: zvalue >= 0;
int numUTXO = sum(e in UTXOs) a[e];
execute writeln(numUTXO);
<File "sub_2.mod"> [frame=single]
/***** OPL 12.9.0.0 Model * Author: Tanaka Kai * Creation
Date: May 5, 2019 at 8:39:24 PM *****/ tuple vin int vid; int
vsize; int vValue;
tuple vout int vid; int vsize; int vValue;
tuple inputSet int n; int m; float outValue; float M; float alpha; float T; float epsilon; int beta; int txsize;
int iosize; float cout; float coutValue;
inputSet input = ...;
vin UTXOs = ...;
vout output = ...;
float k = ...; float Y = ...;
float y_opt = (1 + k) * Y; int outputSize = sum(e in output) e.vsize;
dvar boolean a[UTXOs]; dvar int+ zsize;
dexpr int inputSize = sum(e in UTXOs) a[e]*e.vsize; dexpr int inputValue = sum(e in UTXOs) a[e]*e.vValue;
dexpr float zvalue = inputValue - (input.outValue + input.alpha*Y); dexpr int transactionSize = inputSize +
outputSize + zsize; dexpr int selected = sum(e in UTXOs) a[e]; dexpr float nUTXOs = sum(e in UTXOs) a[e]
- zsize/input.beta;
maximize nUTXOs;
subject to cons0: (zvalue <= input.epsilon - 0.000001) => (zsize == 0); (zvalue >= input.epsilon) => (zsize
== input.beta); cons1: inputSize + outputSize + zsize <= input.M; cons2: forall (out in output) out.vValue
>= input.T; cons3: zvalue >= 0; cons4: transactionSize <= y_opt;
execute writeln(nUTXOs);
<File "ASS1.1.mod"> [frame=single]
/***** OPL 12.9.0.0 Model * Author: Tanaka Kai * Creation
Date: Apr 21, 2019 at 2:43:14 PM *****/ string datFiles = ...;
main var source = new IloOplModelSource("sub.mod"); var cplex = new IloCplex(); var def = new
IloOplModelDefinition(source); var i = 1; for (var datFile in thisOplModel.datFiles) var opl = new IloOplModel(def,
cplex); var data2 = new IloOplDataSource(datFile);
opl.addDataSource(data2); opl.generate(); if (cplex.solve()) opl.postProcess(); var o = new IloOplOutput-
File(i+".txt"); o.writeln("OBJ= "+ cplex.getObjValue()); o.writeln(cplex.getSolvedTime()); o.writeln(opl.numUTXO);
o.writeln(opl.input.n); o.close(); writeln("OBJ= "+ cplex.getObjValue()); else writeln("No solution"); opl.end();
i++;

```

Experimental results

Mô hình đã đề xuất được giải bởi IBM CPLEX Studio Optimization 12.9.0. Giá trị của γ trong Model 2 được thực nghiệm lần lượt là 25%, 50%, 75%. Kết quả lấy được sẽ được so sánh với kết quả sẵn có của các phương pháp HVF, LVF, OF.

***Nhận xét:** hai mô hình đưa ra giảm số lượng UTXOs được chọn rất nhiều so với 2 phương pháp LVF và OF.

Method	HVF	LVF	OF	Model 1	Model 2($\gamma=25\%$)	Model 2($\gamma=50\%$)	Model 2($\gamma=75\%$)
NoSelected	330.00	1361.00	855.00	315.00	373.00	469.00	559.00

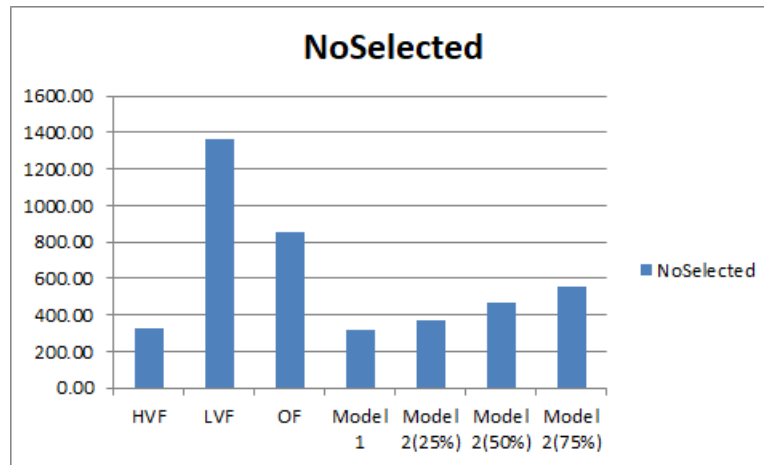


Figure 3: Số lượng UTXOs được chọn ở mỗi phương pháp

Method	HVF	LVF	OF	Model 1	Model 2($\gamma=25\%$)	Model 2($\gamma=50\%$)	Model 2($\gamma=75\%$)
RunTime	69	79	62	13.234	74.622	81.703	52.393

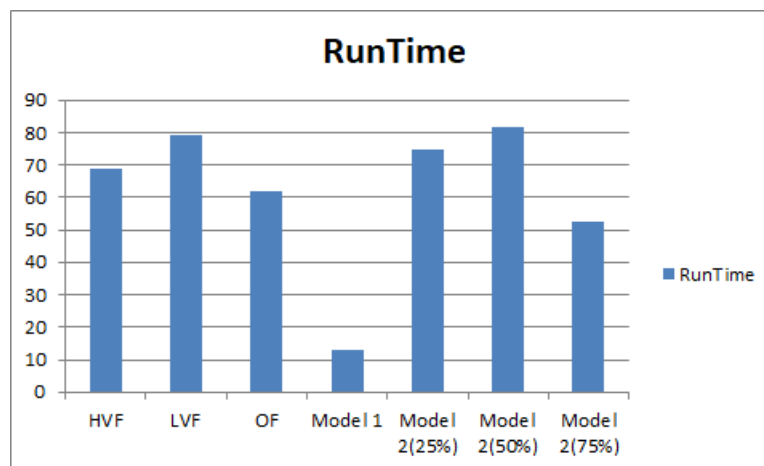


Figure 4: Thời gian chạy giải thuật của mỗi phương pháp

Method	HVF	LVF	OF	Model 1	Model 2($\gamma=25\%$)	Model 2($\gamma=50\%$)	Model 2($\gamma=75\%$)
TransactionSize	399724	551632	477526	85618	93794	107900	121152

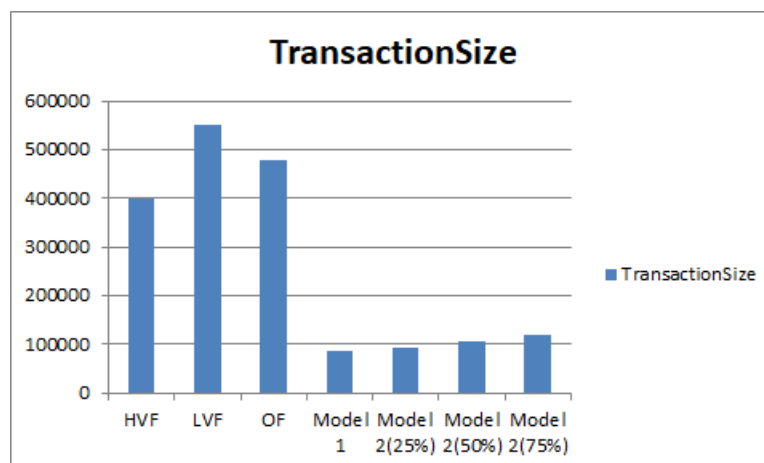


Figure 5: Tổng kích thước giao dịch của mỗi phương pháp

Từ những số liệu trên, ta có thể thấy mô hình 1 tuy có thời gian chạy rất thấp nhưng số lượng UTXOs được chọn cũng rất thấp. Trong khi đó, ở mô hình 2, với những giá trị γ để kiểm soát lần lượt là 25%, 50%, 75%, ta có thể thấy thời gian chạy vẫn được giữ xấp xỉ với các mô hình HVF, LVF, OF, nhưng số lượng UTXOs được chọn đã được tăng lên đáng kể, đồng thời kích thước giao dịch vẫn tương đối ổn định so với mô hình 1.

5 Conclusion

Trong bài báo cáo này, nhóm chúng em đã hiện thực hai mô hình đã đưa ra để đánh giá về việc giải quyết hai mục tiêu thiết yếu khi tạo giao dịch mới trên blockchain. Đó là giảm thiểu kích thước giao dịch để có phí giao dịch tối thiểu và thu nhỏ tập UTXO. mô hình 1 tối thiểu được kích thước giao dịch tuy nhiên không giải quyết được nhu cầu thu nhỏ tập UTXO nên chúng ta đưa ra mô hình 2 để giải quyết được cả 2 vấn đề này. Mô hình 2 được đưa ra để vừa có thể chọn được càng nhiều UTXO càng tốt nhưng không làm cho kích thước giao dịch quá lớn, do chúng ta đã kiểm soát bằng tỉ lệ gamma nên luôn đảm bảo kích thước giao dịch không lớn quá mức cho phép làm tăng phí giao dịch. Qua đó ta có thể thấy được so với các mô hình hiện tại là HVF và LVF thì rõ ràng mô hình trên hiệu quả hơn. Mặc dù cần được thử nghiệm với các tập dữ liệu lớn hơn nữa nhưng rõ ràng mô hình đã đưa ra hoàn toàn khả thi để đưa vào thực tế.

References

- [1] wikipedia. “**link: <http://en.wikipedia.org/>**”, , last access: 05/05/2015.
- [2] Frey, D., Makkes, M. X., Roman, P.-L., Taiani, F., Voulgaris, S.: Bringing secure Bitcoin transactions to your smartphone. The 15th International Workshop on Adaptive and Reflective Middleware, (2016).
- [3] Antonopoulos, A. M.: Mastering Bitcoin. 2nd edn. O'Reilly Media, CA 95472 (2014).
- [4] Bitcoinjs: Open Source Organisation for Bitcoin JavaScript Libraries,<https://github.com/bitcoinjs>. Last accessed 15 August 2018.
- [5] Bitcoinj: Library for working with the Bitcoin protocol,<https://bitcoinj.github.io>. Last accessed 10 August 2018.
- [6] Yanovich, Y., Mischenko, P., Ostrovskiy, A.: Shared Send Untangling in Bitcoin, White paper, Bitfury Group Limited (2016).
- [7] Dai, P., Mahi, N., Earls, J., Norta, A.: Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform, <https://qtum.org/uploads/files/cf6d69348ca50dd985b60425ccf282f3.pdf>, (2016).
- [8] Sergi, D.-S., Cristina, P.-S., Guillermo, N.-A., Jordi, H.-J.: Analysis of the Bitcoin UTXO set, IACR Cryptology ePrint Archive, (2017).
- [9] Erhardt, M.: An Evaluation of Coin Selection Strategies, Master thesis, Karlsruhe Institute of Technology, URL: <http://murch.one/wp-content/uploads/2016/11/erhardt2016coinselection.pdf>, (2016).
- [10] Zahmentferner, J.: Chimeric ledgers: Translating and unifying utxo-based and account-based cryptocurrencies, Cryptology ePrint Archive, Report 2018/262, 2018. <https://eprint.iacr.org/2018/262>, (2018).
- [11] Chepunoy, A., Kharin, V., Meshkov, D.: A Systematic Approach To Cryptocurrency Fees. IACR Cryptology ePrint Archive, (2018).