Data-driven analysis of Bitcoin properties: exploiting the users graph

Article *in* International Journal of Data Science and Analytics · August 2018

DOI: 10.1007/s41060-017-0074-x

Damiano Di Francesco Maesa
University of Cambridge
18 PUBLICATIONS **550** CITATIONS
SEE PROFILE

Laura Ricci
Università di Pisa
152 PUBLICATIONS **1,392** CITATIONS
SEE PROFILE

Noname manuscript No.
(will be inserted by the editor)

Data driven analysis of Bitcoin properties: exploiting the users graph

Damiano Di Francesco Maesa · Andrea Marino · Laura Ricci

the date of receipt and acceptance should be inserted later

Abstract Data analytics has recently enabled the uncovering of interesting properties of several complex networks. Among these, it is worth considering the BITCOIN blockchain, because of its peculiar characteristic of reflecting a niche, but also a real economy whose transactions are publicly available. In this paper we present the analyses we have performed on the users graph inferred from the BITCOIN blockchain, dumped in December 2015, so after the occurrence of the exponential explosion in the number of transactions. We first present the analysis assessing classical graph properties like densification, distance analysis, degree distribution, clustering coefficient, and several centrality measures. Then, we analyse properties strictly tied to the nature of BITCOIN, like rich-get-richer property which measures the concentration of richness in the network.

1 Introduction

The study of methods and tools for the analysis of complex networks has recently gained momentum, due to the presence of complex relational data in different fields. Network analysis has been applied in different scientific areas, like the analysis of biological systems [1], transportation systems [2] and social networks [3]. A novel application field is that of the networks modelling economic transactions occurring in some economic area. However, the analysis of real-life economy networks is not easy as there is no central entity registering all the transactions, since the transaction records are distributed over a large number of commercial entities or banks. An exception is that of transactions generated by digital cryptocurrencies which have been recently proposed to enable a point to point value exchange, so overcoming the need of a third party financial intermediary. Current cryptocurrencies require a distributed public ledger to work, so providing a unique opportunity for analysis of currency transactions.

BITCOIN [4], the first true digital currency, was proposed in 2008 by Satoshi Nakamoto, a pseudonym, and the first client went online in 3rd of January 2009. From then, the system has gained wide mass media coverage and widespread popularity among the broad public of non specialists, so resulting in the first example of cryptocurrency economy worthy of analysis. After almost seven years since the inception of BITCOIN, an economic community has risen around it. BITCOIN still represents a niche and peculiar economical community, nevertheless its importance in the real world has grown enough so that it no longer represents an experimental currency exploited only by computer science specialists. Several events of the BITCOIN economy, like the wild speculation, the value fluctuation and a major exchange failure witness that a true economic system has born around it.

The Bitcoin system operates according to a peer-to-peer philosophy, which avoids the need of a bank account maintained by a central authority. In BITCOIN, each user has a unique address that consists of a pair of public and private keys. Each amount is associated

Damiano Di Francesco Maesa
Department of Computer Science
University of Pisa, Italy
E-mail: damiano.difrancescomaesa@for.unipi.it

Andrea Marino
Department of Computer Science
University of Pisa, Italy
E-mail: marino@di.unipi.it

Laura Ricci
Department of Computer Science
University of Pisa, Italy
E-mail: laura.ricci@unipi.it

to the public key of the owner, and the private key is exploited to sign each payment the owner performs. All the transactions are grouped in blocks, recorded in a distributed ledger, the block-chain, and validated through a distributed consensus procedure. Therefore, BITCOIN keeps the entire transactions history public by design, so it represents one of the few economic communities that can be studied and analyzed in depth.

A distinctive characteristic of BITCOIN is that it puts together economic and technological aspects. This produces interesting interrelations between the BITCOIN real world economy and the technological aspects of the distributed protocol. These interrelations have influenced its development as much as its original design principles.

Several interesting aspects of the protocol are currently under investigation. These range from the improvement of the basic protocol with the definition of new anonymity mechanisms and consensus algorithms, to the analysis of the information mined from the block chain. This paper focuses on the second aspect, i.e. the definition and analysis of the users graph derived from the BITCOIN blockchain. The transactions encoded in the blockchain may be modelled by a multigraph, the transaction graph, whose nodes correspond to addresses, i.e. hash of public keys and edges correspond amount transfers between addresses. The users graph is derived from the transaction graph by a clustering process. Indeed, since each user may control several addresses, these can be grouped in a single cluster, by adopting some heuristics. Each cluster should ideally correspond to a single user. In this way, a new graph is defined, the users graph, whose nodes correspond to the users and whose edges correspond to value transfers between users.

Even if several works [5–10] present analyses of the BITCOIN users graph to find out some interesting property of its economy, almost all of them consider an outdated state of the block chain, mainly an instance of the block-chain at the end of 2013. As shown in [11], the number of transactions has increased from roughly 10 millions in January 2013 to more than 100 millions of transactions in January 2016. This huge economic explosion, occurred in the last years, makes it interesting to analyse the features of the users graph with reference to a more recent state of the blockchains. We believe that an analysis of an up-to-date log of the block-chain may enable a deeper understanding of the BITCOIN network and also highlight novel characteristics of the network arisen in the last few years. On the other hand, the analysis of a so huge amount of data, requires the definition of proper tools to perform both classical and more complex analyses.

In [12] we have presented our support for the analyses of huge amount of data and described a preliminary set of analysis of the BITCOIN users graph performed by that support. The analysis spotted peculiar topological properties of the users graph which are new if compared with other complex networks. We have shown that the topological observations on the users graph can translate in emerging economical trends. In this sense, we can highlight economical outlier trough the observation of topological phenomena and verify economical hypotheses, as rich get richer.

This paper extends our previous work in several directions. Besides giving a more comprehensive survey of related works and a more detailed description of BITCOIN protocol, we present a larger set of experiments, including:

- an analysis of the clustering coefficient of the users graph which shows how it is basically constant over time and that its order of magnitude is similar to the one of the other social complex networks.
- the computation and interpretation of the results of an extended set of centrality measures including also Page-Rank and Eigenvector indexes.
- a refined analysis of the properties we have defined in [12] regarding the concentration of richness in the network. First of all, we propose a characterization of active users based on balance thresholds, explaining the reason behind this choice. We then perform a new rich get richer analysis, by restricting it to active nodes only. We also compute the Gini coefficient of the users graph, as a further measure to support our results.

The paper is organized as follows: Section 2 presents the related works, while Section 3 gives a brief overview of the BITCOIN protocol, while in Section 4 we describe the clustering algorithm. Section 5 presents the results of the analyses. Finally, Section 6 discusses the conclusions and presents the future work.

2 Related Works

Several analyses of the BITCOIN network have been recently proposed. Most of them take in input the "user graph" that is extracted from the transactions graph through a well established heuristic rule. This rule, already introduced in the seminal paper [4], and extensively described in [13], establishes that all the input addresses of a multi-input transaction belong to the same user. The rule is based on the observation that every input of a multi-input transaction must be signed with the right private key and this implies that the signer knows all the private keys of the transaction and so it is the owner of all the input addresses. The resulting graph approximates the real users graph, because the heuristics may underestimate or overestimate the common ownership of some addresses. While underestimation occurs because addresses of the same owner have not been used in the same transaction, overestimation may occur because a set of users may collectively sign the same transaction [14]. The heuristic rule has been subsequently used in most analysis, like in [5–10]. An exception are [6, 8], that also introduce a more sophisticated heuristic based on change addresses, i.e. the mechanism used to give money back to the input user in a transaction. [15] shows that the multi-input address heuristics generally exploited to derive the users graph is really effective and investigates the reason behind this observation. These are the high-levels of address reuse and avoidable merging; the existence of super-clusters with high centrality, and the incremental growth of address clusters. This encourage us to exploit this heuristics for the definition of the users graph, as shown in the following sections.

Let us now briefly review the most important analyses recently proposed. [5] considers only BITCOIN transactions carried out until May 2012. They discovered that the network contains a huge number of small transactions, but also a subset of transactions moving a large amount of money. The analyses are then focused on the large transactions in order to detect the ways amounts are accumulated and dispersed.

[9] does not apply any heuristic and directly analyses the transaction graph, extracted from the blockchain, whose state is considered as in May 2013. The authors identify an initial phase of growth of the BITCOIN network, characterized by a large fluctuation in the network characteristics and a trading phase characterized by more stable network measures. They find out that preferential attachment drives the growth of the network. The authors also analyse the Gini coefficient of the indegree distribution over time, doing something similar to what we do in Section 5.4, but the results are not comparable since their study is based on the simple transaction graph.

The main focus of the analyses presented in [6] is to highlight the gap between the potential and the actual anonymity of the BITCOIN protocol. The authors apply to the blockchain, as in April 2013, the two aforementioned heuristics to contract the transaction graph.

As most previous works, [10] considers the blockchain state at April 2013 and exploits only the first heuristic previously described for the contraction of the transaction graph. The authors categorize the transactions according to business categories by extracting the business tags of each address. Furthermore, they present an analysis of the geographic distribution of BITCOIN transactions.

The authors of [16] analyse the transaction networks of Bitcoin and Litecoin digital currencies. The analyses are applied to a graph where the nodes are the addresses of the Bitcoin users, while the edges are the transaction between two addresses. Since no clustering is performed on the transaction graph, the results are characterized by a high level of approximation. For instance, since the top richest nodes detected in the network correspond to BITCOIN addresses, rather than users, it is possible that a richest user exist which exploits different addresses to store its bitcoins.

[17] presents a set of analyses revealing the presence of a set of unusual topological patterns in the BITCOIN users graph, for instance a set of outliers are detected in the in-degree distribution of the BITCOIN users graph. The authors show that these patterns are not due to normal economic behaviours, but to artificial transactions whose nature is conjectured in the paper.

3 The bitcoin protocol

Users take part in the BITCOIN economy through addresses. An address is a double hash (firstly SHA-256 [18] is applied and then Ripemd-160 [19]) of a public key derived form a ECDSA key pair [20]. The address (and hence the public key) will be used by the user to send and receive payments, while the private key will be used by the user to provide proofs of ownership. Creating new ECDSA pairs (and so addresses) is not expensive at all and so each user can create and use multiple addresses. This leads to the use of pseudonyms. Pseudonymity means that each address carries no information of the identity of its owner and two addresses controlled by the same entity share no information with each other. In other words there is no linking between addresses and identities or other addresses. Pseudonymity is the only (weak) anonymity protection in BITCOIN. So to improve transactions privacy is recommended to create a new address to receive each new payment. While this is not computationally expensive, it can lead to an address management problem if the number of addresses keeps increasing.

To exchange funds between addresses, transactions are created. Transactions are multi input, multi output, it means that a transaction may have more than one input (address from which funds are withdrawn) and more than one output (address where funds are stored). Each transaction completely transfers funds from the inputs to the outputs (no change is left in the input addresses), so a change address (controlled by the paying input address corresponding owner) must be added among the transaction outputs to collect the change. Transactions are the only mean to manage funds, so funds can be divided or aggregated only by being spent. That is possible because a transaction involves addresses and not users and every user can have different addresses, so the user can use a transaction to split, merge or move funds between its own addresses. A transaction can also specify a voluntary fee to cover the expenses of the validation process (that we will explain briefly later). If the sum of input values exceeds the sum of output values then the exceeding value is considered a voluntary fee paid to the validator. In a transaction each output can be seen as a couple (amount, receiver address). Each input specifies, instead, where to withdraw the funds, so it does indicate an address only on abstraction, but in fact it indicates the previous transaction (through its hash) where the funds were created. Funds are represented by a transaction chain showing the passage of value (split and merge) between addresses, validated at each step by the previous owner signature. In BITCOIN transactions alone specify the entire state of the system. There is no coin exchanged between users, the coins are implicitly represented by the flow of value through transactions. New transactions are created by any user and notified to the community with a gossip style broadcast message on the P2P BITCOIN network. We also note that a special kind of transaction called *coinbase* exists to allow for new value creation (distributed minting of new coins as part of the validation process) and fees collection. These special transactions have no inputs, only output addresses to whom newly minted value and fees are credited.

In a transaction each input is signed by the owner with the private key corresponding to the address spending the funds. This digital signature guaranties that only the rightful owner can spend its funds, but it does not prevent it from spending them more than once in different transactions. This is the so called *double spending problem*, present in most cryptocurrencies. BITCOIN solution is to remember the history of all the past transactions to determine the actual owner of a fund, at each given time. The history is maintained in a distributed database called blockchain. Transactions are grouped in blocks linked in a chain and the linking between blocks is achieved by saving the hash of the header of the previous block in the next block header. To make each block header (and so its hash) dependent from all transactions contained in that block, the root of the (implicit) Merkle tree [21], built from the block transactions hashes is included in the header. It is necessary to reach a distributed consensus to choose which block (and so which transactions) to add to the chain, because there could be incompatible transactions caused by a double spending attempt. The distributed consensus protocol introduced and used by BITCOIN is called Nakamoto consensus and relies on HashCash Proof-of-Works (PoW) [22]. This Nakamoto consensus protocol is one of the most interesting aspects of BITCOIN but we will discuss it only briefly since it's beyond the scope of this paper.

The consensus protocol is used to agree on the valid blockchain so consensus determines which transactions are remembered and their partial ordering. Any user can choose to take part in the consensus protocol and become a validator. Validators are called *miners* and the entire validation process is called *mining*. The consensus protocol is divided in steps. At each step every participant chooses a list of valid transactions and builds a block out of those. Then a POW based distributed process (explained later) is employed to choose at random one of the participants to publish his block to be added as next block on the head of the blockchain. Each participant then chooses if the newly published block is to be accepted or rejected. This choice is manifested by deciding to start looking for the next block on top of the new one (accept)or by keeping looking for a block on top of the old chain, ignoring the new block (reject). In the latter case we say that the blockchain as been *forked*, since, during the next step the participants will look for new blocks on different branches of the chain. In case of conflicting chains coexisting at any given time the protocol dictates to look for new blocks (and add them) only on the longest chain branch (longest regarding the cumulative difficulty) to try to maintain only one branch and hence only one official transactions history, since the shortest branches will eventually fall behind and be ignored. It's important to point out that the participants might be malicious and so might ignore the protocol rules and keep adding blocks on a shorter branches. The consensus protocol guarantees that an eventual consensus is reached on the longest branch in the presence of this malicious behavior, as long as the majority of participants is compliant [23].

We still have to explain what is the distributed process adopted to pick a random participant at every step. This distributed process should be deployed in a P2P network and so should be resistant against Sybil attacks. In a pseudonymous P2P scenario is cheap to create big amounts of fake identities controlled by the same entity. This prevents the protocol from using ids. So the process proposed should relay on a resource not so easy forgeable as identities. The resource chosen in Bitcoin is computing power [4]. That means that we chose to use a distributed process that picks nodes at random with probabilities proportional to the nodes computational power dedicated to said process. This allows the protocol to be resilient in the face of computationally bounded adversaries [24] and can be achieved trough a PoW scheme.

Finding a proof-of-work means finding the solution of a computationally intense cryptographic puzzle to prove that some amount of effort was spent. The puzzle used should be asymmetric, which means that it should be computationally difficult to find a solution, but, given a solution, should be computationally easy to verify it. It should also allow to adjust the difficulty (of finding the solution, not of verifying it), and it should be dependent from some parameter to allow to have different puzzles with the same difficulty. This is needed to make the solution effort independent of any a priori computation.

In Bitcoin validating a block means finding a hashcash [25] type proof-of-work with double SHA-256 [18] of the block header. It means finding a *nonce* value to be included in the block header so that the double SHA-256 of such header is less that an established *target* value. The *target* value is automatically updated every 2016 blocks (approximately every two weeks) considering the computational power of the entire network (estimated by the average time passed to validate a block) in order to keep the average validation rime of a new block around ten minutes.

Solving this type of proof-of-work is equivalent to an hash partial inversion, and, since the hash function chosen is cryptographic secure, the best known method is a brute force attack. This means that the best method is trying different random *nonces* until one satisfies. So the average time spent depends only on the computational power (more precisely the hash power) used. This cryptographic puzzle is a PoW because it is computationally expensive to solve but constant to verify (requires only one hash computation), the difficulty is adjustable (changing the target value) and the puzzle is parametric, the parameter being the block header, so it is not possible to work on the puzzle without knowing the block (no a priori advantage possible). Moreover Bitcoin PoW is a probabilistic PoW, that means that finding a solution is computationally expensive on expectation. The randomization is important because otherwise the participant with the biggest computational power would always find the solution first and so would be the only one to produce blocks. The randomized PoW instead allows participants to have a probability to find each block proportional to the hash power dedicated (as long as no single miner controls more than half of the total combined hashing power).

Since solving a PoW requires an important computational effort, the resources dedicated to look for new blocks are expensive and so an incentive mechanism is requested to motivate participants to take part in the consensus protocol. To encourage mining and repay miners for their computational expenses, at each new block is associated a reward collected by the block finder. The reward consists in the sum of all fees of the transactions contained in the block, plus a fixed amount of new coins. This reward is credited to the miner allowing it to add a special *coinbase* transaction to its blocks. The fixed reward starts from 50 BTC and halves over time every 210000 blocks until it will became zero at the 6930000th block, which is expected to be mined during the year 2140 (the decreasing coin minting is adopted to reflect a deflationary money supply growth).

4 Building the Users Graph

In this section, the clustering process which generates the users graph is presented. We describe the clustering algorithm, some statistics on the cluster generated by the algorithm and finally the data acquisition process we have performed.

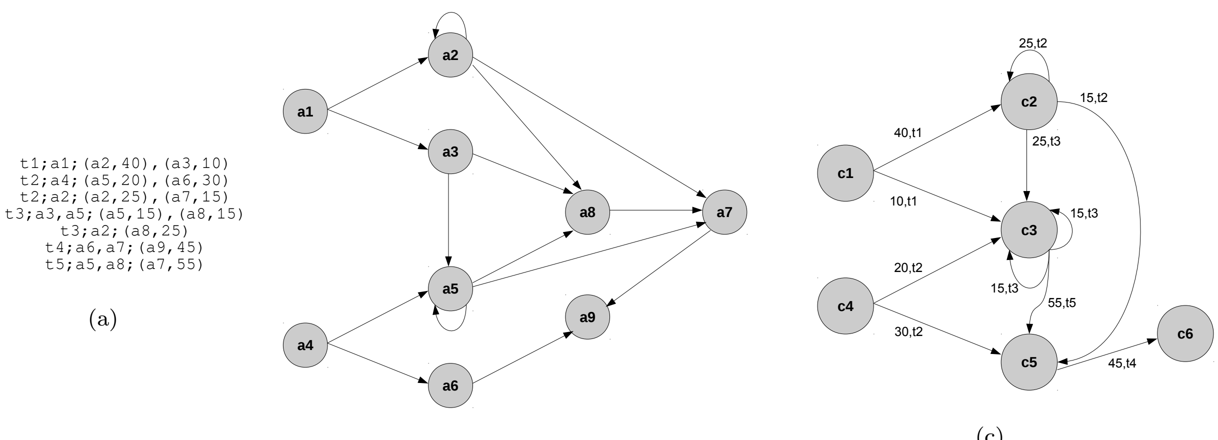
4.1 The Clustering Algorithm

The BITCOIN dataset can be formally modelled by a weighted directed hypergraph $H = (A, T)$ where: A is the set of all addresses; T is the set of transactions, which can be modeled as a set of ordered pairs (A_1, A_2) with $A_1, A_2 \subseteq A$, meaning that the addresses in A_1 are paying the addresses in A_2 (see for instance [9]).

Moreover, to each transaction $s = (A_1, A_2) \in T$, we associate:

- a timestamp telling when the transaction took place.
- a distribution of amounts among the nodes in A_2 denoted as b_s. More formally, b_s is a function associating to each $a \in A_2$ a multiset of real numbers. Indeed, notice that there can be transactions associating to the same $a \in A_2$ more than one single amount.
- a fee ϕ_s (eventually 0) that associates to A_1 the voluntary taxes payed.

As seen in the previous section 3, in BITCOIN each user controls different pseudonymous addresses. In order to infer the users of the network, we want to cluster all the addresses managed by the same user so that each cluster will ideally correspond to a single user. In particular, we group addresses according to the following desired property.



```
t1;a1;(a2,40),(a3,10)
t2;a4;(a5,20),(a6,30)
t2;a2;(a2,25),(a7,15)
t3;a3,a5;(a5,15),(a8,15)
t3;a2;(a8,25)
t4;a6,a7;(a9,45)
t5;a5,a8;(a7,55)
```

(a) (b) (c)

Fig. 1 Simple example of users graph. (a) contains a list of simplified transactions, where each transaction is expressed in the format `timestamp ; comma separated list of input addresses ; comma separated list of couples ( output address , amount )`. (b) represents the corresponding transaction graph and (c) shows the derived users graph where cluster c3 contains the addresses a3, a5 and a8 and cluster c5 contains the addresses a6 and a7.

Property 1 For every two addresses x and y, if there exists a transaction (A_1, A_2) where $x, y \in A_1$, then x and y belong to a same cluster.

Note that this is a sufficient but not necessary condition for x and y to be in the same cluster, since we merge clusters transitively: if for instance x and z belong to A_1 for some transaction $(A_1, A_2) \in T$ and z, y belong to A_3 for some transaction $(A_3, A_4) \in T$, then x, z, y will be assigned to the same cluster.

The clustering algorithm works as shown in procedure CLUSTER in Algorithm 1. Given the hypergraph $H = (A, T)$ above, the clustering algorithm produces a partition of A, i.e. the clusters $C_1, \ldots, C_k \subseteq A$ for some k, with $C_i \cap C_j = \emptyset$ $(1 \leq i, j \leq k\ ,\ i \neq j)$ and $C_1 \cup \ldots \cup C_k = A$. We define the undirected graph G_H whose nodes are all the addresses in A and two nodes $x, y \in A$ are linked whether there exists a transaction $(A_1, A_2) \in T$ such that $x, y \in A_1$. Then the k connected components of G_A are our clusters $C_1, \ldots, C_k$.

The following result holds.

Lemma 1 *Given $H = (A, T)$, the clustering corresponding to the connected components $C_1, \ldots, C_k$ of G_H satisfies Property 1.*

It is worth observing that building G_H as described above can be costly: for each transaction $(A_1, A_2) \in T$, we have to create a clique among all the nodes in A_1, adding a quadratic number of edges, i.e. $|A_1| \cdot (|A_1| - 1)/2$. Instead of creating a clique, procedure CLUSTER in Algorithm 1 adds a simple path between the addresses in A_1, adding each time a linear number of edges. In other words, CLUSTER creates a graph G'_H whose set of nodes is A and whose set of edges is given by the following process: for each transaction $(A_1, A_2) \in T$, create a path among the nodes in A_1. The following property trivially holds.

Algorithm 1: THE GRAPH BUILDING PROCESS

```
Input  : A weighted directed hypergraph $H = (A, T)$,
         $b_s$ for each $s \in T$
Output: A directed multigraph $G = (V, E, w)$
Procedure CLUSTER($H$)
    $G'_H = (A, E') \leftarrow$ undirected graph with $A$ as set of
      vertices and $E'$ empty set of edges
    foreach $s = (A_1, A_2) \in T$ do
        Let $A_1 = \{a_1, a_2, \ldots, a_h\}$
        for $i \in \{1, \ldots, h-1\}$ do add $\{a_i, a_{i+1}\}$ to $E'$
    Let $C_1, \ldots, C_k$ be the connected components of $G'_H$
    return $C_1, \ldots, C_k$
$C_1, \ldots, C_k \leftarrow$ CLUSTER($H$)
Let $c(a)$ be the vector associating to each $a \in A$ the
  cluster $C_j$ such that $a \in C_j$
$\phi(C_i) \leftarrow 0$ for each $C_i$.
$G = (V, E, w) \leftarrow$ graph where $V = \{C_1, \ldots, C_k\}$ is the
  set of nodes, $E$ is an empty set of arcs, $w : E \rightarrow \mathbb{R}$
foreach $s = (A_1, A_2) \in T$ do
    Let $C_i$ be the unique cluster $c(a_1)$ for any $a_1 \in A_1$
    $\phi(C_i) \leftarrow \phi(C_i) + \phi_s$
    foreach $a_2 \in A_2$ do
        Let $C_j$ be $c(a_2)$
        foreach $x \in b_s(a_2)$ do
            Add an arc $e$ from $C_i$ to $C_j$ in $E$ with
              weight $w(e)$ equal to $x$
```

Lemma 2 *G'_H and G_H have the same connected components.*

Once the clusters $V = \{C_1, \ldots, C_k\}$ have been identified in $H = (A, T)$ using G'_H, we create the weighted multigraph G whose set of nodes is V and there is an arc e from $C_i \in V$ to $C_j \in V$ whether there exists a transaction $(A_1, A_2) \in T$ such that $A_1 \cap C_i \neq \emptyset$ and $A_2 \cap C_j \neq \emptyset$. Roughly speaking, there is an arc from a cluster to another whether there exists a transaction from an address of the former to an address of the latter. Note that this is a multigraph since there can be several transactions from a cluster to another, possibly with different (or equal) amount. Moreover, for each value $x \in b_s(a_2)$ with $a_2 \in A_2$, we create an arc with weight x, since, as explained before, a same transaction s can assign more than one amount to a vertex $a_2 \in A_2$. Finally, we define ϕ for each C_i as the sum of ϕ_s for each transaction s payed by C_i.

We will refer to G as BITCOIN users graph. The building method is summarized by Algorithm 1. It is worth noting that the whole process is linear in the size of H, i.e. $O(|A|+\sum_{(A_1,A_2)\in T}(|A_1|+|A_2|)$). An example of users graph is shown in Figure 1.

4.2 Clustering Statistics

For the sake of completeness, in Figure 2 we show the distribution of clusters size. It is worth observing, that this distribution follows a power law.

We report in Table 1 some basic statistics of our dataset, like the total number of addresses and the total number of transactions, and some statistics about the result of our clustering process.

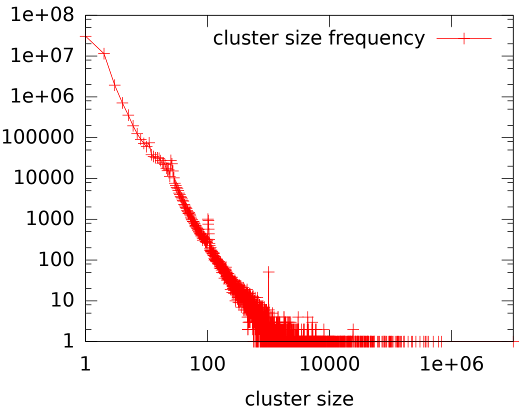


Fig. 2 Distribution of Cluster Sizes.

In the lower part of Table 1 we report the list of the top ten biggest clusters we obtained. We observe that the size of these clusters is several order of magnitude bigger than the average size of the clusters, making the distribution of the clustering size heavy tailed (see Figure 2). We will see that several clusters in this top ten list are in the top ten list for other topological centrality measures.

Number of Addresses, i.e. $\|A\|$	113 221 083
Number of Transactions, i.e. $\|T\|$	99 602 440
Number of Clusters, i.e. nodes of G	46 144 246
Number of Arcs of G	294 705 549

The 10 biggest clusters		
Cluster ID	Identity	Size
66 482	Mt. Gox	10 216 380
2 899 325	LocalBitcoins.com	676 402
26 784 111	GoCoin.com	611 885
11 032 019	AgoraMarket	497 995
12 388 597	EvolutionMarket	420 632
2 477 299	N/A	392 589
2 547 597	SilkRoadMarketplace	372 753
10 072 646	SilkRoad2Market	349 874
1 175 285	BTC-e.com1	348 438
11 828 673	999Dice.com	301 990

Table 1 Some Clustering Statistics.

4.3 Data acquisition

As we explained in the previous section 3 all the BITCOIN transaction history is publicly available in the blockchain as a countermeasure against double spending. To obtain the blockchain is sufficient to set up a BITCOIN node in the P2P network and start requesting blocks to the other nodes. This process can take a lot of time so, to avoid it, we used a blockchain already downloaded and stored in Protocol Buffers format [26]. The blockchain used contains all the first 389 800 blocks, from the Genesis block until block height 389 799, hence containing all the BITCOIN transactions from 2009-01-03 18:15:05 GMT to 2015-12-23 09:40:52 GMT .

In section 3 we have given an high level description of BITCOIN transactions, but in practice the transactions stored in the blockchain contain scripts. The BITCOIN protocol uses a non-Turing complete stack based scripting language, and scripts are (mostly) used in a transaction to specify conditions needed to redeem the funds of that transaction. The most common example of such condition is a signature. When a transaction is tested for validity, the input scripts are concatenated with the output scripts, evaluated, and all transaction scripts must evaluate to true for the transaction to be validated. Scripts can potentially be arbitrarily complex but in practice only few types of standardized scripts are used in transactions. Those scripts and the transactions using them are called *standard*. What's more important is that non-*standard* transactions (so transactions containing non-*standard* scripts) are accepted but not relayed by compliant nodes, so they have less chances of actually ending up in the blockchain. The most used *standard* script types are called Pay to PubKey Hash (`p2pkh`), Pay to PubKey (`p2pk`), Pay to

Script Hash (`p2sh`) and Pay to Multisig (`p2ms`). We have decided to parse the blockchain only interpreting the `p2pkh`, `p2pk` and `p2sh` types of scripts, dropping the transaction outputs containing other kinds of scripts. We did this to not over-complicate our blockchain parser and because we thought that this kind of scripts where the ones used in transactions more suitable to apply our clustering heuristic, hence hoping to reduce the number of false positives returned by the heuristic clustering. In the end we successfully interpreted 295 144 677 scripts and failed to interpret 1 489 903 scripts, resulting in a coverage of 99.4977% of all transaction outputs. So we deem the information loss acceptable.

From the parsing of the raw blockchain with the script interpretation described before, we obtain our transactions dataset and on this dataset we apply our clustering algorithm and perform the analysis. We do not try to infer ourselves addresses identities and instead we rely on the public address tags datasets provided by [27,28]. In the rest of this paper to suppose a cluster identity we look for identity tags (provided by those services) associated to the addresses belonging to that cluster. It's beyond the scope of this paper to evaluate the correctness of those tags.

5 Analysis and Results

In this section we study the topological properties of the BITCOIN users graph G built in Section 4. Recall that G is a weighted directed multigraph. We refer to U as the symmetric version of G, i.e. the graph where all the arcs become undirected.

In Section 5.1, we study the time evolution of G and U. For increasing values of time t we have considered just transactions that took place before t. We indicate with G^t (and U^t), with $1 \leq t \leq 20$, the graph induced by transactions whose time stamp is smaller than t, where t refers to the left part of Table 2. Analogously, we indicate with $\phi^t(u)$ the fees payed by u until time t, i.e. $\phi(u)$ induced by transactions with timestamp smaller than t. The timestamps chosen are at constant intervals in time but with an high initial offset. We chose to start the timestamp snapshots from the beginning of 2013 because we considered it the time when the BITCOIN economy started to rise significantly and was mature enough for a systematical analysis. Moreover, for each graph snapshot at each timestamp considered during the connectivity analysis phase in Section 5.1 we (non recursively) pruned the graph from the nodes with only one incoming arc. We choose to do so because otherwise the graph was biased from more recent nodes artificially isolated by the timestamp cutoff. Indeed, we noticed that most of those nodes corresponded to nodes that had just received a payment and didn't have enough time to use that value in a subsequent transaction. This pruning does not skew our analysis since the nodes pruned were nodes reached by only one incoming arc.

In Section 5.2 we report some centrality analysis based on the connectivity of the last snapshot of the network considering the degrees of the vertices, harmonic centrality and some spectral centrality [29].

In Section 5.3 we define active users and we study how the number of active users changes over time.

In Section 5.4, we define the *richness* of a node according to its number of incoming transactions and its balance. We study how the sets of richest nodes change over time, proving that richness tends to concentrate in terms of balance. We show that this holds even just considering active nodes.

In the remaining part of the section, i.e. Section 5.5, we study the graph G for different transaction amounts. In particular, we call G_a (and U_a) the graph induced by transactions whose amount is smaller than a, with $1 \leq a \leq 12$, where the corresponding values of a are listed in the right part of Table 2. We remark that we have divided our time window in 20 equally spaced time stamps, each one corresponding to roughly 55 days.

Even though the nodes of G and U correspond to clusters of addresses as seen in Section 4, for the sake of simplicity, we will simply refer to them as vertices or nodes. Moreover, we will call the links in G as arcs, which are directed, and the ones of U as edges, which are instead undirected.

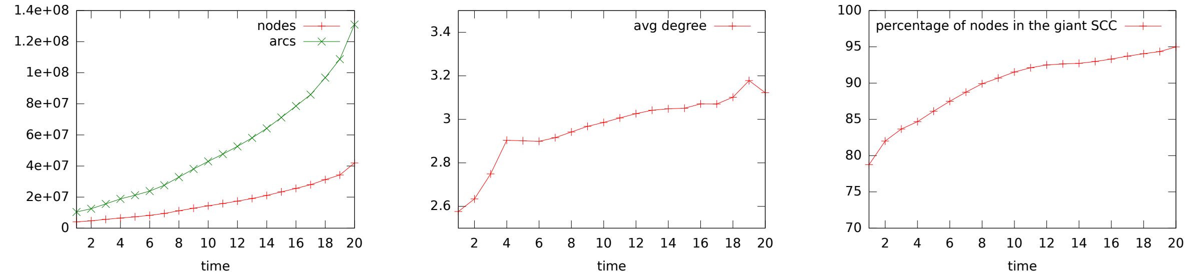
5.1 Connectivity Analysis Over Time

Since the interest of this section relies on the connectivity of the network, we consider G^t and U^t as simple graphs ignoring multiple arcs (or edges). Our analysis framework have been implemented using WEBGRAPH [30].

5.1.1 Densification

This section aims to observe the densification process taking place in BITCOIN. This phenomenon is described by Figure 3.

- Figure 3(a) shows the increase of number of nodes and arcs over time in G^t. Since the time slots are equally spaced, the plot highlights that the increase of nodes and arcs is slightly more than linear.
- Figure 3(b) shows the increase of the average out-degree of the nodes in G^t for increasing values of t (note that the average out-degree and the average in-degree are the same). This plot highlights that



(a) Number of nodes and arcs over time.

(b) Average out-degree over time.

(c) Percentage of nodes in the giant SCC over time.

Fig. 3 Densification of G^t

TIME t	SNAPSHOT
1	Tue Jan 01 00:00:00 GMT 2013
2	Sun Feb 24 07:41:02 GMT 2013
3	Fri Apr 19 15:22:04 GMT 2013
4	Wed Jun 12 23:03:06 GMT 2013
5	Tue Aug 06 06:44:08 GMT 2013
6	Sun Sep 29 14:25:10 GMT 2013
7	Fri Nov 22 22:06:12 GMT 2013
8	Thu Jan 16 05:47:14 GMT 2014
9	Tue Mar 11 13:28:16 GMT 2014
10	Sun May 04 21:09:18 GMT 2014
11	Sat Jun 28 04:50:20 GMT 2014
12	Thu Aug 21 12:31:22 GMT 2014
13	Tue Oct 14 20:12:24 GMT 2014
14	Mon Dec 08 03:53:26 GMT 2014
15	Sat Jan 31 11:34:28 GMT 2015
16	Thu Mar 26 19:15:30 GMT 2015
17	Wed May 20 02:56:32 GMT 2015
18	Mon Jul 13 10:37:34 GMT 2015
19	Sat Sep 05 18:18:36 GMT 2015
20	Wed Dec 23 9:40:52 GMT 2015

AMOUNT a	THRESHOLD
1	0.000 001 BTC
2	0.000 01 BTC
3	0.000 1 BTC
4	0.001 BTC
5	0.01 BTC
6	0.1 BTC
7	1 BTC
8	10 BTC
9	100 BTC
10	1 000 BTC
11	10 000 BTC
12	100 000 BTC

Table 2 The time series we considered (upper part) for G^t and U^t and the different amounts for G_a and U_a (lower part).

the number of arcs in G^t increases faster than the number of nodes.

- Figure 3(c) shows the behaviour of the percentage of nodes in the giant strongly connected component of G^t with respect to the total number of vertices in G^t, i.e. $|V^t|$. This value quickly increases, meaning that, even though $|V^t|$ grows quite fast (Figure 3(a)), the number of nodes in the giant strongly connected component grows much faster making the network much more robust.

5.1.2 Distance Analysis

To perform a distance analysis we have computed diameter and average distance of the BITCOIN network. The distance $d(u, v)$ from a node u to a node v in a graph is the length of the shortest path from u to v. The diameter is defined as the $\max_{(u,v)\in V^t\times V^t} d(u, v)$, while the average distance is simply $\frac{1}{|V^t|^2}\sum_{(u,v)\in V^t\times V^t} d(u, v)$. In order to get more robust analysis, we have done these measurements considering the giant connected component of U^t for increasing values of t, where two users are connected whether they exchanged BITCOINs, ignoring the direction of the link. The average distance has been approximated using [31], while the diameter has been computed exactly using [32]. Note that, for a graph of n nodes and m edges, computing the average distance and the diameter requires $O(n \cdot m)$. The algorithm in [31] allows to approximate the average distance in $O(m)$ and the algorithm in [32] allows to compute exactly the diameter in $O(m)$ in practice in real world graphs.

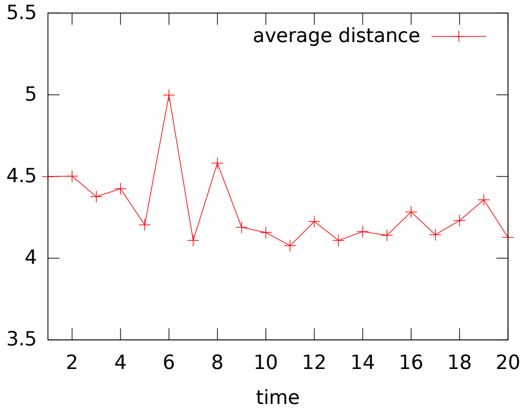


Fig. 4 Average distance of U^t for increasing values of t.

As observed for many other real world networks [33], we have seen that the diameter is not increasing. Surprisingly, the diameter is constant and very long (i.e. 2050) if compared to the diameter of many other real-world networks, like Facebook [34], where the number of vertices is much higher and the diameter is 41, and others (see `lasagne-unifi.sourceforge.net`). Our preliminary observations suggest that this peculiarity of the BITCOIN users graph is caused by the fact that transactions are also used to merge and split user funds and not just for payments, as explained in section 3. This is consistent with rare user cases observed in the past, obfuscating funds ownership using long fund splits chains (as noted for example in [5]). We plan to investigate extensively this topic in our future works.

Figure 4 shows the slow decrease over time of the low average distance. Note that this small average value compared to the high value of the diameter highlights that the nodes connected by long paths are present but few.

5.1.3 Degree Distribution

In Figure 5(a) and Figure 5(b) we show respectively the in-degree and the out-degree distributions of G^t with $t = 20$. As a further remark, we have seen that also the degree distribution of U^t, which for brevity is not shown here, follows a similar behaviour. It can be noticed that in both the plots (a) and (b) there are some outliers: there are some spikes close to $x = 1000$ in Figure 5(a), and to $x = 100$ in Figure 5(b). These spikes have been object of preliminary investigations in [17]. These have been shown to be related to particular topological patterns which are more likely due to transactions caused by unexpected users behavior rather than normal economic interaction.

All the distributions above follow a power law. In Figure 5(c) we show the power law exponent for increasing values of t for the in-degree distribution of G^t (red line), the out-degree distribution of G^t (green line), and the degree distribution of U^t (blue line). The power law exponent seems to be constant over time confirming the estimations done in [9].

5.1.4 Clustering Coefficient

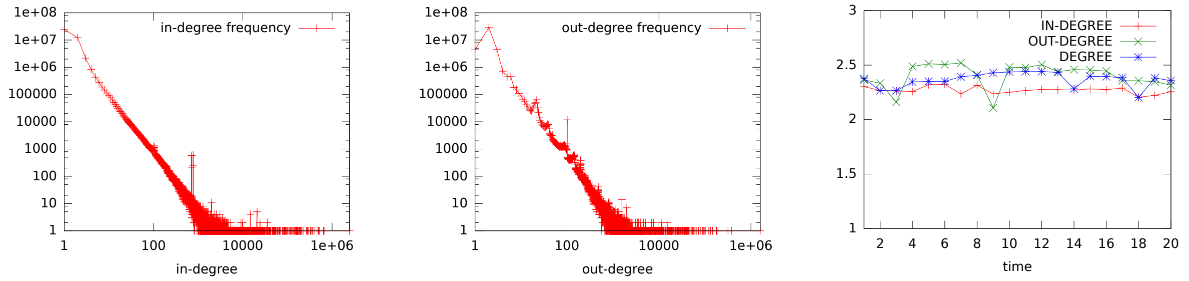
The clustering coefficient $c(v)$ of a node u in $U^t = (V^t, E^t)$ is defined as the percentage of the number of triangles involving v with respect to the possible number of triangles that could involve v. More formally, $c(v)$ is $\frac{2|\{(w,z)\in E^t:\ w,z\in N(v), w\neq z\}|}{|N(v)|\cdot(|N(v)|-1)} \cdot 100$, where $N(v)$ denotes the set of vertices which are neighbors of v in U^t. Since computing exactly the clustering coefficients of a graph with m edges is costly, i.e. the best known algorithm for computing triangles takes $O(m^{3/2})$ [35,36], we have used approximation algorithms based on min-sketches. In particular, we have used [37], setting the size of the sketches equal to 64. Using the latter algorithm, we have computed the average $c(v)$ for each graph U^t for increasing values of t, using the definition of global clustering coefficient by Watts and Strogatz [38]. In Figure 6(b) we report these values showing that the average clustering coefficient is basically constant over time. The order of magnitude of the correspondent values is similar to the one of the values reported in [39] for other complex networks. In Figure 6(a) we report also the distribution of $c(v)$ in U^t with $t = 20$. This distributions shows that the great majority of the vertices have clustering coefficient equal to 0. Among them, there are vertices composing long paths without shortcuts: these are the main responsible of the long diameter we have observed in Section 5.1.2.

Finally, we remark that, despite the high diameter, the slow decrease over time of the low average distance (shown in Figure 4), together with the relatively high clustering coefficient suggests a small world phenomenon.

5.2 Centrality Analysis

In this section we report the most central vertices in the BITCOIN network. These vertices correspond to the most active vertices in G^t or the ones that play a crucial role for the connectivity of the network (see [29], for more details about centrality measures). Given $G^t = (V^t, E^t, w^t)$, the centrality of u can be defined as follows.

- DEGREE: That is the degree of u in U^t, i.e. the number of nodes paying or payed by u. Central nodes are supposed to have many connections.
- IN-DEGREE: That is the in-degree of u in G^t, i.e. the number of nodes that payed u.
- OUT-DEGREE: That is the out-degree of u in G^t, that corresponds to the number of nodes which have been payed by u.
- HARMONIC: That is $\sum_{v\in V^t} \frac{1}{d(u,v)}$, where $d(u, v)$ is the distance between u and v in G^t. According to this measure, a node is central whether its distance from the others is small. This centrality is basically a variant of the closeness centrality which takes into account the disconnection of the network. A large value means high centrality [29].
- PAGE-RANK: this measure is popular because of its usage in Googles ranking algorithm [40]. The score of a node corresponds to the probability of visiting



(a) In-degree distribution of G^t with $t = 20$.

(b) Out-degree distribution of G^t with $t = 20$

(c) Power law exponent as the time goes by.

Fig. 5 Behaviour of Degree Distributions

	DEGREE		IN-DEGREE		OUT-DEGREE	
	IDENTITY	VALUE	IDENTITY	VALUE	IDENTITY	VALUE
1	Mt. Gox	3 386 581	Mt. Gox	2 452 049	Mt. Gox	1 591 319
2	LocalBitcoins.com	902 151	BTC-e.com1	683 875	2477299	381 426
3	2477299	848 176	LocalBitcoins.com	650 269	SatoshiDice.com	317 742
4	BTC-e.com1	740 402	AgoraMarket	636 969	LocalBitcoins.com	301 692
5	AgoraMarket	722 331	SilkRoadMarketplace	527 718	14782788	191 867
6	SilkRoadMarketplace	577 124	2477299	511 239	MoonBit.co.in	180 161
7	BitPay.com1	500 990	BitPay.com1	493 067	FaucetBOX.com	178 349
8	BTC-e.com2	492 219	BTC-e.com2	479 452	26638073	176 508
9	Cryptsy.com	461 111	BitPay.com2	394 447	Cryptsy.com	148 015
10	BitPay.com2	401 254	Cryptsy.com	361 298	23144512	146 624

Table 3 The top 10 central nodes according to degree centralities: degree in U^t, in-degree and out-degree in G^t for the last snapshot, i.e. $t = 20$. For the unknown identity we report the number of the identifier in our dataset.

	HARMONIC		PAGE-RANK		EIGENVECTOR	
	IDENTITY	VALUE	IDENTITY	VALUE	IDENTITY	VALUE
1	Mt. Gox	11 798 171	Mt. Gox	0.0183119	Mt. Gox	0.5369751
2	2477299	10 447 302	BTC-e.com1	0.0059199	SilkRoadMarketplace	0.0379121
3	LocalBitcoins.com	10 320 862	BTC-e.com2	0.0057736	BTC-e.com2	0.0303681
4	Cex.io	10 144 968	LocalBitcoins.com	0.0053508	BitPay.com1	0.0282049
5	FaucetBOX.com	10 136 604	SilkRoadMarketplace	0.0046309	Cex.io	0.0254740
6	26638073	10 071 881	AgoraMarket	0.0042383	SatoshiDice.com	0.0247234
7	MoonBit.co.in	10 065 853	BitPay.com1	0.0041264	LocalBitcoins.com	0.0242701
8	19860816	10 025 701	BitPay.com2	0.0036458	BTC-e.com1	0.0202184
9	Poloniex.com	9 976 766	SatoshiDice.com	0.0031628	BitPay.com2	0.0185086
10	Bittrex.com	9 926 321	2477299	0.0026843	AgoraMarket	0.0171055

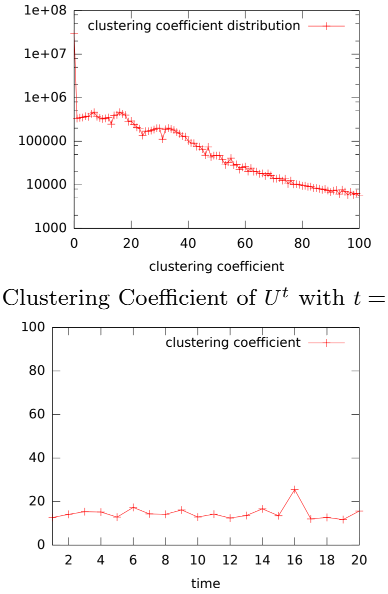
Table 4 The top 10 central nodes according to harmonic, page-rank, and eigenvector centrality in U^t, for the last snapshot, i.e. $t = 20$.

that node when performing a random walk in the graph or of visiting the node intentionally.

- EIGENVECTOR: this spectral measure uses the left dominant eigenvector of the plain adjacency matrix. Spectral measures, like also PAGE-RANK, in general are based on the assumption that a node is important if it is linked to by other important nodes. In this case, the centrality of a node is the result of a convergent iterative process which replaces the score of a node with the sum of the scores of its predecessors [41].

In Table 3, we report the top-k users according to the above measures based on degree, with $k = 10$. On the other hand, Table 4 shows the top-k with $k = 10$ according to the other measures. The HARMONIC centrality has been computed using [42] while the PAGE-RANK and EIGENVECTOR centrality have been approximated using the software provided by [29].

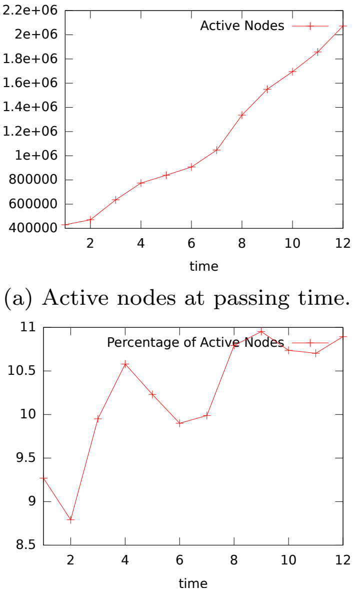
We remark there is a big correlations between degree centralities and the other measures. As expected, the selected central vertices are almost all very popular. Mt. Gox (the most famous BITCOIN exchange before its failure at the beginning of 2014) is the most central according to all the measures not only considering its local connectivity, i.e. the degrees, but also for the connectivity of the whole network. The same applies to LocalBit-



(a) Clustering Coefficient of U^t with $t = 20$.

(b) Average Clustering Coefficient at passing time.

Fig. 6 Behaviour of Clustering Coefficient



(a) Active nodes at passing time.

(b) Percentage of active nodes at passing time.

Fig. 7 Active nodes

coins.com (the largest P2P BITCOIN trading platform). Interestingly, it seems difficult to identify some users, as for instance the one corresponding to vertex 2477299, that seems to be very central.

The degree centralities consider the size of the local community of each node, so that the vertices shown in Table 3 are the ones which have participated to (respectively, received or send) more transactions. Some popular nodes are equally central both considering incoming and outgoing transactions. On the other hand, some nodes, like MoonBit.co.in or FaucetBOX.com, are more likely paying than receiving with respect to others, while some others, like AgoraMarket and SilkRoadMarketplace are more likely receiving than paying. This is consistent with the nodes category since the first two are faucets (giving away tiny amounts of BTC for free), while the second ones are marketplaces accepting payments in BTC.

The HARMONIC rank highlights which nodes are closer to all the others from a global point of view considering shortest paths, while PAGE-RANK and EIGENVECTOR consider arbitrary paths. The nodes in the first column of Table 4 are nodes basically making shortcuts between different part of the network. It is interesting to note that the nodes ranked from 6 to 10 in the first column do not appear in the rightmost columns. We argue that considering shortest paths rather than arbitrary paths significantly changes the centrality of a node probably because of the several alternative paths that are present in the network. On the other hand, as expected, the rightmost columns of Table 4, both concerning spectral measures, are quite similar. These nodes are the ones more likely to meet when performing a random walk, which can obviously take place along arbitrary paths. However, we notice a particular similarity with DEGREE and IN-DEGREE centrality.

5.3 Active Users Over Time

We've shown in Section 4.3 that transactions are divided among standard and non-standard, where non-standard transactions are not relayed on the network nor included in blocks by compliant nodes. We remark that non-standard transactions can still be valid so that they are accepted if they appear in the blockchain. We've shown how scripts can make transactions non-standard but other factors have a role as well. The rules defining standard transactions are dictated and updated over time by the official client implementation. Regarding the possible amount spendable by transactions, the standardization rules introduced the concept of dust [43]: *an amount* a *is called* dust *if creating a transaction to spend* a *would cost more in recommended fees than* a. The rationale behind the dust definition is that the amount transferred in the transaction is economically unreasonable to be spent.

So we consider the *dust limit* as the threshold to consider a balance negligible. This is because in the majority compliant network would be almost impossible for a user with a current balance lower than the dust limit to create a transaction that will be actually included in the blockchain. It is worth remarking that the value stored by such a user is not lost. The user cannot spend that value alone in a transaction but he can cooperate with others to collect the dust to obtain transactions with total value high enough to be considered standard. There exist protocol to do so (called *dust collectors*) to collect the dust and give it directly to the miners. What we want to state is that the economic power of a below dust account is irrelevant, in the sense that it cannot be spent by the user alone to benefit him (as regular value would instead do). As a result, in practice we can consider all users with a current balance value lower than the dust limit equivalent to users with a zero balance. The actual value of the dust limit has varied over time and has been dependent of other parameters (such as the *minimum relay transaction fee* value) that have varied over time as well or have been left under discretion of nodes owners. The current standard can be seen in [43]. For the sake of simplicity, in order to set a fixed dust limit over the entire time span of our analysis, we've decided to choose as dust limit the most conservative "official" value (i.e., the lowest) adopted during the same time span, which is 0.00000546 BTC.

We consider each cluster as *active* at a give time if its current balance is greater than the dust limit.

In Figure 7(a), we show how the number of active nodes changes over time. The increase seems to be slightly more than linear. For $t = 20$, we think the estimation does not follow the trend because of the clustering phase, which is less effective to recognize very recent users, as shown in Section 4. This plot should be compared with the one in Figure 3(a), which showed the increase of nodes over time. In Figure 7(b), we show the percentage of active nodes with respect to the total number of nodes in the network over time. This ratio slightly increases, passing from 9% to 12%, suggesting that a greater portion of the network is remaining active with the passing of time.

5.4 Rich get Richer and Concentration of Richness

This section is devoted to verify the *rich get richer* hypothesis and measure the concentration of richness, both on the balance and the connectivity point of view. Indeed, we consider two different definitions of richness: we say that a user is rich whether its balance or its number of incoming transactions is high with respect to the other users in the network.

We remark that our observations are not an artifact of the protocol but rely on truly economical and connectivity properties. Indeed, they depend only on the number of transactions and on their amount, that only relate to users behaviour.

Differently from the analysis done in the preliminary version [12] of this paper, we have restricted our studies to active users only. We think that this new restriction strengthens our results. Indeed, from a balance point of view, considering only active users does not change the richness ordering. It only ignores the long tail of lower clusters with zero or negligible current balances, hence influencing the overall average but not the ordering. In such a context, we show the rich get richer phenomenon and an high concentration of richness even neglecting very poor users, i.e. non-active users. Hence, if we compare our new results with the ones in [12] we can see how despite restricting just to active users, the trend observable in the graphs is the same.

Thus, in the following, we will consider G^t as the graph without non-active nodes.

We aim to verify the following properties for both the definitions.

Property 2

1. The richest users at time t are richer than the richest users at time $t' < t$.
2. The richest users at a certain time t tend to remain the richest at time $t' > t$.
3. The richness gets more concentrated with the progression of time.

Given the weighted multigraph $G^t = (V^t, E^t, w^t)$ (without non-active nodes) and ϕ^t for each $v \in V^t$, we formally define the richness of a node $u \in V^t$ as its balance $b^t(u)$ or its number of incoming transactions $d_t(u)$ as follows.

$$b^t(u) = \sum_{(v,u)\in E^t} w(v,u) - \sum_{(u,v)\in E^t} w(u,v) - \phi^t(u) + \beta^t(u) \quad (1)$$

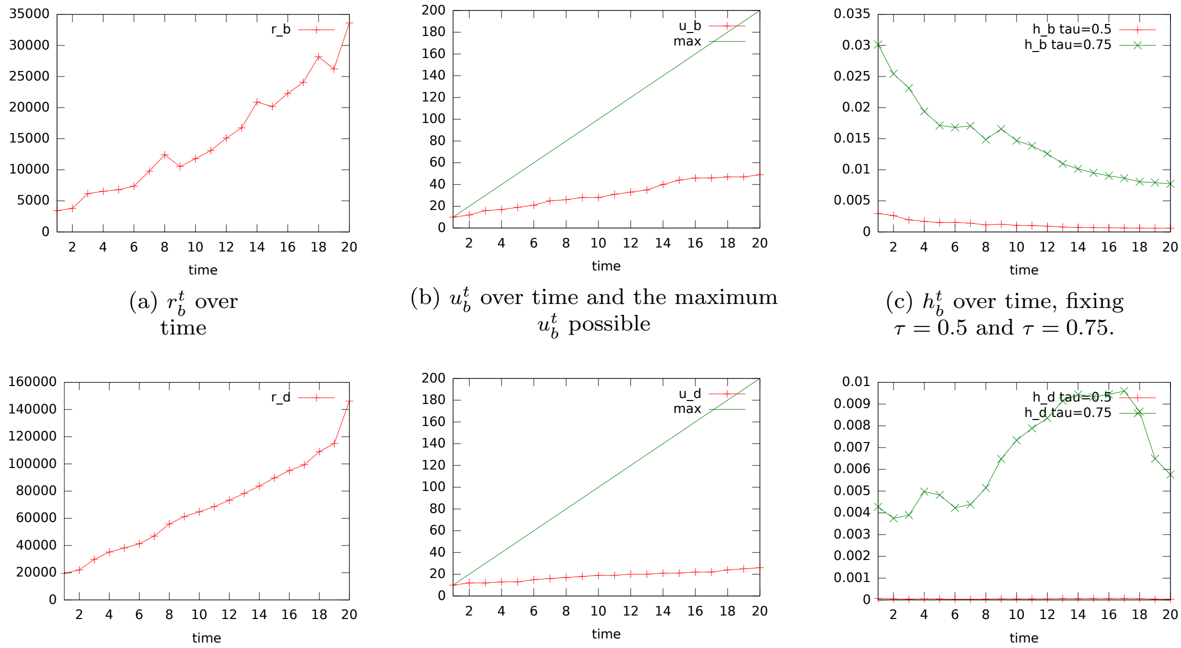
$$d^t(u) = |\{(v,u) : (v,u) \in E^t\}| \quad (2)$$

Observe that the measure $b^t(u)$ is taking into account also the fees payed by user u as $\phi^t(u)$. Moreover, $\beta^t(u)$ is the increase of the balance of u up to time t (eventually 0) coming from special transactions called *coinbase* whose aim is minting new coins (as explained in Section 3).

Given an integer k, we denote respectively as B^t_k and D^t_k the k nodes having maximum b^t and d^t. Table 5 shows the sets D^t_k (IN-TRANS columns) and B^t_k (BALANCE columns) for $t = 20$ and $k = 10$. The identity of these nodes is the name or the identifier of the node in our dataset in the case we do not know its name. As far as we know, many of the BALANCE column identifiers correspond to BITCOIN accumulator addresses (single addresses that received huge amounts of BTCs over time without spending them).

RANK	IN-TRANS		BALANCE	
	IDENTITY	VALUE	IDENTITY	VALUE
1	Mt. Gox	22 399 043	39912924	169 731
2	SatoshiDice.com	12 879 343	23638585	157 997
3	LuckyB.it	3 620 428	542746	87 111
4	BitZillions.com	1 651 456	Mt. Gox	81 492
5	BTC-e.com1	1 430 992	177808	79 957
6	LocalBitcoins.com	1 356 029	6128144	69 370
7	Xapo.com	1 264 648	10597666	66 650
8	AgoraMarket	1 184 011	10467915	66 612
9	BetcoinDice.tm	1 159 024	10484095	66 583
10	2477299	1 101 024	10475912	66 452

Table 5 The top 10 richest nodes in G^t with $t = 20$.



(a) r_b^t over time

(b) u_b^t over time and the maximum u_b^t possible

(c) h_b^t over time, fixing $\tau = 0.5$ and $\tau = 0.75$.

(d) r_d^t over time

(e) u_d^t over time and the maximum u_d^t possible

(f) h_d^t over time, fixing $\tau = 0.5$ and $\tau = 0.75$.

Fig. 8 Verifying Property 2.1, 2.2, and 2.3 for the definitions of richness given by Equation 1 (upper part, respectively (a), (b), and (c)) and Equation 2 (lower part, respectively (d), (e), and (f)).

Note that, the measure $d^t(u)$ corresponds to the in-degree of u in the multigraph, i.e. the number of transaction outputs paying u. This is different from the in-degree considered in Section 5.2 which corresponds to the degree in the simple graph, i.e. the number of users paying u.

A definition similar to $d^t(u)$ can be done considering the transactions outgoing from u, obtaining similar results. However, the number of transactions outgoing from a user can be arbitrarily increased by the user (performing transactions with small amounts) to increase its connectivity. We argue that our definition of $d^t(u)$ is more robust for modelling economic importance.

5.4.1 Verifying Property 2.1

To verify that the richest users in G^t are richer than the richest in $G^{t'}$ with $t' < t$, we study the following quantities over time.

$$r_b^t = \frac{\sum_{u \in B_k^t} b^t(u)/k}{\sum_{u \in V^t} b^t(u)/|V^t|}, \qquad r_d^t = \frac{\sum_{u \in D_k^t} d^t(u)/k}{\sum_{u \in V^t} d^t(u)/|V^t|}$$

Basically, r_b^t (resp. r_d^t) is the ratio between the average balance (resp. incoming transactions count) of the top-k richest users with respect to the average balance (resp. incoming transactions count) of all the users in the network. As this ratio gets higher, the disparity of the richest nodes with respect to all the others gets bigger.

Figure 8(a) clearly shows that r_b^t increases over time, meaning that this disparity increases. On the other hand, Figure 8(d) shows that the same applies to r_d^t.

5.4.2 Verifying Property 2.2

In order to test the diversity of the richest node sets, i.e. B_k^t (resp. D_K^t), varying t, we study the following quantities.

$$u_b^t = |\bigcup_{i=1}^{t} B_k^i| \qquad u_d^t = |\bigcup_{i=1}^{t} D_k^i|$$

Since $|B_k^t| = k$ (resp. $|D_k^t| = k$), in the case the richest node sets does not change for each time i with $1 \leq i \leq t$, we have $u_b^t = k$ (resp. $u_d^t = k$). On the other hand, if the sets B_k^i (resp. D_k^i) change completely for each i then we have $u_b^t = t \cdot k$ (resp. $u_d^t = t \cdot k$).

Figure 8(b) and (e) show the behavior of u_b^t and u_d^t over time with respect to the expected behaviour if sets would change (green line). Both the sets B_k^t and D_k^t are very stable. Fixing $t = 20$ and $k = 10$, u_b^t, the set of all the k-richest nodes in the *history* of BITCOIN, is less than 50 instead of 200. This stability seems to be even more evident in the case of u_d^t, where $|\bigcup_{i=1}^{t} D_k^i|$ is smaller than 30 instead of 200.

5.4.3 Verifying Property 2.3

To measure the concentration of richness in $G^t = (V^t, E^t, w^t)$, fixing a threshold τ, we considered the following measures.

$$h_b^t = min\left\{k \; : \; \frac{\sum_{u \in B_k^t} b^t(u)}{\sum_{u \in V^t} b^t(u)} > \tau\right\} /|V^t|$$

$$h_d^t = min\left\{k \; : \; \frac{\sum_{u \in D_k^t} d^t(u)}{\sum_{u \in V^t} d^t(u)} > \tau\right\} /|V^t|$$

As an example, consider $\tau = 0.75$:

– h_b^t is the minimum (normalized) k such that B_k^t owns the 75% of the richness, in terms of balance, of the whole network;
– h_d^t is the minimum (normalized) k such that D_k^t owns the 75% of incoming connections of the network.

A small value for h_b^t (or h_d^t) means that the richness is concentrated in few users. The increase of concentration of richness can be witnessed checking whether h_b^t (and h_d^t) decreases over time.

In Figure 8(c) and (f), we report the behaviour of both h_b^t and h_d^t for increasing values of time t in our time series setting $\tau = 0.5$ and $\tau = 0.75$. Figure 8(c) refers to h_b^t and clearly decreases over time showing that the balance becomes more concentrated as the time passes; this is more evident especially considering an higher value of τ, i.e. $\tau = 0.75$.

On the other hand, the results for h_d^t, reported in Figure 8(f), shows that connectivity does satisfy Property 2.3. Indeed, contrarily to h_b^t, h_d^t seems to increase over time (except for the last data points). We argue that, for the densification process shown in Section 5.1.1, the increase of arcs is too big to be suitably absorbed from a same percentage of nodes.

5.4.4 Gini Coefficient

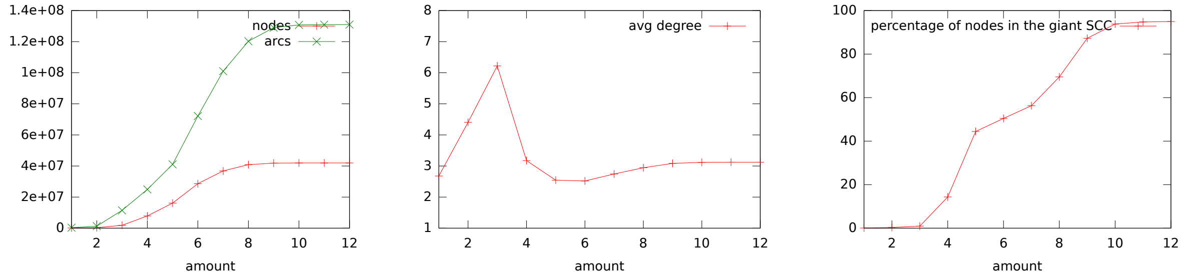
As a further measure to support our concentration studies we have also considered the Gini coefficient, a well established inequality measure defined as follows: given x_i indexed in non-decreasing order, the Gini coefficient is

$$g = \frac{2\sum_{i=1}^{n} i x_i}{n \sum_{i=1}^{n} x_i} - \frac{n+1}{n}$$

In our case, x_i is the richness of the i-th node u in the network G^t, respectively defined as $b^t(u)$ and $d^t(u)$. In Figure 10, we show how the Gini coefficient changes over time, considering both $b^t(u)$ and $d^t(u)$, respectively in Figure 10(a) and Figure 10(b). First of all, notice that the values are very high in general and relatively close to the maximum possible, i.e. 1. In the first case, when richness is defined as balance, the Gini coefficient clearly increases over time, confirming the fact that the balance based definition richness satisfies Property 2. On the other hand, in the case of Figure 10(b), when richness is defined in terms of connectivity, the Gini coefficient decreases over time. This fact is consistent with our findings in the previous sections, since we have seen that connectivity richness does not satisfy Property 2.3.

5.5 Further Analysis for Different Transaction Amounts

In this section we show the growth of graph G_a for increasing values of a. Recall that G_a is the graph G built in Section 4 induced by transactions whose amount is smaller than a, where the correspondence between a and real BTCs is provided by Table 2.

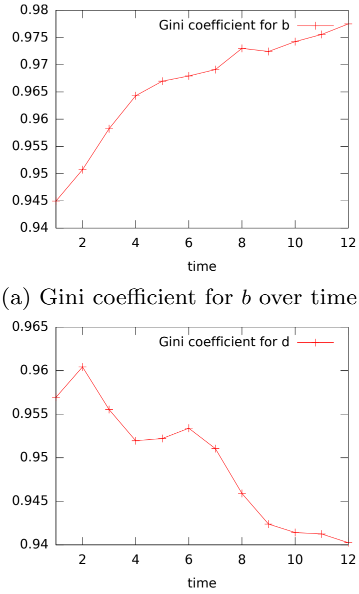


(a) Number of nodes and arcs of G_a for increasing amounts a.

(b) Average degree in G_a for increasing amounts a.

(c) Percentage of nodes in the giant SCC of G_a for increasing amounts a.

Fig. 9 Densification of G_a for increasing amounts a (see also Table 2)



(a) Gini coefficient for b over time.

(b) Gini coefficient for d over time.

Fig. 10 Gini coefficient for both the definitions of richness.

Figure 9(a) shows the increase of number of nodes and arcs in G_a for increasing a. We can see that the bigger increase of nodes is when transactions of amount between 3 and 7 are introduced (see the angles of red and green lines). From $a \geq 8$, the number of nodes and arcs is stable, meaning that not many transactions have an amount greater than 1 BTC. Figure 9(b) shows the average degree for increasing a. There is a spike for $a = 3$ meaning that the maximum relative increase of edges with respect to the nodes is when introducing transaction with amount between 0.00001 BTC and 0.0001 BTC. This is due to the fact that nodes doing this kind of transactions often perform even smaller transactions. Looking at Figure 9(c), we can see that these smaller transactions are not well connected meaning that these take place in independent parts of the network: some parts of the network are giving, some other parts are receiving, but rarely they are exchanging. This suggests that no well connected micro-economy is present, but all the transactions up to $a = 8$ are needed to make the great majority of the network strongly connected. Interestingly, we can see a similar shape between Figure 9(c) and the green line in Figure 9(a).

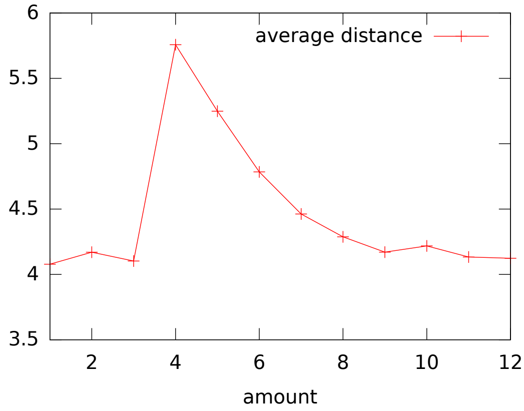


Fig. 11 Average distance of U_a for increasing values of a.

Figure 11 shows the average distance in U_a, the undirected version of G_a. The starting increase for $a < 4$ is due to the fact that we are merging independent parts of the network, creating new paths (this is consistent with Figure 9(c)). The decrease for $a > 4$ is due to the fact that bigger transactions are creating shortcuts for relationships already existing.

For the sake of completeness, we mention that we observed a stable power law exponent for increasing values of a for all the degree distributions, similarly to Figure 5. In particular, the exponent is constantly about 2.3 for the out-degree distribution in G_a and the degree distribution in U_a; it is 2.2 for the in-degree distribution in G_a.

6 Conclusions

This paper introduces a scalable clustering algorithm able to support the construction of the BITCOIN users graph and a set of analyses, applied on the users graph produced by this algorithm, which allows to uncover several properties of the BITCOIN network. The users graph is derived from the transactions present in the blockchain until December 2015, this includes about 100 millions of multi input, multi output transactions. The paper extends our previous work [12] in several directions. The analysis of the rich get richer property has been refined by considering only users with a balance larger than a threshold. To further validate our analysis, we have also computed the Gini coefficient of these users. The results confirm the "rich get richer" property and the existence of central nodes acting as hubs between different parts of the network. We have also verified that the clustering coefficient of the users graph is the same of order of magnitude of other complex networks [44], which highlights the complex nature of the BITCOIN network. Finally, we have computed further centrality measures to detect the most central nodes in the network. The results confirm the presence of a set of nodes which are central according to several definition of centrality and these nodes correspond to very popular nodes. Moreover, we have seen that considering shortest paths or arbitrary paths to define centrality of nodes can significantly change the ranking. We conjecture that this is mainly due to the presence of many alternative paths and we expect to verify this conjecture for instance computing the hyperbolicity of the network [45].

We plan to extend our analysis in several directions. Since our experiments point out some anomalous behaviours in the node degree distribution and a very large diameter of the users graph, we plan to investigate the actual users behaviors leading to these peculiar properties. A preliminary result has been presented in [17], where we conjecture that these topological patterns are due to unexpected users behaviors, not strictly related to normal economic interaction. We plan to further investigate this issue in our future work and to find economical reasons for our observations. As a further future work, it would be interesting to find graph structures whose number of occurrences correlates with price trends.[aggiunte e modificate ultime frasi]$_{A}$

Acknowledgment

The authors would like to thank Dr. Christian Decker and Prof. Roger Wattenhofer of the Distributed Computing Group, ETH Zurich for providing us the blockchain in Protocol Buffers format.

This work was supported by PRA, Progetto di Ricerca di Ateneo, "Big Data, Social Mining and Risk Management", University of Pisa.

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. W. M. K. Komurov, MH.Gunes, "Fine-scale dissection of functional protein network organization by statistical network analysis," *PLoS ONE*, vol. 4, no. 6, 2009.
2. D. Cheung and M. H. Gunes, "A complex network analysis of the united states air transportation," in *Proceedings IEEE/ACM ASONAM, Washington, DC*, 2012, pp. 699–701.
3. M. G. H. Kardes, A. Sevincer and M. Yuksel, in *Six degrees of separation among US researchers Proceedings of IEEE/ACM SONAM*, 2012, pp. 654–659.
4. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
5. D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, 2013, pp. 6–24.
6. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013*, 2013, pp. 127–140.
7. M. Ober, S. Katzenbeisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future Internet*, vol. 5, no. 2, pp. 237–250, 2013.
8. E. Androulaki, G. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, 2013, pp. 34–51.
9. D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, "Do the rich get richer? an empirical analysis of the bitcoin transaction network," *PloS one*, vol. 9, no. 2, p. e86197, 2014.
10. M. Lischke and B. Fabian, "Analyzing the bitcoin network: The first four years," *Future Internet*, vol. 8, no. 1, 2016.
11. "Block chain info charts." [Online]. Available: https://blockchain.info/charts/
12. D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the bitcoin blockchain: an analysis of the full users graph," in *IEEE DSAA 2016, 3rd IEEE International Conference on Data Science and Advanced Analytics, Montreal, October*, 2016.
13. R. Fergal and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Proceeding of 2011 PASSAT/SocialCom 2011*. IEEE, 2011, pp. 1318–1326.
14. T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *Computer Security-ESORICS 2014*. Springer, 2014, pp. 345–364.

15. M. Harrigan and C. Fretter, “The unreasonable effectiveness of address clustering,” in *13th IEEE International Conference on Advanced and Trusted Computing (ATC16)*, 2016.
16. M. K. Popuri and M. H. Gunes, “empirical analysis of crypto currencies,” in *7th Workshop on Complex Networks (CompleNet), Dijon, France, Mar 23-25*, 2016.
17. D. D. F. Maesa, A. Marino, and L. Ricci, “An analysis of the bitcoin users graph: inferring unusual behaviours,” in *Proceedings of the 5-th International Workshop on Complex Networks and their Applications, Milan*, 2016.
18. U. NIST, “Descriptions of sha-256, sha-384 and sha-512,” 2001.
19. B. Preneel, A. Bosselaers, and H. Dobbertin, “The cryptographic hash function ripemd-160,” 1997.
20. D. Johnson, A. Menezes, and S. Vanstone, “The elliptic curve digital signature algorithm (ecdsa),” *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.
21. R. C. Merkle, “A digital signature based on a conventional encryption function,” in *Advances in Cryptology - CRYPTO ’87, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, 1987, pp. 369–378.
22. C. Dwork and M. Naor, “Pricing via processing or combatting junk mail,” in *Advances in CryptologyCRYPTO92*. Springer, 1992, pp. 139–147.
23. J. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: Analysis and applications,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 281–310.
24. A. Miller and J. J. LaViola Jr, “Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin,” *Available on line: http://nakamotoinstitute.org/research/anonymous-byzantine-consensus*, 2014.
25. A. Back *et al.*, “Hashcash-a denial of service counter-measure,” 2002.
26. “Protocolbuffers.” [Online]. Available: https://developers.google.com/protocol-buffers/
27. “Block chain info tags.” [Online]. Available: https://blockchain.info/tags
28. “Wallet explorer.” [Online]. Available: https://www.walletexplorer.com/
29. P. Boldi and S. Vigna, “Axioms for centrality,” *Internet Mathematics*, vol. 10, no. 3-4, pp. 222–262, 2014. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/15427951.2013.865686
30. ——, “The webgraph framework i: Compression techniques,” in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW ’04. ACM, 2004, pp. 595–602.
31. P. Boldi, M. Rosa, and S. Vigna, “Hyperanf: Approximating the neighbourhood function of very large graphs on a budget,” in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 625–634.
32. M. Borassi, P. Crescenzi, M. Habib, W. A. Kosters, A. Marino, and F. W. Takes, “On the solvability of the six degrees of kevin bacon game - A faster graph diameter and radius computation method,” in *Fun with Algorithms - 7th International Conference, FUN 2014, Lipari Island, Sicily, Italy, July 1-3, 2014. Proceedings*, 2014, pp. 52–63.
33. J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: densification laws, shrinking diameters and possible explanations,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.
34. L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna, “Four degrees of separation,” in *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 2012, pp. 33–42.
35. A. Itai and M. Rodeh, “Finding a minimum circuit in a graph,” *SIAM Journal on Computing*, vol. 7, no. 4, pp. 413–423, 1978.
36. N. Chiba and T. Nishizeki, “Arboricity and subgraph listing algorithms,” *SIAM Journal on Computing*, vol. 14, no. 1, pp. 210–223, 1985.
37. L. Becchetti, P. Boldi, C. Castillo, and A. Gionis, “Efficient semi-streaming algorithms for local triangle counting in massive graphs,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 16–24.
38. D. J. Watts and S. H. Strogatz, “Collective dynamics of small-worldnetworks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.
39. R. Albert and A. lszl Barabsi, “Statistical mechanics of complex networks,” *Rev. Mod. Phys*, 2002.
40. L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: bringing order to the web.” 1999.
41. A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Society for Industrial and Applied Mathematics, 1994. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611971262
42. P. Boldi and S. Vigna, “In-core computation of geometric centralities with hyperball: A hundred billion nodes and beyond,” in *Proceedings of the 13th IEEE International Conference on Data Mining Workshops (ICDM)*, 2013, pp. 621–628.
43. “Current standard for dust limit.” [Online]. Available: https://github.com/bitcoin/bitcoin/blob/v0.10.0rc3/src/primitives/transaction.h\#L137
44. M. E. Newman, “The structure and function of complex networks,” *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.
45. M. Borassi, D. Coudert, P. Crescenzi, and A. Marino, “On computing the hyperbolicity of real-world graphs,” in *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, 2015, pp. 215–226.