

区块链矩阵

段志朝

mistakenine@gmail.com

摘要：在区块链系统中，人们使用区块链这种数据模型实现数据的存储，但是区块链这一数据模型天生存在着低效和安全冲突的问题。一个自然的想法是采取多条区块链成为一个整体——一种新的数据模型（例如 BTC 中存在多条区块链一起存储 BTC，多条链中的交易互相引用），来解决在同等安全性下提高效率的问题。但是采取多条区块链并行的数据模型在存储数据时会产生相当复杂的问题从而导致根本无法形成一致性，这一问题至今没有一个解决方案。本文的目的在于给出上述问题一个解决方案，我们把多条区块链看成一个整体的数据模型称为区块链矩阵。

1、区块链介绍

1.1、区块链系统中重要的三个定义

为了避免现阶段对区块链没有明确的学术定义导致的探讨问题出现分歧，本文将首先按照作者的想法整理一些基本的概念。

普遍描述的区块链是一个复杂概念，因为它包含了以下几种概念在一起：

- 一种新的区块-链式数据结构；
- 验证和存储数据的过程；
- 分布式节点的共识算法保证新的区块生成和验证；
- 密码学的方式保证数据传输的访问的安全；
- 自动化脚本代码组成的智能合约来编程和操纵数据；
- 使用某种协议或算法通过寻找最长链来保证一致性。

基于本文探讨的内容，我们可以从上述这些概念中做出三个定义：

(1)、区跨链——一种数据结构

区块链本身就是一种数据结构，是一种比图、树更复杂的数据结构，区块链中的每个区块至少包含区块头部和区块数据两部分内容；

区块数据中存储着的数据，我们称为交易。交易可能存在两种情况：

- 区块链系统中规定的生成区块时给予发现区块者而生成的交易，我们称之为初始交易。初始交易是在系统执行之前就定义好的，不可变更的。
- 区块链中除初始交易之外的任意一个交易 Tx ，都是建立在其他一系列交易之上的，记这些交易的集合为 T ，我们称 T 为 Tx 的依赖交易集合，对于 $\forall Tx' \in T$ ，我们称 Tx 依赖于 Tx' ，记为 $Tx \gg Tx'$ 。

在区块链系统中，强制要求区块链上的一个交易 Tx 最多被依赖一次。如果有存在两个交易 Tx_1, Tx_2 ，均依赖于某个交易 Tx ，我们称 Tx_1, Tx_2 发生依赖冲突，记为 $Tx_1 <> Tx_2$ ，这在区块链系统中是不允许的。

一个交易必须符合交易的规范，交易的规范是区块链系统中统一规定的，不同的区块链系统中，规范可能不同，但是上述条件都是充分的。

(2)、共识算法 P

区块链系统中需要有一个在给定系统中唯一存在的共识算法 P ，该算法是为了生成新的区块和验证区块而存在的，决定各区块如何通过区块头部链接在一起形成区块链；

(3)、区块链一致性算法 $P_{\text{区块链}}$

跨链系统中分布式节点上的数据一致性，是由区块链一致性算法 $P_{\text{区块链}}$ 保证的，一个区块链算法 $P_{\text{区块链}}$ 是要解决如下的问题：

问题：在容错分布式系统中，在下列 1 个假设的前提下：求一个算法来保证系统中数据的一致性。

假设：给定共识算法 P ，依据共识算法 P 对算力的定义，系统中拜占庭节点持有的算力不能超过 51%。

一个区块链算法 $P_{\text{区块链}}$ 还依赖于以下两个给定的条件：

条件 1：给定区块链中区块的数据结构；

条件 2：给定一种交易的规范，能够检查交易的合法性，规范必须要求交易和交易之间不能依赖冲突；

在中本聪的文章里，给出了一个区块链一致性算法 $P_{\text{区块链}}$ 的解：

1、新的交易发生时，向所有的对等网络中的节点发起广播；

2、每个节点通过广播收集这些交易，按照次序将交易排序，检查交易是否符合规范，剔除不符合规范的交易然后把这些交易打包进新的区块；

3、每个节点根据共识算法 P ，在算法 P 的规范下去争取生成新的区块，我们称这个过程为争夺记账权；

4、当一个节点发现一个新的区块时，它向所有对等网络中的节点发起广播；

5、节点收到新区块的广播时，首先使用共识算法 P 对新区块发起验证，其次检验区块中的交易是否符合交易规范，如果两点均符合，就接受这个区块。如果使用共识算法 P 对新区块做验证时，可能会发现自己的区块链不够长，即有一些区块的广播没有被收到，要先接受那些没有被收到的区块；

6、节点接受的区块接入已验证的区块链，形成新的区块链，并重新开始依据共识算法 P 继续寻找新的区块；

7、所有节点认为最长的区块链是正确的区块链，并在最长的链上依据共识算法 P 寻找新的区块。

该算法的输出，是整个系统的一致性。

1.2、区块链系统中的瓶颈

区块链系统中，区块的数据结构和交易规范如果一定，区块链一致性算法 $P_{\text{区块链}}$ 是几乎不会有大的改动的，唯一能改动的只有共识算法 P ，现阶段知名的共识算法有 POW、POS、DPOS、PBFT 等。

由于区块链这一数据结构本身的制约，存在以下问题：

- 区块的大小变大，就会增加区块的传输时间，从而增加区块的验证

时间，使得安全性降低，区块大小变小，就会减少能存储的交易数量；

- 改变共识算法使得生成区块时间缩短，就会使得安全性降低，改变共识算法使得生成区块时间增加，就会让单位时间内可发生的交易量降低。

也就是说，在区块链系统中，提高效率代表着安全性降低。但是，我们很多时候都不能够牺牲安全性，尤其是在公链的建设上。所以，我们迫切的希望寻找到新的数据结构和新的一致性算法，能够在相同或相近级别安全性的前提下，极大地提高系统效率。

自然可以想到，我们希望拥有一个并行的区块链形成新的数据结构，和基于这个数据结构下给出一个新的一致性算法。但是探讨这一问题，首先我们必须明确安全性的度量。

1.3、安全性界定

安全性问题是个抽象的东西，我们必须给出一个能量化的方案去考量。在比特币系统中，如果一个交易被记录在某一个区块B中，那么人们都认为，在区块B后面再加上 5 个区块即可认为是一个已确认的交易，相当于 6 个区块的确认，而现在普遍的比特币钱包采取的是 3 个区块的确认。而基于 DPOS 的算法下的 BTS 系统中，人们认为需要再加上 14 个区块即可认为是一个已确认的交易，即 15 个区块的确认。。

在区块链交易系统中，是没有 100% 交易确认的，只能是这个确认概率无限接近于 1，也就是说，依然存在极低概率，在大家认为交易已经被确认的时候，发生攻击事件导致交易完全失败。但是在现实人们使用中，当大家公认的一个交易被确认的时候，如果发生攻击导致交易失败，我们就认为是安全性发生了问题。对此，我们给出如下的安全性量化标准：

安全性量化标准：对于一条区块链 A ，在共识算法 P 下，必然存在一个最小的正整数 c ，如果区块 B 上再接受到 $c - 1$ 个区块，记录在 B 中的所有交易的确定性不低于给定的概率 p 。其中 c 被认为是共识算法 P 在给定的概率 p 下的一个安全性量化标准。

记包含区块 B 及 B 以上的区块这一段区块链为 $up(B)$ 。

毫无疑问，这个量化标准是合理的，因为对于区块链矩阵来说，如果我们认为交易必须达到某个概率 p 才能被确认， p 在共识算法 P 给定的情况下， c 能被唯一找到。而且明显的 p 越接近 1， c 的值越大。

对于不同的共识算法 P ，如果所需的 p 一定，出块时间越短的，一般情况下，所需要的 c 越大，特殊情况只在共识算法 P 对算力定义不同时出现。从本质上看，区块链一致性算法完全是靠抑制区块的生成速度，让获得区块困难增大，使得在一段时间内，必然出现一条最长链达成的。

对于给定的共识算法 P ，如果所需的 p 一定，区块大小越大的， c 越大。

易知，下面的定理是一个安全性量化标准的必要条件：

定理 1：如果 c 是共识算法 P 在给定的概率 p 下的一个安全性量化标准，设 q 为在单条区块链中两个区块同时生成的概率，则满足： $q^c < 1 - p$ 。

2、思路概述和问题提出

2.1、思路概述

因为文章过长，我们先对并行多条区块链的想法做个基本的思路分析。

自然的我们希望做如下几件事情，让算法得以成立：

(1)、把并行的区块链看成一个整体，当成一种全新的数据结构来看，

我们称之为区块链矩阵，区块中的交易因为依赖性在不同的链上形成一个极端复杂的图，不同链上的区块中初始交易生成的货币都是相同的；

(2)、区块链矩阵中，不同区块链上生成区块的过程是相互独立的，所以效率会成倍上升；

(3)、每一个新的交易，在生成的时候就能通过某种规则知道应该被哪条区块链所记录，不会因为随机性产生混乱；

(4)、区块链矩阵这一整体数据结构拥有一个“长度”，让每个节点能辨别哪个是正确的区块链矩阵，当区块链矩阵出现分叉的时候，能根据这个“长度”规则自然而然的知道应该如何取舍“分叉”。

毫无疑问，如果我们做到以上的四点，一个新的算法就成立了。

这里面临一个难题：区块链矩阵的分叉问题，分叉问题必然是在一定的时间内，同时的接受到了互相冲突的区块导致的，在单条区块链上，是因为拥有同一个父区块才会导致这一冲突，但是在区块链矩阵中，两条不相关的区块链中的交易引用发生冲突，导致“双花”问题也会产生不同区块链上的区块冲突。而因为区块链条数的增加，区块链矩阵又希望区块的生成是独立的，所以如果单条区块链上同时生成两个区块的概率为 q ，那么多条区块链上同时生成两个区块的概率就是 q 的倍数。如果不能立刻降低这个概率，则区块链矩阵的安全性还是会降低。

下文中，第3章对应算法概述的第(1)点，第4章对应算法概述的第(2)和(3)点，第5章对应第(4)点，第(6)章给出了算法和证明。

2.2、问题

根据第一章的描述，自然的提出了以下的问题：

问题 1: 在容错分布式系统中, 在下列 2 个条件和 1 个假设的前提下, 求一个并行区块链的算法来保证系统中数据的一致性, 且该算法比同等条件下的区块链一致性算法 $P_{\text{区块链}}$ 在同等安全级别的情况下, 效率极大地提高。

假设: 给定共识算法 P , 依据共识算法 P 对算力的定义, 系统中拜占庭节点持有的算力不能超过 51%。

以下是两个给定的条件:

条件 1: 给定区块链中区块的数据结构;
条件 2: 给定一种交易的规范, 能够检查交易的合法性, 规范必须要求交易和交易之间不能依赖冲突。

3、区块链矩阵

对于一个 $m \times n$ 的矩阵 $A = \{a_{i,j}\}$ (其中 $1 \leq i \leq n, 1 \leq j \leq m, \{n, m\} \in \mathbb{Z}^+$), 对于任意 $a_{i,j} \in A$ 是一个区块链, A 就是一个区块链矩阵。

对于任意的 $a_{i,j} \in A$, $L(a_{i,j})$ 表示区块链 $a_{i,j}$ 的高度。易知, $L(a_{i,j})$ 是属于整数域且随着时间的推移不断地变大的。

对于任意的 $a_{i,j} \in A$, $0 \leq k \leq L(a_{i,j})$, $k \in \mathbb{Z}$, $B(i, j, k)$ 表示区块链 $a_{i,j}$ 上高度为 k 的区块。

每条区块链上产生的数字货币是完全相同的数字货币。我们通过构建复杂的跨链交易, 保证这一点。

在区块链系统中, 一个区块头是只拥有父区块哈希值的, 换句话说, 区块本身是不知道他自己的高度的。我们需要在区块头中插入区块所属的

区块链和区块本身的高度，以明确单个区块的位置。

4、选区和交易通道

独立的生成区块和直接定位交易均可以通过映射直接获得，这是非常自然且容易做到的想法。因为下面 f 和 g 的选取是多样的，所以在区块链矩阵一致性算法中并不规定死如何选取，我们在算法实现的章节给出一个例子，让大家去模仿的寻找 f 和 g ，而且证明例子中的 f 和 g 是个准等概率满射是非常容易的，可以参考 SHA 的生成方案，参考文献中给出了具体的文献，具体的本文就不赘述。

4. 1、准等概率满射

若存在一个满射 $f: A \rightarrow A'$ ，其中 A' 是一个有限集， A 是一个元素总数远超 A' 中元素总数的集合或是个无限集，如果在定义域 A 中随机选取远超 A' 中元素总数的元素形成新的集合 A_1 ， A_1 中的元素通过满射 f 映射到陪域 A' 中各个元素的概率都相差是工程上可接受的，我们称 f 是一个准等概率满射。

以上并不是一个严格的定义，因为准等概率满射并不复杂而且在工程上要求也不严格，他只是帮助我们来理解如何独立的生成区块和直接定位交易。

4. 2、选区

我们把挖矿节点用来收取初始交易的关联账户成为挖矿地址。区块链矩阵 A 中拥有 $m \times n$ 个区块链，每个区块链都需要节点争夺记账权，我们要求争夺某个区块链的记账权时必须拥有特定的挖矿地址。我们把这些地址

尽量平均的分配到区块链矩阵 A 上的每个元素所代表的区块链上。就相当于我们把挖矿地址分成了不同的选区，每个选区对应唯一的一个在 A 中的区块链。

下面，我们使用正规的方式去定义一个选区。记区块链矩阵系统中所有挖矿地址的集合为 M ，记区块链矩阵 A 中所有元素的集合为 A' ，易知 A' 中共有 $m \times n$ 个元素，若存在一个准等概率满射 $f: M \rightarrow A'$ ，对 $\forall a_{i,j} \in A'$ ，所有满足： $f(addr_{i,j}) = a_{i,j}$, $addr_{i,j} \in M$ 的 $addr_{i,j}$ 的集合称为在 f 下的属于 $a_{i,j}$ 的选区。

因为任意两个选区之间的交集为空，所以区块的生成可以看做是独立进行的。

4.3、交易通道

我们把区块链矩阵 A 中的 $a_{i,j}$ 代表的区块链称之为第 (i,j) 个交易通道。易知，区块链矩阵 A 中一共存在 $m \times n$ 个交易通道。

同选区一样，我们必然可以找一个从所有交易的集合到所有交易通道集合的均匀的满射 g 。

当一个交易 Tx 发生时，它都可以通过 f 知道自己应该属于哪个交易通道，全网所有的节点也都知道 Tx 应该属于哪个交易通道，该交易通道对应的选区就会接受这个交易，然后把它打包在这个交易通道区块中。

当大量的交易同时发生时，通过 f 可以较为均匀的把这些交易映射到不同的交易通道上，因为每个交易通道上的竞争记账权是独立进行的，所以在不改变原有区块链系统的算法下，使用区块链矩阵模型可以让交易效率提高约 $m \times n$ 倍。

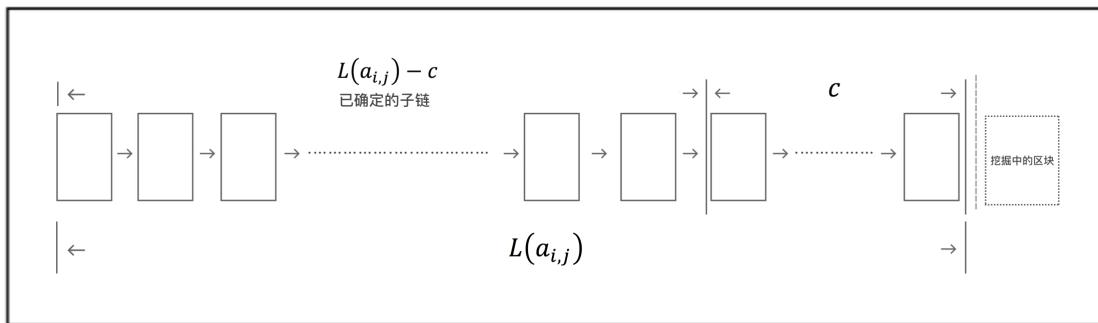
5、有效交易和已确认的子阵

5.1、概率 p 下已确认的交易

由第 2.2 章可知，对于给定的共识算法 P 和概率 p 下，必然存在一个最小的正整数 c ，使得 $a_{i,j}$ 中高度小于等于 $\text{MAX}\{0, (L(a_{i,j}) - c)\}$ (表示 0 和 $(L(a_{i,j}) - c)$ 取大的值)的所有区块中的交易都是概率 p 下已确认的交易(下文简称为已确认的交易)。我们把在 $a_{i,j}$ 中从高度从 0 到 $\text{MAX}\{0, (L(a_{i,j}) - c)\}$ 的区块构成的子链称为已确认的子链，记为 $S(a_{i,j})$ ，如下图表示。

每条区块链中高度为 0 的区块中的交易必然是已确认的交易，而且是 100% 已确认的交易，所以已确认的子链最小为高度为 0 的那个区块。

显然，在一个区块链矩阵 A 中，所有 $a_{i,j}$ 对应的 c 都是相同的，因为对于给定的 A ，算法 P 是一定的。



5.2、同链交易

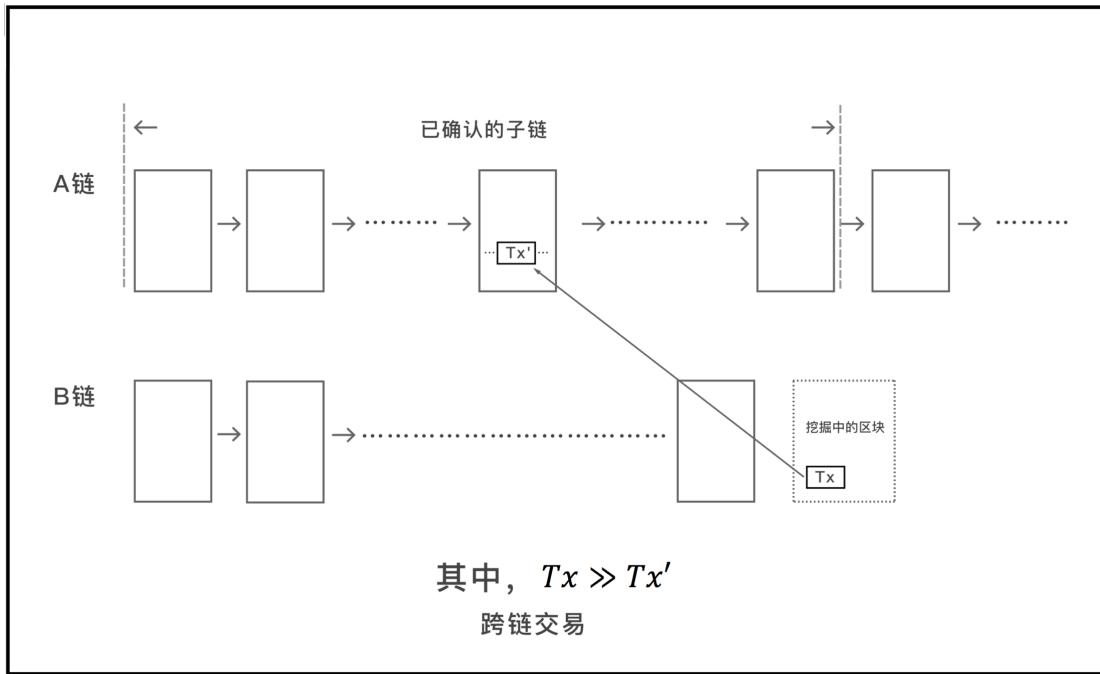
如果对于一个交易 Tx ， Tx 的依赖交易集合 T 中的任意交易 Tx' 对应的交易通道均和 Tx 对应的交易通道相同，则我们认为 Tx 是一个同链交易。

如果一个交易 Tx 是同链交易，我们称 Tx 是一个有效交易。

5.3、跨链交易

如果对于一个交易 Tx , Tx 的依赖交易集合 T 中的存在一个 Tx' 对应的交易通道和 Tx 对应的交易通道不同, 则我们认为 Tx 是一个跨链交易。

一个跨链交易 Tx , 如果 $\exists Tx' \in T$, Tx' 在区块链矩阵 A 中不是一个已确认的交易, 则我们认为这是一个无效的跨链交易。否则, 就是一个有效的跨链交易。我们称无效的跨链交易为无效交易, 有效的跨链交易为有效交易。下图给出了跨链交易的形象的解释。



5.4、子阵和已确认的子阵

区块链矩阵 A 中的每个区块链 $a_{i,j}$ 中的从高度 l_1 到高度 l_2 的子链可以用 $subchain(l_1, l_2)_{i,j}$ 来表示, 其中 $\{l_1, l_2\} \in Z^*, l_1 < l_2$, 如果 $l_1 = 0$, 我们称该子链是一个源子链。

对于一个 $m \times n$ 区块链矩阵 $A = \{a_{i,j}\}$, 如果存在一个 $m \times n$ 的区块链矩阵 $G = \{b_{i,j}\}$, 如果 $\forall i, j, 1 \leq i \leq m, 1 \leq j \leq n, b_{i,j}$ 均为 $a_{i,j}$ 的一个源子链, 我们称 G 为 A 的子阵。如果 G 中的每一笔交易在 A 中都是已确认的交易, 那么称 G 为 A 的已确认的子阵。

G_1, G_2 是区块链 A 的两个子阵，如果 $\forall b_{i,j} \in G_1, b'_{i,j} \in G_2, L(b_{i,j}) \leq L(b'_{i,j})$ ，则我们认为 G_2 比 G_1 高。对于 A 的一个子阵 G ，如果不存在另外一个子阵比 G 高，则称 G 为 A 的最高子阵。明显的， A 的最高子阵就是它自己。

令 G 是 A 的已确认的子阵，如果不存在另外一个已确认的子阵比 G 高，则称 G 为 A 的最高已确认的子阵。

以上的所有探讨都是在给定的共识算法 P 和概率 p 下。

5.5、区块链矩阵的长度

对于一个 $m \times n$ 区块链矩阵 $A = \{a_{i,j}\}$ ，区块链矩阵的长度 $LEN(A)$ 定义为：

$$LEN(A) = \sum [L(a_{i,j})]$$

5.6、排序值函数

再定义一个排序值函数，防止区块同时出现时产生冲突。

$$f(x, k) = \frac{1}{2^{[x \bmod (mn+k)]}}$$

对于一个区块 B ， x 为区块的头部 hash 值转换为十进制， k 为区块的高度。

毫无疑问，取任意区块的排序值函数之和去比较大小时，不需要真的去计算上述的函数的值即可比较成功。对于两个同时生成的区块，拥有同样的排序值函数的概率非常低。

6、区块链矩阵一致性算法

6.1、区块链矩阵一致性算法

对于给定的共识算法 P ，下面给出了一个区块链矩阵一致性算法 $P_{\text{区块链矩阵}}$ ，该算法需要在如下 4 个条件中讨论：

条件 1：给定一个 $m \times n$ 区块链矩阵 $A = \{a_{i,j}\}$, 给定准等概率的满射 f, g
确保选区的生成和交易的定位；

条件 2：给定区块链矩阵中区块的数据结构；
条件 3：给定一种交易的规范，能够检查交易的合法性，规范必须要
求交易和交易之间不能依赖冲突；
条件 4：给定一个在共识算法 P 和给定的概率 p 下的安全性量化标准，
见第 2.2 章。

区块链矩阵一致性算法 $P_{\text{区块链矩阵}}$ 如下，该算法是问题 1 的解：

- 1、新的交易发生时，向所有的对等网络中的节点发起广播；
- 2、每个节点通过广播收集这些交易，按照次序将交易排序，检查交易是否符合规范，剔除不符合规范的交易，然后把这些交易按照交易通道打包进对应区块链上新的区块；
- 3、每个节点先生成一些挖矿地址，寻找到这些挖矿地址所属选区对应的区块链，根据共识算法 P ，在算法 P 的规范下在对应的区块链上去争取生成新的区块；
- 4、当一个节点发现一个新的区块时，它向所有对等网络中的节点发起广播，区块中需要带有区块标识；
- 5、节点收到新区块的广播时，需要验证：
 - (1)、该区块被挖出的挖矿地址，是否属于正确的选区；
 - (2)、使用共识算法 P 对新区块发起验证；
 - (3)、检验区块中的交易是否符合交易规范；
 - (4)、检查是否该区块中所有的交易对于该节点来讲都是有效交易(即

同链交易或有效的跨链交易)。

如果以上四点均符合，就接受这个区块，按照区块上的标识把它连接到区块链矩阵中对应的区块链上。如果使用共识算法 P 对新区块做验证时，可能会发现自己的区块链矩阵长度不够长，即有一些区块的广播没有被收到，要先接受那些没有被收到的区块；

6、节点接受的区块接入已验证的区块链，形成新的区块链矩阵，并在此区块链矩阵上重新开始依据共识算法 P 继续寻找新的区块；

7、所有节点认为满足（1）具有最大的 $LEN(A)$ 的区块链矩阵；（2）区块链矩阵长度存在一样的时候，比较这些区块链矩阵中所有区块的排序值之和，取最大的区块链矩阵；满足以上两点的是正确的区块链矩阵，并在正确的区块链矩阵上依据共识算法 P 根据选区在对应的区块链上寻找新的区块。

该算法的输出，是整个系统关于最高已确认的子阵的一致性。

6.2、问题分析与提出

因为该算法是问题 1 的解，所以需要证明如下三个定理：

定理 2：在条件 1、2、3、4 的情况下， $P_{\text{区块链矩阵}}$ 是能达到一致性的。

定理 3：在条件 1、2、3、4 的情况下， $P_{\text{区块链矩阵}}$ 比 $P_{\text{区块链}}$ 的交易效率有极大地提高。

定理 4：定理 2 \wedge 定理 3 \Rightarrow 问题 1。

其中，定理 2 是在讲算法的正确性，定理 3 是讲交易效率的提升，定理 4 是讲定理 2 和定理 3 是可以推出问题 1 的。

6.3、证明

先证明定理 3 和定理 4，因为这两个非常简单，定理 2 是非常复杂的，最后给出证明。

定理 3 的证明：因为竞争记账权是独立的，所以在准等概率满射下，算力被分摊到每个区块链上，而区块的生成速度只与共识算法 P 有关，和算力无关，所以 $P_{\text{区块链矩阵}}$ 比 $P_{\text{区块链}}$ 大约快了 $m \times n$ 倍。得到了极大地提升。

定理 4 的证明：条件 1、2、3、4 包含了问题 1 中的两个条件，所以定理 2 和定理 3 在一起之后，是问题 1 的充分条件，即能推出问题 1。

定理 2 的证明：

定理 2 讲述了如下事实：对于给定的共识算法 P 和给定的概率 p ，区块链矩阵在 $P_{\text{区块链矩阵}}$ 运行一段时间后，最高已确认的子阵出现分叉的概率小于 $1 - p$ ，也就是说在概率 p 下，区块链矩阵的运行中不会出现已确认的交易失效的情况。

如果任意时间上，区块链矩阵均不分叉，上述结论是易知的。

所以，如果上述结论有问题，一定是因为区块链矩阵分叉引起的。我们分析区块链矩阵分叉的原因。下面给出引理 1：

引理 1：区块链出现分叉的原因在于同时生成了一系列区块，而这些区块集合中至少存在两个区块之间是不能共存的。

引理 1 证明：如果不存在不能共存的区块，那么自然不会有分叉。所以如果分叉，必然存在一些区块不能共存。下面我们需要证明，不存在一个这些区块组成的集合 $\{B_1, B_2 \dots B_n\}$ ，在集合元素个数大于 2 的情况下，集合中任意两个区块都是可以共存的，但是组合在一起就不能共存。对于

任意两个区块可以共存，因为这些区块都是同一时刻生成的，所以任意两个区块不可能处于同一条区块链。任意两个不同链的区块如果可以共存，说明两个区块中记录的交易均没有依赖冲突。如果存在一种组合起来不能共存的情况，必然在这些区块中所记录的所有交易里，拥有两个交易产生交易冲突，这明显矛盾。所以如果区块链矩阵产生分叉，原因一定是：同时生成了一系列区块，而这些区块集合中至少存在两个区块之间是不能共存的。引理 1 证毕。

我们选取两个不能共存的区块，根据区块所属的区块链分成如下两类：

- (1)、两个不能共存的区块属于区块链矩阵中同一区块链（区块链分叉的情形）；
- (2)、两个不能共存的区块属于区块链矩阵中不同的区块链（“双花”的情形）。

对于一次分叉，同时出现了一系列区块，如果有任何一个区块和其他任意区块都可以共存，那么就把满足这个条件的区块都去掉，会得到一个相互之间不能共存的区块集合 $\{B_{i,j}^1, B_{i,j}^2 \dots B_{i,j}^{k_{i,j}} \dots B_{s,t}^1, B_{s,t}^2 \dots B_{s,t}^{k_{s,t}}\}$ 。其中 $B_{i,j}^p$ 的脚标代表所属的是区块链矩阵A的区块链 $a_{i,j}$ 。根据集合的获取方式，易知其中任意一个区块，均有另外一个区块处于该集合中，他们是不能共存的。

任意一个节点，接受到这些区块的顺序都不同，所以理论上会产生有限种排列组合，这些排列组合的集合为 $\{S_1, S_2 \dots S_r\}$ ，每个组合对应一组能最多共存的区块的组合。

如果存在一个组合 S_r ，其中区块的数量是大于任意一种其他组合的，那么根据 $P_{\text{区块链矩阵}}$ 的第 7 点，则该组合直接胜出，不会产生区块链矩阵的

分叉。

如果不存在这么一个组合，必然有多于 2 个的组合，这些组合的可共存的区块链是在所有组合中最多的，而且这些组合是一样多的，分叉就在这些组合中产生了，有多少这样的组合，理论上就有多少分叉。考虑任意一个这样的组合，易知，该组合中任意两个区块不处于同一条区块链。

根据 $P_{\text{区块链矩阵}}$ 的第 7 点，所有的节点都应该选取的是上述的组合，因为他们所拥有的区块数最多。

设 q 为在单条区块链中两个区块同时生成的概率，则在区块链矩阵中，同时生成两个区块的概率为 $m n q$ ，毫无疑问，同时生成超过 2 个区块的概率是远小于 $m n q$ 的。在上述被选中组合中，任意两组在一起共同出现的概率毫无疑问不高于 2 个区块同时生成的概率，也就是远小于 $m n q$ 。

区块链矩阵中采取的区块排序规则表明，两个区块拥有相同值的概率为 q_1 ，如果有两个区块值不同，则两个组合不可能产生同样的值，也就是说，根据 $P_{\text{区块链矩阵}}$ 的第 7 点，就可以抉择出正确的组合。如果没有出现，那这种概率是不高于两个区块值不同的概率，即 q_1 。

也就是说，如果出现两个以上的组合，他们之中区块数量相同，排序值也相同，那么这种概率不高于 $m n q q_1$ 。根据题设，其中 $m n q_1 < 1$ ，所以这种概率低于 q 。

考虑一个时刻，该时刻内所有有效的节点认可此时的最高已确认子阵为 A_H ，如果在下一次出块中， A_H 分叉，则必然有一些区块，他们在这次出块中进入到了最高已确认子阵，但是没有结束分叉，也就是说，这些区块后续的区块在这次出块中包含自身拥有了 c 个区块，而这些区块都有其

他区块是和它同时出块的，这种经历必然连续拥有 c 次，也就是说，概率低于 q^c ，根据定理 1 可知，该概率低于 $1 - p$ 。

否则，分叉不会发生，也就是说，是符合安全性要求的。

定理 2 证毕。

下面再给出一个在拉取区块中，不会出现锁的问题的证明：

在获取过程中，因为是并行的链，所以我们需要证明如下问题：即在现有区块链矩阵 A 的状态下，不存在两个新区块 $B(i, j, k)$ 和 $B(p, q, t)$ ，使得 $B(i, j, k)$ 加入 A 是依赖于 $B(p, q, t)$ 先加入 A 中，而 $B(p, q, t)$ 加入 A 是依赖于 $B(i, j, k)$ 先加入 A 中的，即出现死锁。

假设上述问题出现，则在区块链矩阵 A 中， $B(i, j, k)$ 对应的区块链 $a_{i,j}$ 和 $B(p, q, t)$ 所对应的区块链 $a_{p,q}$ 必然是两条不同的链，否则易知不会出现上述问题。则在 $a_{p,q}$ 必然存在一个区块 $B(p, q, t')$, $t' < t$, $B(p, q, t')$ 不是 $a_{p,q}$ 已确认的子链 $S(a_{p,q})$ 中的区块，而当 $B(p, q, t)$ 加入 A 中后， $B(p, q, t')$ 就变成是 $S(a_{p,q})$ 中的区块，并且使得存在：一个交易 Tx'_1 是记录在 $B(p, q, t')$ 内的，一个交易 Tx 是记录在 $B(i, j, k)$ 内的，且 $Tx \gg Tx'_1$ 。

如果这两个块都是诚实节点给出的块，那么必然在网络中存在一个节点 M_1 ，在 M_1 上， $B(i, j, k)$ 还没生成的情况下， $B(p, q, t)$ 已经生成了。而根据假设， $B(p, q, t)$ 加入 A 是依赖于 $B(i, j, k)$ 先加入 A 中的。这明显矛盾。

所以不会存在锁死的问题，同理可证，不会出现环状死锁（即区块 B_1 依赖 B_2 ， B_2 依赖 B_3 ， B_3 依赖 B_1 ）的问题。

6.4、算法实施

在本例中，我们直接使用比特币系统中对交易的 $txid$ 的生成办法和矿

工地址的生成办法。

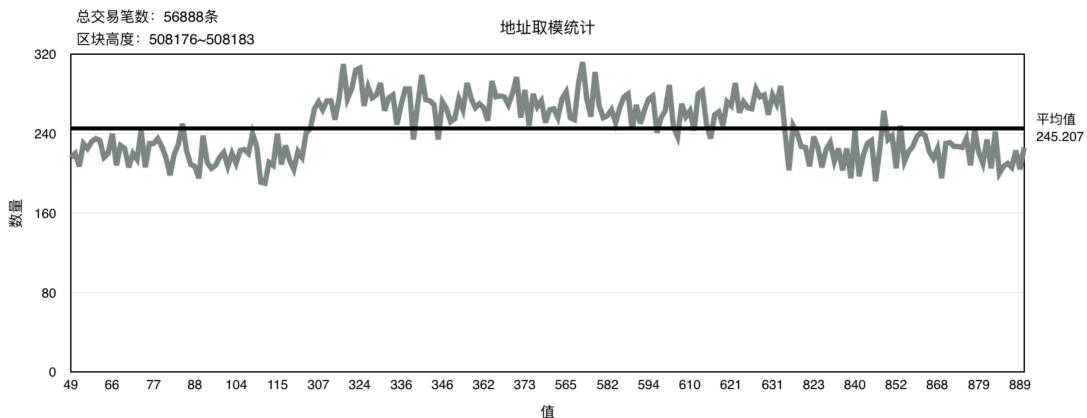
我们选取 $m=1024$, $n=1$ 这一区块链矩阵。

6.4.1、矿工地址到选区的满射

下面，我们给出一个从矿工地址到选区的较为均匀的满射 f 。

对于任何一个矿工地址 $addr$, 把该地址转换为 ASCII 码, 再把该二进制换算成十进制, 然后对 $m \times n$ 取模。易知, 该 f 是一个满射。在本例中, 使用的是在比特币系统中矿工地址的生成方式。

本文的作者顺序的抽取了从高度为 508176 到 508183 的 8 个区块中共计 56888 个已经发生在比特币中的出现的矿工地址, 然后根据 f 算出了如下的分布结果:



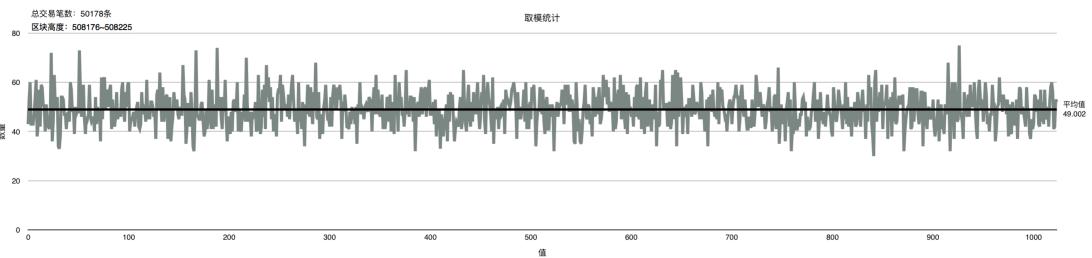
6.4.2、交易到交易通道的满射

下面，我们给出一个从交易到交易通道的较为均匀的满射 f 。

定义一个从所有交易的集合到所有交易通道的集合的映射 f , 对于任何一个交易 Tx , 把该交易对应的唯一的 $txid$ 换算成十进制, 然后对 $m \times n$ 取模。易知, 该 f 是一个满射。在本例中, 使用的是在比特币系统中 $txid$ 的生成方式。

本文的作者顺序的抽取了从高度为 508176 到 508225 的 50 个区块中共

计 50178 个已经发生在比特币中的交易对应的 $txid$ ，然后根据 f 算出了如下的分布结果：



6.5、算法探究

本文中给出同链交易的目的，是在于满足小额极速转账的需求，可以理解为同链交易是不需要确认即可转账的，跨链交易需要等到其他人确认，这意味着，如果用户使用的钱包够智能，完全可以实现到账之后立刻转出，这在小额频繁交易中很有用。

区块链矩阵中的分叉是无处不在的，我们通过选取排序值函数可以对其做优化。其实，在实施过程中，完全可以让挖矿节点更正确的判别双花攻击来控制问题的存在，只需要持有一个交易一段时间即可。但是这不能作为全系统评判的标准。

拜占庭系统中的一致性问题是非常繁琐和复杂的，区块链矩阵技术解决的是在非拜占庭节点算力占优的一致性问题，达成的是所有节点拥有的区块链矩阵，他们的最高已确认的子阵是一致的。

区块链系统相当于是单核的区块链矩阵系统，将其应用范围选取不同的准等概率满射和不同的矩阵，能在同等共识算法和同等安全性方面极大地获得效率提升，消耗的是节点的服务器要求。然后在现实区块链应用中，普通电脑依然是无法胜任任何工作的，所以这并不改变现有区块链中各节点的应用人群。任何的区块链项目都可以平移至区块链矩阵项目，所需的

只是底层上的开发，对于上层应用并无影响。将单核处理器变为多核处理器是个必然的过程，所以每个区块链项目都应该替换为区块链矩阵项目。

因为每个节点均可以使用多个挖矿地址，所以对于矿池等挖矿行为，并无本质的改变，而且因为挖矿消耗的是某种算力，共识算法是均衡出块时间，所以平摊算力是合理的，算力会自动汇聚到高度低的区块链上去，因为那里的区块获得较大排序值的可能更大。

参考文献:

- [1] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, <https://bitcoin.org/bitcoin.pdf>, 2008.
- [2] Andreas M Antonopoulos, Mastering Bitcoin, <https://github.com/bitcoinbook/bitcoinbook/blob/develop/book.asciidoc>, 2017.
- [3] dantheman, DPOS Consensus Algorithm – The Missing White Paper, <https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper>, 2017.
- [4] delegated-proof-of-stake-consensus, <https://bitshares.org/technology/delegated-proof-of-stake-consensus/>.
- [5] vitalik buterin, A Next-Generation Smart Contract and Decentralized Application Platform, <https://github.com/ethereum/wiki/wiki/White-Paper>, 2013.
- [6] Vlad Zamfir, introducing-casper-friendly-ghost, <https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/>, 2015.