基于信用模型的工作量证明算法

王缵[1,2,3]，田有亮[1,2,3]，李秋贤[1,2,3]，杨新欢[1,2,3]

（1. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；2. 贵州省公共大数据重点实验室（贵州大学），贵州 贵阳 550025；
3. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025）

摘 要：提出了一种基于信用模型的共识协议。首先，该共识协议借鉴了个人信用风险评估的思想，设计了一种基于 BP 神经网络的节点信用度模型。其次，构造了一种分片轮转模型，它可以根据节点的信用度高低分割搜索空间产生新区块，同时对协议所面临的可能攻击进行分析，修复了协议存在的漏洞。最后，仿真实验表明共识协议既能有效地降低新区块产生过程中重复计算的巨大资源消耗，也能抑制大型矿池的产生，使整个区块链系统变得更加安全可靠。

关键词：PoW 共识；BP 神经网络；信用度模型；搜索空间；区块链
中图分类号：TP302
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018138

Proof of work algorithm based on credit model

WANG Zuan[1,2,3],TIAN Youliang[1,2,3], LI Qiuxian[1,2,3], YANG Xinhuan[1,2,3]

1. College of Computer Science & Technology, Guizhou University, Guiyang 550025, China
2. Guizhou Provincial Key Labortory of Public Big Data (Guizhou University), Guiyang 550025, China
3. Institute of Cryptography & Data Security, Guizhou University, Guiyang 550025, China

Abstract: A consensus protocol based on the credit model was proposed. Firstly, the consensus agreement drew on the idea of personal credit risk assessment and a node credit model based on BP neural network was designed. Secondly, a piecewise rotation model was also constructed to segment the search space according to the node's credit level to generate new blocks. At the same time, the possible attack of the protocol was analyzed and the vulnerability of this protocol was fixed. Finally, the simulation experiments show that the protocol not only effectively reduces the huge resource consumption in the process of new block generation, but also suppresses the generation of the large mine pool, making the whole blockchain system more secure and reliable.

Key words: PoW consensus, BP neural network, credit model, search space, blockchain

1 引言

文献[1]是公认最早的关于区块链的描述性文章，但该文献并没有明确提出区块链的定义和概念，只是提出了一种电子现金系统。在该文献中，区块链被描述为用于记录虚拟货币交易的一种分

收稿日期：2017−10−30；**修回日期**：2018−07−20
通信作者：youliangtian@163.com
基金项目：国家自然科学基金资助项目（No.61662009, No.61772008）；贵州省教育厅科技拔尖人才基金资助项目（No.[2016] 060）；贵州省科技重大专项计划基金资助项目（No.20183001）；贵州省科技计划基金资助项目（No.[2017]5788）；教育部—中国移动科研基金研发资助项目（No.MCM20170401）；贵州大学培育基金资助项目（No. [2017]5788）；贵州省联合基金资助项目（No. LH20147476）

Foundation Items: The National Natural Science Foundation of China (No.61662009, No.61772008), Topnotch Talent in Science and Technology Support Program of Guizhou Province Education Department (No.[2016] 060), Science and Technology Major Support Program of Guizhou Province (No.20183001), Guizhou Provincial Science and Technology Plan Project (No.[2017]5788), Ministry of Educatio China Mobile Research Fund Project (No.MCM20170401), Guizhou University Cultivation Project (No. [2017]5788), The Joint Science and Technology Foundation of Guizhou Province (No.LH20147476)

布式账本技术[2]，它通过密码学中的椭圆曲线数字签名算法实现去中心化的 P2P 系统。但区块链的应用不只局限于虚拟货币等电子货币系统，还涉及隐私保护[3]、金融服务、物联网与供应链等众多领域。

区块链技术原理来源于数学上的拜占庭将军问题[4-7]。在互联网活动大背景下，拜占庭将军问题是指当与不熟悉的对方进行价值交换时，参与者如何才能不会因为恶意攻击者的欺骗和迷惑而做出错误的决策；从互联网技术领域的角度看，拜占庭将军问题是指当不存在可信第三方（TTP, trusted third party）时，分布在网络中的各个节点应如何达成共识[8]。从这些角度看，区块链技术在不需要单个信任节点的情况下可以很好地解决拜占庭将军问题。

区块链具有可追溯、不可篡改、去中心化等优势。然而和基于传统分布式一致性算法[5,9]的分布式系统一样，区块链系统也会存在传输消息延时、损坏、丢失、篡改等问题。此外，去中心化的特点决定系统不存在被信任节点，甚至存在恶意节点，以及各种原因导致的数据不一致等问题。为了实现必要的容错，区块链系统需要一个高效的共识机制来确保每一个节点都有唯一公认的全局账本。基于这种诉求，专家学者相继提出了工作量证明（PoW, proof of work）、权益证明（PoS, proof of stake）、股份授权证明（DPoS, delegated proof of stake）、活动证明（PoA, proof of activity）等在内的多种共识机制。其中，权益证明机制是利用区块生成难度与节点所占股权成反比来进行“挖矿”[10]；股份授权证明机制的是利用股东投票数最多的几位代表按既定时间段轮流产生区块[11]。Vitalik 提出了类 PoS 共识 Casper。Bentov 等[12]提出了活动证明，用来代替虚拟货币的激励结构。Sawtooth 项目应用了基于 Intel SGX 可信硬件的逝去时间证明（PoET, proof of elapsed time）机制。Burstcoin 加密货币是基于硬盘容量空间的能力证明（PoC, proof of capacity）。这些共识机制在资源消耗、安全性或共识时间等方面各有侧重。作为区块链技术最成功的应用，虚拟货币系统就是采用工作量证明，非常巧妙地解决这些问题，实现交易的不可伪造性和不可篡改性，而该共识机制也是虚拟货币系统安全模型的基石，是虚拟货币正常运行约 10 年的关键所在。

所谓的工作量证明，其核心思想就是各个节点通过算力竞争来解决一个 hash 难题，以此来保证数据的一致性和共识的安全性。具体地，各个节点不断猜测并验证随机数值是否为一个 SHA256 数学难题的解，其中，最快解决该难题的节点将获得区块链记账权和虚拟货币奖励。当某个节点成功解决该难题并产生区块，就会广播这个合法区块。同时，收到的用户会验证这个区块，如果验证通过，就会将这个区块接入区块链上，并在此基础上继续尝试解决新的 hash 难题。目前，除了暴力计算外，还没有有效的算法来解决这些求解复杂但验证简单的 hash 难题。换而言之，如果成功解决了该 hash 难题，则说明在概率上该节点付出了对应的算力。即 PoW 共识机制会导致算力越大，解决问题的概率就越大的问题。当某个节点或矿池控制超过全网一半的算力时，从概率上就能掌控区块链的走向，这也是所谓的 51%攻击。很显然，参与 PoW 共识的节点将付出很大的经济成本（硬件、电力、维护等）。当没有成为首个算出合理 hash 值的节点时，这些成本都将是没有回报的。

在虚拟货币系统中，产生区块的过程称为挖矿，从事挖矿活动的参与者称为矿工[13]。由于虚拟货币大约每 10 min 产生一个区块，这就意味着大部分矿工在一定时间内很难产生区块并获得奖励。因此，矿工们会选择加入开放矿池进行合作挖矿，以此来获得稳定的收益。矿池[14]一般分为 2 种，一种是托管矿池，另一种是 P2P 矿池。具体地，P2P 矿池是一种去中心化的矿池服务器，其原理与区块链系统类似，也被称为份额链。份额链的工作量证明难度低于虚拟货币区块链，其上记录了贡献工作量证明的矿工份额。当一个份额区块的难度达到了虚拟货币网络的难度目标时，就会奖励所有已经在份额链区块中取得份额的矿工。P2P 矿池虽然实现了去中心化，但其挖矿方式复杂，对节点性能要求非常高，效率远不如托管矿池，所以没能吸引太多算力；托管矿池中的矿工需要消耗资源来不断尝试产生新的合法区块，即发送完整工作量证明给矿池管理者。但是完整工作量很难产生，矿工可以选择发送部分工作量证明获得相应收益。无论是哪个矿工产生区块，获得的收益将按贡献比例分给每个矿工。这种集中式的矿池通过这样的方式不断聚集算力，已经开始引起 51%攻击的担忧。同时，由于中心化控制的矿池存在矿池管理者，很容易出现管理者出于利益而实施攻击的风险。

针对 PoW 共识计算过程中所有节点（包括无

法获得奖励的节点）资源消耗大、达成共识周期较长及矿池中心化等问题，本文提出了一种基于信用模型的工作量证明（CPoW, proof of work based on credit）算法。在这个新的共识机制中，本文利用 BP 神经网络[15-17]设计了一种基于节点信用评估体系的节点信用度评估模型，根据这个模型，可以定量地计算各个参与 PoW 共识节点的信用度并进行信用度排名。然后，根据这个信用度排名按比例划分 SHA256 数学难题的搜索空间给计算节点，让每个节点计算验证搜索空间是否满足这个 hash 难题，直到有某个节点通过验证的方式成功解决 hash 难题，此时这个节点将获得区块链的记账权并得到相应奖励。如果某个节点已经计算验证完划分给自己的搜索空间，但是本轮计算还没有解决 hash 难题，那么这个节点还可以继续根据信用度排名来获取新一轮的搜索空间。本文将这种划分搜索空间的方案叫作分片轮转模型。

CPoW 算法通过分片轮转模型，让所有参与工作量证明的节点一起合作验证搜索空间来解决 hash 难题，实现了搜索空间分配的去中心化，大大提高了解决 hash 难题的效率，从而缩短了共识达成的周期，也避免了巨大资源消耗的重复。由于验证搜索空间存在概率性，在一定程度上抑制了大型矿池的产生。

2　系统模型

2.1　虚拟货币节点信用度模型

2.1.1　节点信用度评估指标体系

所谓节点“信用”是指节点在参与共识机制过程中的表现，是对节点运行状况、节点账户财富水平和诚信水平的全面考察。对于节点信用度评估，首先要建立评估指标体系。根据节点在网络中运行的实际情况，本文提出了一种分层结构指标体系，如表 1 所示。其中，二级指标 10 项，为反映节点运行情况、节点账户财富水平和诚信水平的具体指标信息；一级指标 3 项，为对二级指标的归纳，反映了节点信用评估的 3 个方面。

2.1.2　数据的标准化处理

在节点信用度评估要素中，包含网络延时时间段、节点离线次数段、节点离线时间段、节点获取搜索空间次数段、节点加入网络时间段、节点提供分叉区块次数段、节点是否提供无效区块这 7 项离散数据，对于这些数据，要进行统一量化，即根据不同属性值在共识过程中的重要性对其属性值赋予不同的值。属性值量化如表 2 所示。

按照表 2 统一量化后，可以通过最小—最大规范化方法按比例进行缩放，使之落入[0,1]区间，再作为神经网络模型的输入。具体方法为：设 min 和 max 分别为某一属性的最小值和最大值，最小—最大规范方法是指通过计算

$$v' = \frac{v - \min}{\max - \min} \tag{1}$$

将属性值映射到[0,1]中。例如，对节点加入网络时间段，如果属性值为 28 h，则记分为 8，该属性的记分中最大值为 11，最小值为 1，根据式(1)，新属性值为

$$v' = \frac{8-1}{11-1} = 0.7 \tag{2}$$

表 1　　**节点信用度评估指标体系**

一级指标	二级指标	指标解释	属性值性质
节点账户的财务能力	coin age	交易金额乘以其代币存在于钱包中的时间	近似正态分布
	虚拟货币流动比率	账户中支出、收入的虚拟货币之和与时间的比值	近似正态分布
节点的性能	网络延时时间段	将节点网络延时的时间分成各个时间段	离散
	节点离线次数段	将节点离线的次数分成各个次数段	离散
	节点离线时间段	将节点离线的时间分成各个时间段	离散
	节点获取搜索空间次数段	将节点获取 nonce 值搜索空间的次数分成各个次数段，即节点算力的等级	离散
节点的诚信水平	节点加入网络时间段	将节点加入网络的时间分成各个时间段	离散
	节点提供分叉区块次数段	将节点提供分叉区块的次数分成各个次数段	离散
	是否提供无效区块	略	离散
	节点上一轮的信用度	略	近似正态分布

表 2 节点信用度离散数据记分表

网络延时时间段		节点离线次数段		节点离线时间段		节点获取搜索空间次数段		节点加入网络时间段		节点提供分叉区块次数段		节点是否提供无效区块	
区间/ms	积分	区间	积分	区间/h	积分	区间	积分	区间/h	积分	区间	积分	区间	积分
(0,30]	12	0	12	(0,0.5]	11	(1000,+∞]	12	(72,+∞]	11	[0,1]	11	是	1
(30,50]	8	(0,2]	8	(0.5,2]	8	(800,1000]	8	(24,72]	8	(1,3]	8	否	6
(50,80]	6	(2,5]	6	(2,24]	4	(400,800]	6	(12,24]	4	(3,5]	4		
(80,100]	4	(5,8]	4	(24,+∞]	1	(100,400]	4	(0,12]	1	(5,8]	2		
(100,+∞)	2	(8,+∞]	2			(0,100]	2			(8,+∞]	1		

通过这种方法，可以处理所有给定的最大、最小的离散属性的数据。

在节点信用评估的输入要素中，有 coin age、虚拟货币流动比率、节点上一轮的信用度这 3 个属性值数据，其中，节点上一轮的信用度是标准化的数据，其他 2 个近似于正态分布，这些要素的属性值可以通过正态分布函数进行转化，将属性值映射在(0,1)的数值。正态分布函数如图 1 所示，其函数为

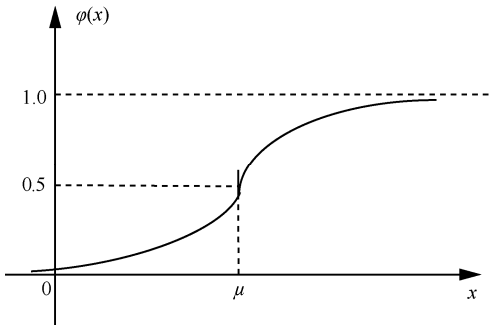


图 1 正态分布函数曲线

$$\phi(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}\mathrm{d}x,\ -\infty<x<\infty \tag{3}$$

其中，μ、σ 为常数，简记为 $X\sim N(\mu,\sigma^2)$。

在节点信用度评估模型中，不同属性的 μ 和 σ 的取值如表 3 所示。

表 3 μ 和 σ 的取值对应表

属性	coin age	虚拟货币流动比率
μ	133	0.5
σ	20	0.15

例如，某节点的虚拟货币流动比率为 0.6，根据式(3)，得

$$y_1=\int_{-\infty}^{0.6}\frac{1}{\sqrt{2\pi}0.15}\mathrm{e}^{\frac{-(t-0.5)^2}{2(0.15)^2}}\mathrm{d}t=0.745\,4 \tag{4}$$

其中，y_1 为转换后的属性值。

通过这种方法，就可以将节点信用评估要素中的属性值映射到[0,1]区间的值，便于信用度模型的处理。

2.1.3 节点信用度模型构造

在这里，本文选择的转移函数为单极性 sigmoid 函数，函数表达式为

$$f(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{5}$$

该函数是一个实数域 R 到[0,1]闭集的非减连续函数，代表了状态连续的神经元模型。

据此，本文设计 3 层神经网络信用评估模型，如图 2 所示。其中，输入层节点数为 10，以网络延时时间段、节点离线次数段、节点离线时间段、节点获取搜索空间次数段、节点加入网络时间段、节点提供分叉区块次数段、节点是否提供无效区块、coin age、虚拟货币流动比率、节点上一轮的信用度作为输入向量，即输入向量 $\boldsymbol{X}=(x_1,x_2,\cdots,x_{10})$；隐层节点数为 3，即隐层向量 $\boldsymbol{Y}=(y_1,y_2,y_3)$，分量依次表示节点账户的财务能力、节点的性能和节点的诚信水平；输出层节点数为 1，用变量 $Z=(z)$ 表示，其输出值为模型实际输出，且 $z\in[0,1]$。将根据经验得出的实际输出数据，转化为[0,1]的数值，并作为期望输入，即 $D=(d_1)$；输入层节点到隐层节点的权值用向量 $\boldsymbol{V}=(v_{1,1},v_{1,2},\cdots,v_{10,3})$ 表示，隐层节点到输入层节点的权值用向量 $\boldsymbol{W}=(w_{1,1},w_{2,1},w_{3,1})$ 表示。

对于隐层，有

$$y_j = f\left(\sum_{i=1}^{10} v_{ij}x_i\right), j=1,2,3 \tag{6}$$

对于输出层，有

$$z = f\left(\sum_{j=1}^{3} w_1 y_j\right) \tag{7}$$

其中，$f(x)$为转移函数。节点的信用评估 3 层神经网络模型如图 2 所示。

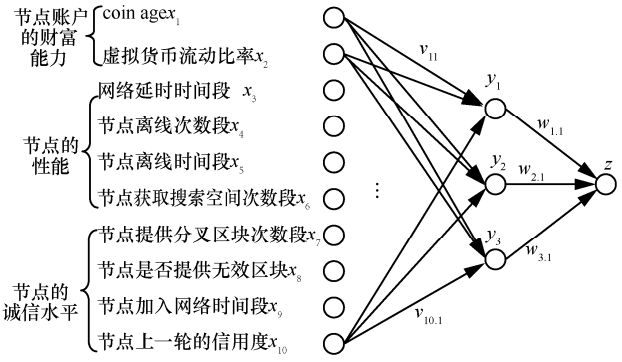


图 2　节点信用评估 3 层神经网络模型

神经网络通过不断学习，会对各个权值进行调整，使误差不断减少。根据 BP 算法，得出模型的权值调整式为

$$\Delta w_{j1} = \eta(d_1 - z)(1-z)zy_j,\ j=1,2,3 \tag{8}$$

$$\Delta v_{ij} = \eta(d_1 - z)z(1-z)w_{j1}(1-y_j)y_j x_i, j=1,2,3 \tag{9}$$

其中，η 为学习效率，是一个常数。对于每一个输入的样本，计算出相应的 Δw_{j1} 和 Δv_{ij}，得到权值调整式为

$$w_{j1} \Leftarrow w_{j1} - \Delta w_{j1},\ j=1,2,3 \tag{10}$$

$$v_{ij} \Leftarrow v_{ij} - \Delta v_{ij},\ j=1,2,3 \tag{11}$$

当 z 和 d_1 的误差达到要求的精度时，算法停止，学习过程结束。

2.1.4　模型评价

基于 BP 神经网络的算法思想，通过对节点历史数据的训练和学习，调整模型神经单元之间的连接权重，确定输入和输出的内在联系，建立了节点的信用度模型，使模型具备了对节点信用进行评估的能力。通过该模型进行节点信用的评估，挑选出了性能高的、网络环境良好的非拜占庭节点，淘汰那些拜占庭节点，使参与共识的节点都能获得一致的信用度。

2.2　分片轮转模型

所谓分片是指对 hash 难题的搜索空间进行划分，每个节点按照自己的信用度排名去获取自己需要验证的搜索空间。所谓轮转是指每个节点计算验证完成自己获取到的搜索空间之后，如果 hash 难题仍未解决，它依旧可以再次获取需要验证的一定范围的搜索空间。如此不断地获取需要验证的搜索空间，直至 hash 难题得到解决。

2.2.1　搜索空间的问题

“挖矿”本质是执行 hash 函数的过程，而 hash 函数是一个单输入单输出函数，输入数据被称为区块头（blockheader），所以本节首先解析区块头结构。例如，虚拟货币区块头中共 6 个字段，介绍如下。

nVersion：区块版本号，存储空间为 4 B。

hashPrevBlock：上一个区块的区块头的 hash 值，存储空间为 32 B。

hashMerkleRoot：包含进本区块的所有交易构造的 Merkle 根，存储空间为 32 B。

nTime：Unix 时间戳，存储空间为 4 B。

nBits：区块的难度值，存储空间为 4 B。

nNonce：随机数，存储空间为 4 B。

对于虚拟货币系统中的 PoW 共识机制而言，矿工可以自由调整的字段就 3 个，分别是 nTime、nBits、nNonce。nTime 合理的区块时间是有一定范围的，比前一个区块时间太早会被其他节点拒绝；nNonce 提供 2^{32} 种可能取值；hashMerkleRoot 理论上提供 2^{256} 种可能，对包含进区块的交易进行增删、调整顺序或修改 Coinbase 交易的输入字段都会导致本字段变化。

对于 CPoW 共识机制而言，矿工可以自由调整的字段就只有 2 个，分别是 nNonce 和 nTime 字段。nNonce 提供 2^{32} 种可能取值；nTime 取值范围是上一个区块产生时间至产生后的 2 min。hashMerkleRoot 的值是固定的，并严格规定了交易格式的填充，这保证了同一笔交易填充的一致性。对于所有在 2 min 内产生的交易全部按照时间顺序纳入区块，并指定 Coinbasse 交易为这段时间中产生的第一笔交易。通过以上方法，即可保证 hashMerkleRoot 值的一致性。

2.2.2　分片轮转模型构造

在这里，划分搜索空间是指划分 nNonce 字段的搜索空间，而对于 nTime 字段，模型不做搜索空

间的划分，即每个节点都拥有完整的 nTime 字段搜索空间。

假设待验证的每一轮的 nNonce 搜索空间的一个标准范围为 *runit*，*R* 为轮转次数，总的节点数为 *num*，节点 *i* 的信用度排名为 a_i（a_i 值越小，排名越低），则排名为 a_i 的节点对应的第一轮的 nNonce 的搜索空间的范围大小为

$$\frac{2runit \cdot a_i}{(1+num)num} \tag{12}$$

这样做的目的是信用度排名较高的节点可以一次获取更多的 nNonce 字段搜索空间，避免每次获取搜索空间的计算消耗，本文根据信用度排名顺序按照比例划分搜索空间，则排名为 a_i 的节点对应的第一轮 nNonce 的搜索空间的范围为

$$\left[\frac{a_i(a_i-1)runit}{(1+num)num}, \frac{a_i(a_i+1)runit}{(1+num)num}\right) \tag{13}$$

如果某节点已经验证完分配的 nNonce 的搜索空间和 nTime 字段搜索空间的组合后，但并没有找到符合 hash 难题要求的 nNonce 值，该节点先会查看有没有收到来自网络的其他节点解决该难题的 nNonce 值。如果没有，则它可以通过计算获取 *R* 轮（下一轮）的搜索空间的范围为

$$\left[\frac{a_i(a_i-1)runit}{(1+num)num}+(R-1)runit, \frac{a_i(a_i+1)runit}{(1+num)num}+(R-1)runit\right) \tag{14}$$

2.2.3 模型评价

根据信用模型产生节点信用度，并进行排名，通过排名的顺序，节点按照分片轮转模型来获取自己的搜索空间，从而实现了搜索空间分配的去中心化。由于 nTime 字段的搜索空间非常有限，因此并没有对其搜索空间进行划分，而是给每个节点都分配全部搜索空间。当然，由于在分片轮转模型中对区块头的规定限制了搜索空间的大小，这也是模型有待改进之处。

3 共识算法

算法同时提供了安全性和可用性，只要参与共识的拜占庭节点数不超过 $\left\lfloor\frac{num-1}{3}\right\rfloor$，就能保证系统能够达成正确的共识。由于实际上全局账本仅由参与共识的节点来维护，因此系统中的普通节点不参与共识算法，但是可以查看共识的全过程。如果普通节点要参与共识计算，它们需要获得相应的信用度值，只有通过信用模型评估获得一定额度的信用度之后，节点状态记录表才会登记这些普通节点。这些被登记的普通节点才有机会参与到共识的过程中，并获得搜索空间。

参与共识的节点必须拥有一定的信用度排名。这个信用度排名不是每轮共识都要计算的，只有当产生 1 440 个区块后，系统才会提前更新排名顺序，各个参与共识的节点根据新一轮的排名进行搜索空间的验证。

每个参与共识的节点需要维护一个信用度排名数组 *a* 和节点状态记录表 *t*。系统为每一个参与共识的节点分配一个序号，从 1 开始，最后一个节点的编号为 *num*，数组的下标即为节点的编号，数组存储的就是节点的信用度排名，而对于 *a*[0]存储的是参与共识节点的个数。状态记录表 *T* 记录了所有参与共识节点的网络延时情况、离线情况、节点提供分叉区块的情况、节点是否提供违法区块的情况、节点获取搜索空间的次数、节点上一轮的信用度等。整个算法分为 4 个步骤：初始化阶段、构建区块、校验新区块和区块链的组装。

3.1 一般流程

算法要求每次产生区块的时间间隔大约为 2 min，执行的一般流程如下所示。

1) 初始化阶段

在初始化阶段，各个节点根据状态记录表按照信用度模型计算自己的信用度。然后，由主节点发起信用度排名请求，主节点 $p=(q+h)\bmod num$ 给出，其中，*q* 为阶段数，*h* 为区块链长度，*num* 为上一轮参与共识的节点个数。算法 1 给出了信用度排名的伪代码。令 master 为主节点，slave 为参加共识的节点，C_i 为节点 *i* 的信用度。

该算法初始由主节点新建一个空的数组 *a*，由于信用度的范围为(0,1)，因此主节点设置一个初始值 *step*=0.5（*step* 范围为[0.5,0.51)）。

主节点将 *step* 发给每个节点，每个节点比较自己的信用度与发过来的 *step*，如果自己的信用度在 *step* 的区间，则向主节点返回 message=<*reply*, *t*,*i*,*step*,*c*>和对 message 的签名，其中，*reply* 是一个标志码，*t* 是时间戳，*i* 是节点编号，*c* 是节点自己的信用度。主节点先验证签名是否合法，然后查看状态记录表，核实此节点是否已经“报到”，如果

没有就将这些“报到”的节点按照信用度从小到大进行排名并写入排名数组中，最后在状态记录表中标记这个节点。

重复上一步，直到 *step* 达到 1 时，算法终止。

算法 1　信用度排名算法

1)　$a[num]=\{0\}$,$step=0.5$
2)　repeat
3)　　master 广播 *step*
4)　　if $C_i \geqslant step \&\& C_i < step+0.01$ then
5)　　　slave 发送 message
6)　　end if
7)　　if 签名合法&&未报到 then
8)　　在 a 中添加 C_i
9)　　　标记 C_i
10)　　end if
11)　$step = step+0.01$
12)　until $step = 1$
13)　$result= a$

通过信用度排名算法完成节点信用度的排名，然后将排名分发到各个参与共识的节点。由于网络中可能存在 f 个拜占庭节点，同时系统需要足够数量的非失效节点的响应，故要满足 $num>3f$。在 $num>3f$ 的情况下，如果一个好节点提议排名数组 a，不会有其他的好节点提议另一值 b（a 和 b 不同）。同时，因为存在 f 个拜占庭节点，无法确定主节点不是拜占庭节点，所以要重复排名算法 $f+1$ 次，本文将这种重复称为阶段，且每个阶段的主节点不能相同。这样就会保证在 $f+1$ 次重复中，必定有一个主节点为好节点。本文在文献[18]的基础上，修改其算法并给出了具体的伪代码，如算法 2 所示，成功解决了拜占庭协议问题。当满足以下 2 个条件之一时，算法就会达成协定。

①当 $num>3f$ 时，如果某一轮的主节点是好节点，所有好节点在这轮之后都不会改变它们的值，即不会改变排名数组。

②所有节点起始值（即排名数组）相同，则好节点都会提议这个值。所有好节点将会接收到至少 $num-f$ 个提案，因此所有好节点将会继续保持这个值，并且不会切换到主节点的值。

算法 2　信用度排名分发算法

1)　a 为排名数组
2)　for 从第 1 到第 $f+1$ 个阶段 do
3)　　执行信用度排名算法进行排名
4)　　广播 *value*(a)
5)　　if 接收到 b 至少 $num-f$ 次 then
6)　　　广播 *propose*(b)
7)　　end if
8)　　if 接收 *propose*(c)至少 f 次 then
9)　　　$a=c$
10)　　end if
11)　　设节点 v_i 是第 i 个阶段的主节点
12)　　主节点 v_i 广播它当前的值 w
13)　　if 接收 *propose*(a)的次数严格少于 $num-f$ 次 then
14)　　　$a=w$
15)　　end if
16)　end for

初始化阶段，CPoW 共识通过算法 1 实现了节点信用度的排名，通过算法 2 实现了节点排名的无差错分发，从而使每个节点获取自己的信用度排名，以便构建区块。

2) 构建区块

CPoW 共识机制的第二步是构建新的区块。当节点通过初始化阶段获取了自己的排名后，根据分片轮转模型获取自己的搜索空间，对区块头进行重复 SHA256 散列函数运算，根据搜索空间不断修改参数，直到散列运算的结果小于某一难度值。当某一节点验证出满足问题难度的目标值时，该节点立刻将所构成的区块发给它的所有相邻节点。这些相邻节点成功验证并接收这个新区块后，继续向全网传播此区块。当这个新区块被验证通过，每个节点都会将它加到自身节点的区块链副本中。当节点收到并验证了新的区块后，它们会放弃构建相同高度区块的计算，并立即转向计算下一个区块的工作。

3) 校验新区块

当新区块在网络中扩散时，每一个节点接收它之前，都需要进行一系列的验证，确保只有有效的区块才会在网络中传播。因为这些校验都是独立进行的，只有确保诚实的节点生成的区块才会被纳入区块链，从而获得奖励。反之，由于区块的无效性会导致节点失去奖励，并会将这种“不诚实行为”写入节点状态记录表，影响节点信用度。

当一个节点验证收到的新区块时，与标准验证流程清单一一核对，若没有通过验证，该区块将被拒绝，其标准一般如下。

① 验证提交新区块节点信用度的真实性。

② 区块的数据结构语法上有效。

③ 搜索空间属于记账节点需要验证的搜索空间。

④ 区块头的 hash 值小于目标难度。

⑤ 区块大小符合长度限制。

4) 区块链的组装

共识机制的最后一步是将区块装配至区块链的主链中。一旦某个节点验证通过了新的区块，它会尝试将新的区块连接到现有的区块链上，并将它们组装起来。

节点进行区块组装一般分为 3 种情况。第一种情况，由于区块的 hashPrevBlock 字段是该区块对其父区块的引用，如果该父区块是区块链的“顶点”，那么该区块就可以直接连接到区块链的父区块后。第二种情况，当主链出现了分支，即所谓的备用链，在这种情况下，节点会将新的区块添加到备用链，同时，节点会比较备用连和主链的长度，如果备用链长度更长，那么节点将收敛于备用链，这就意味着备用链成为新的主链，原来的主链变成备用链。第三种情况，如果节点没有在区块链上找到该区块的父区块，那么这个区块就是“孤块”。孤块会被保存到孤块池中，当其父区块出现并成功连接到区块链上，孤块才会被连接到父区块后，成为区块链的一部分。

随着越来越多的区块被加入区块链上，链上的工作量证明就更多，链的暂时性差异最终会得到解决。

由于 CPoW 共识机制的初始阶段节点太少，所以获得的节点状态数据较少。在这种情况下，由于节点状态数据缺乏，导致训练 BP 神经网络效果不佳，因此在 CPoW 共识机制开始运行前，会采集一些虚拟货币系统的节点状态数据来训练信用模型，同时，还要保证采集数据的全面性，以保障初始阶段模型的准确性，同时，需要将学习好的权值保存下来，以节省再次训练的时间。

4 安全性分析

4.1 区块链分叉

CPoW 共识因为解决 hash 难题的时间更短，所以比 PoW 算法更容易产生分叉，因此，本文在信用模型还考虑了区块链分叉这一因素，在一定程度上抑制了区块链的分叉。

当出现分叉后，节点会依据 2 个准则来判断主链，且这 2 个准则的优先级由高到低。第一准则：当出现 2 个或多个链分叉时，节点会优先选择在更长的链上进行“挖矿”，如图 3 所示。其中，区块 B、D 所在链等长，区块 E 所在的链更长，所以节点会选择以区块 E 作为父区块。第二准则：当出现等长的 2 个或多个链出现分叉时，节点会优先选择在信用度更高的节点产生的父区块上进行“挖矿”，如图 4 所示。其中，区块 B、C、D 等长，故节点会选择在信用度更高的节点产生的区块 C 上进行挖矿。

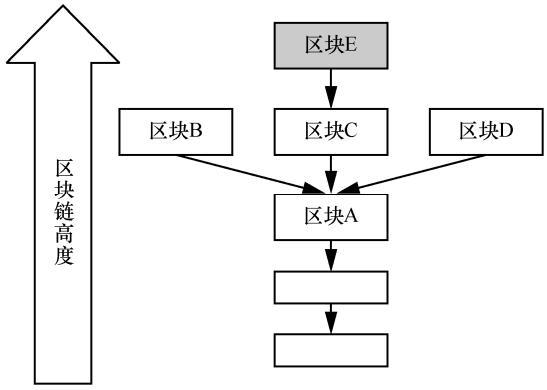


图 3 在更长链上进行“挖矿”

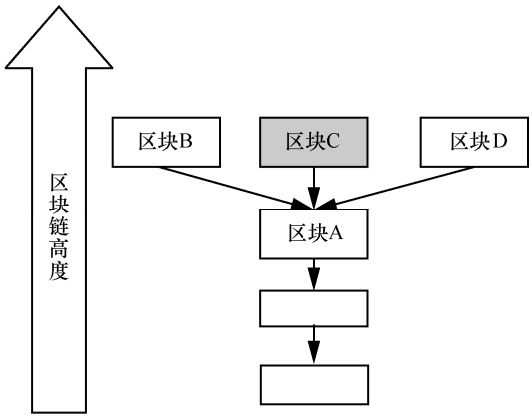


图 4 在信用度更高的节点产生的区块上进行“挖矿”

为了防止恶意节点不断进行区块分叉来实现“双花”或扩大区块链的暂时差异性的时间，CPoW 机制会将因产生新区块导致区块链分叉的节点记录在节点状态记录表中，通过这样的方法，使下一轮节点在信用度排名时降低其排名甚至是无法参与共识计算。由于区块链的分叉会导致一定程度的资源浪费，CPoW 机制还加入了全网心跳机制，通过不断地询问周围节点是否收到待验证的新区块，以此来降低产生区块链分叉的风险。

4.2　心跳机制

当节点获得自己的排名，通过分片轮转模型计算获得自己的搜索空间。值得注意的是，这里会存在因为节点故障、网络分区等原因而没有获得信用度排名的节点，或节点断电离线导致无法进行搜索空间的验证计算。对于以上这些情况，CPoW 共识中出现的非常少，这是由于该共识机制中的信用度模型保证了更多的好节点会参与共识机制。

即使有这样的保证，CPoW 共识还提供了容错机制，即全网心跳机制，具体如下。排名为 r 的节点每隔 N s 向全网发送一个心跳消息（这种心跳消息可以是询问是否收到新区块），收到心跳消息的节点会回复心跳消息给 r 节点。如果排名为 r+1 的节点在 M s（M>N）内都没有收到 r 节点的心跳消息，即心跳超时，那么 r+1 的节点会向全网发送询问是否收到 r 节点心跳消息，等收到 num−f 的答复（收到的不需要答复，没有收到的就会答复）后，就会代替 r 节点验证它的搜索空间，同时网络上任何监测到心跳超时的节点都会在状态记录表中进行日志记录。另外，r+1 节点还会检查 r−1 节点的心跳是否正常，如果不正常则进行类似的操作，同时询问 r−2 节点心跳是否正常，直至检查到心跳正常的节点为止。

4.3　共识攻击

共识机制依赖于这样一个前提：绝大多数的矿工因为考虑自己利益的最大化，所以都会通过诚实地挖矿来获取奖励，从而维持整个系统安全。单个矿工因自身算力的限制，理论上实施欺骗或破坏的难度很大。然而，当一群拥有了整个系统 51%算力的矿工们通过合作攻击共识机制，这样就会破坏系统的安全性和可靠性，产生 51%攻击。

当矿工们发动 51%攻击，CPoW 共识会比虚拟货币系统的 PoW 共识代价更高，在虚拟货币系统的 PoW 共识中，当攻击者的算力达到 51%这个阈值时，其发起的攻击尝试几乎肯定会成功，而对于 CPoW 共识，影响 51%攻击的因素不是算力而是信用度，因为 BP 神经网络可以通过参数调整进一步弱化算力的权重。无论对于 PoW 共识还是 CPoW 共识来说，所谓的 51%攻击只是来说明节点能够验证的搜索空间的大小而已。对于 PoW 共识而言，51%攻击意味着该矿工或矿池可以验证 51%的搜索空间。对于 CPoW 共识算法，由于弱化算力的原因，其远不能达到能够验证 51%的搜索空间的程度。很显然，攻击者花费更大的代价才能实现 51%攻击。

4.4　抑制矿池的产生

虽然由于全网算力的急剧增加使单个矿工已经不可能占据全网 1%的算力，但是托管矿池的引入导致算力大量集中，同时带来了矿池操作者出于利益而施行攻击的风险。对于虚拟货币系统中的 PoW 共识来说，矿池操作者不仅能够控制候选块的生成，也能控制交易的填充，即矿池操作者拥有剔除特定交易或双重支付的权利。

但对于 CPoW 共识来说却不是这样。虽然矿池占据大量算力，但是其仍需要获取待验证的搜索空间，这就意味着它获取搜索空间本身也是需要花费时间的，相对于同等算力的多个单矿工节点，他们获取搜索空间是并发执行的，这无疑节省了时间成本。其次，占据大量算力的矿池由于信用模型的原因无法获得算力所对应的搜索空间，导致矿工会选择逃离矿池。最后，因为节点需要获取待验证的搜索空间，而成功在某一搜索空间上解决 hash 难题具有概率性，这大大降低了大算力对解决 hash 难题的影响。总之，一方面，CPoW 算法抑制了大型矿池和中心化节点的出现，避免了算力的集中化；另一方面，由于某一搜索空间恰好“命中”hash 难题具有随机性，这也降低了节点之间硬件恶性升级的可能性。

4.5　信用度攻击

对于节点信用度的攻击可以分为 2 种，一种是直接攻击，另一种是间接攻击。直接攻击是指节点直接篡改自己的信用度，然后提交给主节点进行排名，以此来获得更高的信用度排名和更大的搜索空间。间接攻击是指节点修改节点状态记录表，通过修改其他节点的记录参数降低其他节点的信用度；通过修改自己的记录参数提升自己的信用度。

对于直接攻击，由于 CPoW 共识机制的维护有 2 张记录表，一张是上一轮节点的状态记录表，另一张是本轮正在记录的状态记录表。由于每个节点在进行新区块验证时，会根据上一轮状态记录表和区块链交易数据并通过信用度模型再次核算节点信用度，如果验证失败，就会丢弃此区块并在本轮状态表中记录提供无效区块的节点。

对于间接攻击，它主要修改的是本轮状态记录表。为了预防这种攻击，CPoW 共识机制要求每一

个节点在向状态记录表中写入记录时，必须经过预写入过程。这种预写入过程是指节点先将数据写入状态记录表，但标记数据为预写入数据，同时向全网广播本条记录，收到此记录的节点查看自己状态记录表是否有此条记录，如果有则也广播这条记录，否则只是将此条记录预写入状态记录表，不做广播操作。当节点收到 $n-f$ 条记录，就会将预写入标志去掉。节点通过这种方式实现了状态记录表的一致性和不可篡改性。

5 实验仿真及分析

5.1 信用度模型的仿真及结果分析

CPoW 共识机制利用 BP 神经网络构建了节点信用模型来评估节点的信用度，然后利用信用度排名算法实现了各个节点的信用度排名，并利用算法 2 将排名结果分发至每一个参与共识的节点。CPoW 机制产生的一次排名结果将用于产生 1 440 个区块后，才会重新计算排名。由于评估信用度、计算排名和分发排名这 3 个过程与 hash 计算是并行进行的（除了首次运行共识协议需要等待排名结果，才能执行 hash 计算）。这些都使 CPoW 共识机制的初始化阶段的时间复杂度相对于产生 1 440 区块的时间可以忽略不计。所以在初始化阶段，本节主要关注 BP 神经网络是否能够准确地构建信用模型。

选取 110 个节点指标数据及其信用评估结果，根据信用评估模型的方法，对选取的指标数据首先进行标准化处理，将其转化为[0,1]的数据，以便于神经网络处理。每次用 100 个节点的数据作为训练数据，剩余 10 个节点的数据作为测试数据，共进行 380 次测试。实验时设定学习速率为 0.001，误差变化如图 5 所示。

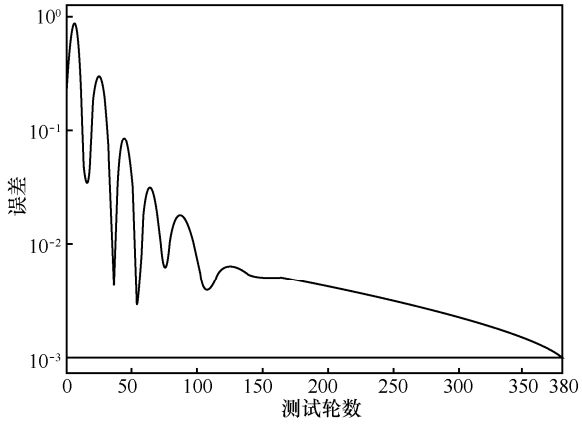


图 5 误差变化曲线

模型输出和目标输出之间的对比如图 6 所示。

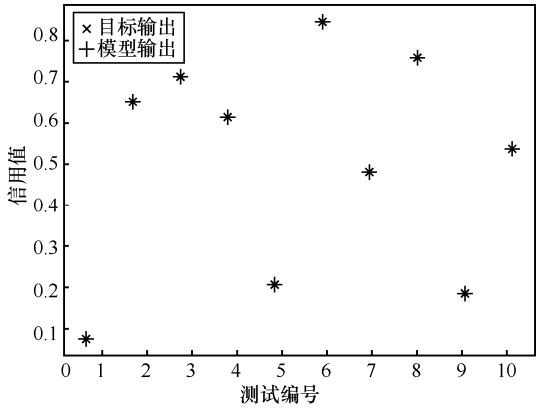


图 6 模型输出和目标输出的对比

如图 5 和图 6 所示，BP 神经网络构建的信用度模型具有非常好的准确性，但是其仍存在学习效率慢、网络的学习和记忆具有不稳定性等问题，有许多改进之处。

5.2 hash 计算的仿真及结果分析

对于 PoW 共识机制而言，随着难度值（二进制以 0 开头位数）的增加，hash 难题解决时间会不断增大。对于 CPoW 共识机制而言，同样如此。但由于 CPoW 共识机制将搜索空间进行了划分，使“挖矿”行为由一种竞争行为变成了一种合作行为，从而提高了“挖矿”效率，降低了资源消耗。所以本节主要对区块生成阶段进行仿真实验。

5.2.1 实验准备

此仿真实验主机有 12 台，CPU 为 Intel Core i5-7500U，内存为 8 GB，操作系统为 Window 10 企业版。实验选用 Python3.5 为主要编程语言，并使用其 Matplotlib-2.1.0rc1 模块实现数据的可视化。

5.2.2 仿真过程

通过 Python 语言模拟了一个简单的 PoW 算法，该算法通过不断调整 hash 问题的难度然后统计“挖矿”成功的时间。因为 PoW 算法是竞争“挖矿”，实验只需要在一台 CPU 为 i7-7500U 的主机上运行。

而对于 CPoW 算法，因为它是一个划分搜索空间的合作“挖矿”，所以除了核心的“挖矿”模块，还需要有通信模块和划分搜索空间模块。为了简化实验，仿真实验指定了各个节点的信用度排名。实验主要比较 CPoW 与 PoW 的算法效率和随机性，探索解决 hash 难题时间和难度值，节点个数的关系，以及节点解决 hash 难题个数的分布情况。

5.2.3　仿真结果

PoW 共识与 CPoW 共识解决难题的时间对比如图 7 所示。其中，横轴代表 hash 难题的难度值，纵轴代表解决 hash 难题的时间。由图 7 可知，在难度值相同时，CPoW 共识机制解决 hash 难题的时间明显小于 PoW 共识机制。并且，3 个节点合作“挖矿”的效率会好于 2 个节点“挖矿”的效率，即节点越多，“挖矿”效率越高，图 8 和图 9 更能够说明这一点。但是，在图 7 中仍出现了 2 个节点合作“挖矿”所用时间小于 3 个节点合作“挖矿”的情况，这种情况约占考察总数的 14.3%，这是由于合理 hash 值的产生具有不确定性。

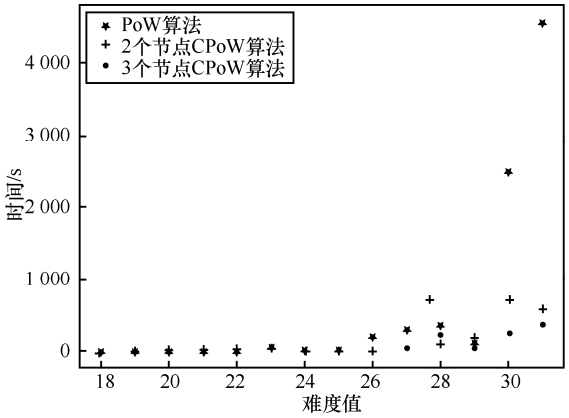


图 7　PoW 共识与 CPoW 共识解决难题时间对比

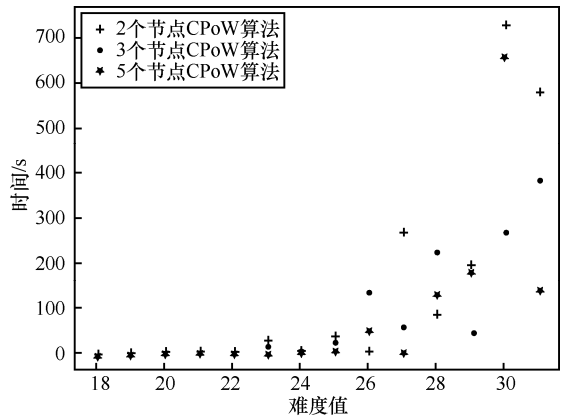


图 8　节点个数为 2、3、5 时对 CPoW 共识解决 hash 难题的影响

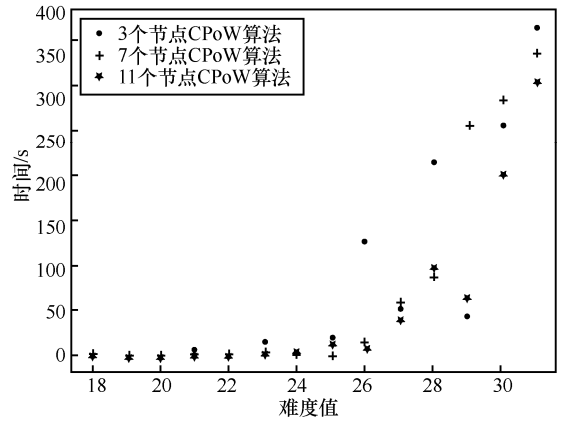


图 9　节点个数为 3、7、11 时对 CPoW 共识解决 hash 难题的影响

图 10 和图 11 其难度值分别为 29 和 30，通过增加节点数量，可以发现随着节点数量的增加，hash 难题的解决时间总体上呈递减的趋势。图 10 和图 11 中出现了与预期不符的波动节点，这是由于在寻找合理 hash 值的过程中，CPoW 共识算法良好的随机性，导致在寻找 nNonce 值时具有不确定性，致使在“挖矿”时间上出现了波动，但其整体呈下降趋势。

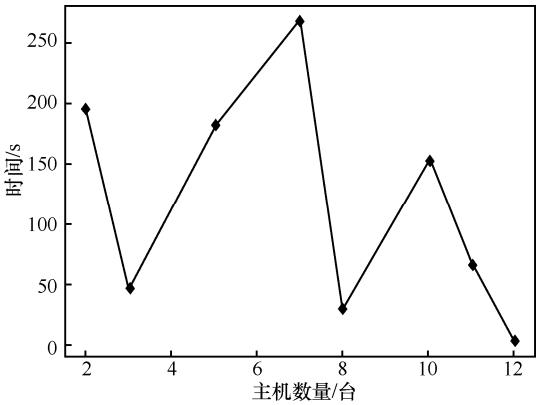


图 10　CPoW 共识解决 hash 难题的时间趋势(1)

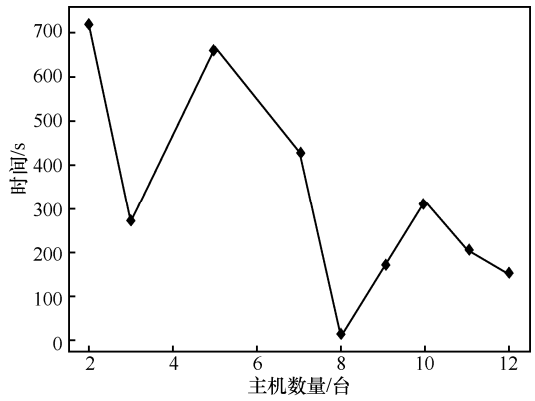


图 11　CPoW 共识解决 hash 难题的时间趋势(2)

实验还在难度值相同的情况下，通过改变区块填充内容，尝试“挖矿”100 次，分别统计了 12 个节点和 11 个节点解决 hash 难题的次数的分布情况，如表 4 所示和表 5 所示。从表 4 和表 5 可以发现，寻找合理 hash 值是一种不确定事件，但是排名的高低（排名值越大，排名越高）对解决 hash 难题还是有影响的，如 2 个表中都出现了低排名解决 hash 难题个数偏少的情况。但这也不意味着最高的排名

就一定能够更多地解决 hash 难题，如表 5 所示。

表 4　　12 个节点解决 hash 难题

节点排名	解决 hash 难题次数	所占比例
1	6	6%
2	3	3%
3	7	7%
4	6	6%
5	28	28%
6	7	7%
7	4	4%
8	1	1%
9	14	14%
10	3	3%
11	7	7%
12	14	14%

表 5　　11 个节点解决 hash 难题

节点排名	解决 hash 难题次数	所占比例
1	2	2%
2	4	4%
3	8	8%
4	12	12%
5	10	10%
6	13	13%
7	6	6%
8	18	18%
9	15	15%
10	5	5%
11	7	7%

综上所述，CPoW 共识机制大大提高了“挖矿”的效率，随着参与共识的节点数的增加，其效率提高越明显。同时，新的 CPoW 共识机制在选择记账节点的问题上具有很好的随机性，但仍存在信用度高低对“挖矿”成功的影响。仿真实验也从侧面反映了解决相同难度的 hash 难题时，由于时间的减少，其资源消耗会减少。

5.3　性能分析

本节主要从寻找合理 hash 值时间和资源消耗这 2 个角度对 CPoW 共识机制进行性能分析。

节点执行 CPoW 共识寻找合理 hash 值的过程如图 12 所示。其中，虚线框代表分片，实线框代表轮次，其中，五角星代表合理 hash 位置（忽略其具体位置，默认其在这个搜索空间的最后），它位于第 n 轮排名为 a_i 的节点的搜索空间中。

假设用节点的平均算力 C 代替网络中节点的算力，用 T' 表示获取搜索空间的时间，由于节点信用度的排名和分发都是与 CPoW 共识的 hash 计算并发执行的，因此时间可以忽略，则在 CPoW 共识下寻找到合理 hash 值的时间为

$$h_1 = \frac{2a \cdot n \cdot runit}{(1+num)num \cdot C} + nT' \tag{15}$$

而对于 PoW 共识而言，寻找合理 hash 值的时间为

$$h_2 = \frac{runit \cdot num(n-1)(1+num) + a_i(a_i+1)runit}{(1+num)num \cdot C} \tag{16}$$

通过比较可以发现，$h_1 < h_2$。

在资源消耗方面，假设进行一次 hash 运算需要消耗的资源为 s_c，获取一次搜索空间需要消耗的资源为 s_n，进行一次信用度排名和分发需要消耗的资源为 s_p，则 CPoW 共识所消耗的资源为

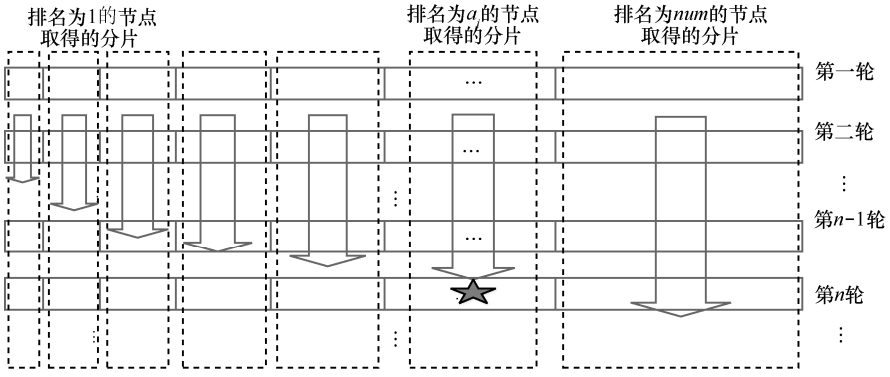


图 12　寻找合理 hash 值的过程

$$w_1' = \frac{2a \cdot n \cdot s_c \cdot runit}{(1+num)num} + ns_n + s_p \quad (17)$$

而对于 PoW 共识而言，所消耗的资源为

$$w_2 = \frac{runit \cdot num(n-1)(1+num) + a_i(a_i+1)runit}{(1+num)num} \cdot s_c \quad (18)$$

在寻找一次合理 hash 值的情况下，w_1 和 w_2 无法进行大小的比较，因为无法确定 s_n 和 s_c 的大小。但是当在成功寻找了 1 440 个合理 hash 值的情况下（产生 1 440 个区块后，系统会提前更新信用度排名），并假设在这 1 440 次计算中合理 hash 值的位置不变，则 CPoW 共识下所消耗的资源为

$$w_1 = \frac{2\,880a \cdot n \cdot s_c \cdot runit}{(1+num)num} + 1\,440n \cdot s_n + s_p \quad (19)$$

而对于 PoW 共识而言，所消耗的资源为

$$w_2' = 1\,440w_2 \quad (20)$$

通过比较可以发现，$w_1' < w_2'$ 。

5.4　适用性分析

目前，根据区块链开放等级的不同，区块链被分为 3 类：私有链、公有链、联盟链。PoW 共识机制就是经典的公有链共识算法，它很好地适应了公有链的发展，且设计精简。CPoW 共识机制主要包含信用度评估、信用度排名（算法 1）、排名分发（算法 2）和共识计算这 4 个主要算法，其在算法设计上较为复杂，但是它同样适用于公有链。但是，其区块链网络的规模（参与共识的节点数量）远少于 PoW 共识机制所适用的区块链网络规模。因为理论上 PoW 共识机制可以使参与共识节点的数量接近无穷大，而 CPoW 共识机制受限于信用度排名（算法 1）和排名分发（算法 2），虽然本文在算法流程设计上已经进行了优化，但本质上其算法执行时间仍与参与共识节点的数量相关。

综上所述，CPoW 共识算法适用公有链。其算法性能决定了它能够满足大多数区块链应用场景的共识计算，不仅适用于虚拟货币等电子货币系统，还适用于智能电网[19]、供应链系统、数据存储与溯源、食品安全等众多领域。

6　结束语

对于虚拟货币的 PoW 共识机制而言，寻找合理 hash 值是一种概率事件，占有更多的算力会有更大的概率成功“挖矿”。对于点点币的 PoS 共识机制而言，同样是寻找合理的 hash 值，其过程也是一个概率事件，拥有 coin age 的多少会等比例地降低“挖矿”难度，从而增加“挖矿”成功的概率。而对于 CPoW 共识而言，寻找合理 hash 值还是一个概率事件，拥有更高的信用度会获得更多的搜索空间，从而增加“挖矿”成功的概率。同时，某一节点获得的搜索空间恰好能够产生新区块，这也是一个概率事件，从而又给成功“挖矿”增加了随机性。

一方面，CPoW 共识机制降低了资源消耗，另一方面，信用排名的机制确保了节点们致力于获得更高的信用排名，而不是进行盲目的算力升级和恶意的区块分叉。总之，CPoW 共识机制能够消耗更少的计算资源产生一个区块，同时也使记账节点的产生更具随机性，很好地适应了公有链的发展。

此外，对于 PoS 共识机制而言，同样可以设计基于信用模型的共识算法，让其根据信用度的高低等比例地降低“挖矿”的难度，从而防止币龄攻击（save-up attack），提高共识机制的安全性。后续工作将对 PoS 共识机制及 PoS 共识的变种机制（如 DPoS）进行信用模型的构造和分析，从而设计更加高效的共识机制。

参考文献：

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[J]. Consulted, 2008.

[2] MCCONAGHY T, MARQUES R, MÜLLER A, et al. BigchainDB: a scalable blockchain database[R]. White Paper, BigChainDB, 2016.

[3] ZYSKIND G, NATHAN O A. Decentralizing privacy: using blockchain to protect personal data[C]// IEEE Security and Privacy Workshops. 2015:180-184.

[4] STOLZ D, WATTENHOFER R. Byzantine agreement with median validity[C]//19th International Conference on Priniciples of Distributed Systems (OPODIS). 2015.

[5] CASTRO M, LISKOV B. Practical Byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4): 398-461.

[6] CASTRO M, LISKOV B. Practical byzantine fault tolerance[C]//The Third Symposium on Operating Systems Design and Implementation. 1999: 173-186.

[7] 范捷, 易乐天, 舒继武. 拜占庭系统技术研究综述[J]. 软件学报, 2013, 24(6): 1346-1360.
FAN J, YI L T, SHU J W. Research on the technologies of Byzantine system[J]. Journal of Software, 2013, 24(6): 1346-1360.

[8] 唐长兵, 杨珍, 郑忠龙, 等. PoW 共识算法中的博弈困境分析与优

化[J]. 自动化学报, 2017, 43(9):1520-1531.
TANG C B, YANG Z, ZHENG Z L, et al. Game dilemma analysis and optimization of PoW consensus algorithm[J]. Acta Automatica Sinica, 2017, 43(9):1520-1531.

[9] LAMPORT L. Paxos made simple[J]. ACM Sigact News, 2001, 32(4): 18-25.

[10] KING S, NADAL S. PPCoin: peer-to-peer crypto-currency with proof-of-stake[J]. 2012.

[11] LARIMER D. Delegated proof-of-stake[R]. White Paper, 2014.

[12] BENTOV I, LEE C, MIZRAHI A, et al. Proof of activity: extending bitcoin0s proof of work via proof of stake[J]. ACM Sigmetrics Performance Evaluation Review, 2014, 42(3): 34-37.

[13] ROSENFELD M. Analysis of bitcoin pooled mining reward systems[J]. Computer Science, 2011.

[14] ANDREAS M. Antonopoulos. mastering bitcoin [M]. O'Reilly Media, 2014.

[15] DING S F, JIA W K, SU C Y, et al. An improved BP neural network algorithm based on factor analysis[J]. Journal of Convergence Information Technology, 2010, 5(4):103-108.

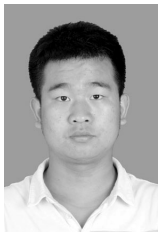
[16] MA Y X, WANG S G. The application of artificial neural network in the forecasting on incidence of a disease[C]//International Conference on Biomedical Engineering and Informatics. 2010:1269-1272.

[17] JIN W, LI Z J, WEI L S, et al. The improvements of BP neural network learning algorithm[C]// International Conference on Signal Processing Proceedings. 2002:1647-1649.

[18] BERMAN P, GARAY J A, PERRY K J. Towards optimal distributed consensus[C]//Symposium on Foundations of Computer Science. 1989:410-415.

[19] RAHBARI A N, OJHA U, ZHANG Z, et al. Incremental welfare consensus algorithm for cooperative distributed generation/demand response in smart grid[J]. IEEE Transactions on Smart Grid, 2017, 5(6):2836-2845.

[作者简介]



王缵（1992–），男，安徽安庆人，贵州大学硕士生，主要研究方向为信息安全、区块链应用与共识机制、机器学习。

田有亮（1982–），男，贵州盘县人，博士，贵州大学教授、博士生导师，主要研究方向为算法博弈论、密码学与安全协议、大数据安全与隐私保护等。

李秋贤（1992–），女，河南焦作人，贵州大学硕士生，主要研究方向为密码学与安全协议。

杨新欢（1993–），女，山西运城人，贵州大学硕士生，主要研究方向为信息安全、数据通信安全。