

基于声誉分数的改进 PBFT 共识算法

徐莹臣^{1,2} 田野^{1,2} 景莹^{1,2} 袁富江^{1,2}
XU Yingchen TIAN Ye JING Ying YUAN Fujiang

摘要

针对实用拜占庭容错 (practical byzantine fault tolerance, PBFT) 共识算法中存在的主节点选取随机、通信开销大、无奖惩机制等问题, 提出一种基于声誉分数的改进 PBFT 共识算法 (reputation scores byzantine fault-tolerant algorithm, FY-PBFT)。首先采用 Fisher - Yates 洗牌算法将节点划分至共识节点集和候选节点集, 同时引入声誉奖惩机制, 依照节点在共识过程中的表现动态计算声誉分数并依此调整节点集, 同时改变主节点的选取方式使可靠的节点当选主节点。仿真实验表明, FY-PBFT 共识算法在通信开销、共识时延等方面相较于 PBFT 共识算法有一定提升。

关键词

区块链; 共识算法; 实用拜占庭容错; Fisher - Yates 洗牌算法; 声誉分数

doi: 10.3969/j.issn.1672-9528.2025.03.043

0 引言

当今信息技术日新月异, 区块链技术以其去中心化、可追溯、防篡改等特点在科技发展以及社会进步的多个领域起到了积极正向的推进作用^[1-4], 而共识算法作为区块链的核心技术便得到了人们广泛的关注^[5]。常用的共识算法有工作量证明 (PoW)、权益证明 (PoS)、股份授权证明 (DPoS)、Raft、实用拜占庭容错 (practical byzantine fault tolerance, PBFT)。PoW 主要依托于 hash 算法构建工作量, 存在计算能力浪费, 大量资源被用于无意义的 Hash 计算和交易确认效率低的问题^[6], 但安全性较强。PoS 共识算法依靠节点所持有的权益来决定其是否拥有记账权, 但容易存在权益集中的问题, 对去中心化程度有一定的影响, 安全机制仍需完善, 但 PoS 相较于 PoW 可以节省超过 99% 的能源消耗^[7]。DPoS 由持币人投票选举出代表节点负责记账, 虽然存在一定的中心化风险, 但与 PoS 相比, DPoS 参与的共识节点的数量大幅减少, 从而缩短了区块链的出块时间^[8]。Raft 算法的关键问题是领导者选举和日志同步^[9], 其通信复杂度较低, 适用于相对可靠的网络环境, 但存在伪造日志的问题, 并且容错能力有限。PBFT 中无权益的抵押或竞争资源的消耗, 使得恶意节点的成本降低, PBFT 通过三阶段的投票机制排除恶意节点对共识的影响, 提供不超过 1/3 总节点数量的容错能力^[10], 但当前 PBFT 算法仍然有很多值得改进的地方, 如通信开销大, 主节点选取随机等问题最终导致算法的共识效率

较低, 为了提高共识效率, 大部分学者从主节点选取方式和共识一致性方面进行改进。

在主节点选取方面, 文献 [11] 给共识节点设置一系列活动状态使得节点通过状态转换在系统中拥有完整生命周期, 由此节点可以动态地加入和退出, 同时增加以最长链为选举原则的主节点选举过程, 使选举出的主节点可信度更高。文献 [12] 提出一种基于节点可靠性评估的改进拜占庭容错算法, 该方案引入节点基础配置评分机制及信誉评分机制, 并以此为依据将节点标记为 3 种不同的信任状态, 改进主节点的选取方式, 提高了系统的共识效率以及安全性。上述方法对主节点选取方式进行了改进, 虽然保证了主节点的可靠性, 但是随着节点数量的增加, 通信开销问题仍然面临挑战。在共识一致性方面, 文献 [13] 引入节点的自身状态和投票数, 并根据节点信誉值选择部分节点作为聚类中心运行 K-means 聚类算法, 实行组内、组间分开的方式进行共识以减轻节点的存储负担, 提高了系统的共识效率和安全性, 但存在聚类效果不稳定的情况。文献 [14] 提出了 DS-PBFT 算法, 优化一致性协议, 简化三阶段通信流程从而提高共识效率, 并提出节点评分分组机制, 提高了共识的效率, 但在网络不稳定、共识节点数量增多时共识效率仍面临挑战。

针对 PBFT 共识算法存在的问题, 本文提出改进方案。首先, 针对 PBFT 缺乏奖惩机制的问题, 提出基于声誉分数的奖惩机制以确保节点参与共识、投票的积极性; 其次, 针对大规模网络场景中 PBFT 通信开销大的问题, 使用 Fisher - Yates 洗牌算法将节点划分为共识、候选节点集, 选择一部分可靠的节点参与共识, 减少不必要的通信开销。最后, 改进主节点的选取方式, 提出声誉机制结合投票双重评估的方式选取可靠的主节点。

1. 太原师范学院计算机科学与技术学院 山西晋中 030619
2. 太原师范学院智能优化计算与区块链技术山西省重点实验室 山西晋中 030619

1 PBFT 共识算法

在分布式系统中，节点可能会出现各种意外情况，如网络故障或恶意攻击等。拜占庭容错算法旨在在此情况下确保系统仍然能够正常运行并达成一致的结果，它能够在最多有 f 个恶意节点存在的情况下保证分布式系统的一致性与可行性。在一个包含 N 个节点的系统中，当节点个数 N 与拜占庭节点个数 f 满足 $N \geq 3f+1$ 时，便可确保消息回复的正确性，从而达成共识。PBFT 共识算法在一致性协议、视图更换协议、检查点协议下运行。其中一致性协议是 PBFT 共识算法的核心协议，共分为 5 个阶段，分别是请求阶段、预准备阶段、准备阶段、确认阶段、回复阶段。执行过程如图 1 所示。

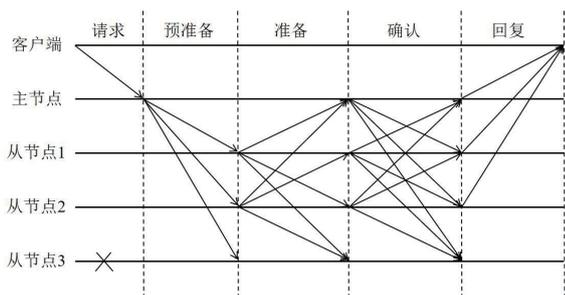


图 1 PBFT 算法共识流程

PBFT 共识算法的一致性协议基本执行过程如下：

- (1) 请求阶段：客户端向主节点发送包含操作内容的请求，主节点对请求进行序列号分配、签名等操作将其整合为预准备消息。
- (2) 预准备阶段：主节点将预准备消息广播给从节点，从节点对消息进行验证，若验证通过则将其记录于自己的日志中。
- (3) 准备阶段：从节点向其他节点发送准备消息表明已认可预准备消息，从节点收到 $2f+1$ 个相同序列号的准备消息时，便进入确认阶段。
- (4) 确认阶段：包括主节点在内的共识节点向其他节点广播确认消息，当接收到 $2f+1$ 个相同序列号的确认消息时，便执行请求中的操作。
- (5) 回复阶段：执行完请求的操作后，节点向客户端发送回复消息，客户端接收到 $f+1$ 个相同的回复消息时，便可确认消息已经被成功执行。

2 FY-PBFT 算法设计

本文提出一种基于声誉分数的改进 PBFT 共识算法，使用 Fisher-Yates 洗牌算法将所有节点公平地划分为两类：共识节点集和候选节点集。在首轮共识前，初始化所有节点的声誉分数为 60，并使用洗牌算法随机选出 $2/3$ 的加入共识节点集（若节点数量为奇数无法被整除，则向下取整保留整数），

另外 $1/3$ 的节点为候选节点，仅参与投票同时被动同步在共识节点集中达成一致的消息。在后续不断的共识过程中，根据节点在共识过程中的表现计算出每个节点的声誉分数，并对其进行降序排列，当节点的声誉分数低于共识节点 $2/3$ 的阈值时，将被移至候选节点集，同时在候选节点集中选出声誉分数高的节点加入到共识节点集。每轮共识后，选出声誉分数前 10% 的共识节点，所有节点对这些节点进行投票，目的是根据声誉分数即节点在共识中的表现结合投票的形式，动态的选举主节点，这种通过声誉分数和投票双重评估的方式能够选举出更加可靠的主节点。

2.1 Fisher-Yates 洗牌算法分组

本文在初始分组时，在一个节点数量为 N 的系统中使用 Fisher-Yates 洗牌算法随机挑选出 $2N/3$ 比例的节点加入到共识节点集参与共识，其他节点加入到候选节点集。后续随着共识算法的进行，会根据节点声誉分数的高低动态的将节点划入共识节点集或者候选节点集，分组步骤具体如下：

- (1) 将系统中的节点总数设为 N ，并设共识节点集 C 大小为 n_1 ，候选节点集 A 大小为 n_2 ，且 $n_1 + n_2 = N$ 。本文按照 $2/3$ 和 $1/3$ 的比例分别划分共识节点集和候选节点集，则 $n_1 = 2N/3$ ， $n_2 = N - n_1$ 。
- (2) 将所有节点按照加入系统的先后顺序从 1 到 N 进行编号，形成一个节点列表 $L = [1, 2, \dots, N]$ 。
- (3) 使用 Fisher-Yates 洗牌算法对节点列表 L 进行洗牌，打乱节点的顺序，以此保证所有节点被初始选中加入到共识节点集的机会是均等的，算法具有良好的随机性，提高系统的安全性。
- (4) 将洗牌后的节点列表 L 中的前 n_1 个节点作为共识节点集 C 。剩余的 n_2 个节点作为候选节点集 A 。
- (5) 在划分后的共识节点集 C 中，采取节点投票的形式选举出第一个主节点，至此，初始化分组结束。

2.2 节点声誉评分机制

声誉分数计算是 FY-PBFT 算法的关键部分，根据节点在共识过程中的表现进行评分，是节点动态划分集合、主节点选取的基础。

2.2.1 共识节点声誉分数评分公式

本文中共识节点声誉分数的更新由其在共识过程中的表现所决定，而候选节点暂时无法参与共识，故将共识节点和候选节点的声誉分数计算公式进行区分，给予候选节点增加信誉分数的机会，共识节点 i 的声誉分数评分方式为：

$$S_{i,k} = S_{i,k-1} + \Delta R + \theta \cdot W + \gamma \cdot P_i \quad (1)$$

式中： $S_{i,k}$ 是共识节点 i 完成当前轮次共识并更新的声誉分数； $S_{i,k-1}$ 是共识节点 i 完成上一轮次共识并更新的声誉分数；

θ 、 γ 是权重，本文中默认 $\theta=1$ 、 $\gamma=-2$ 。

下面将依次介绍公式的组成部分：

(1) 总共识完成率影响因子 ΔR ：总共识完成率影响因子能够反应节点在连续两轮参与共识的情况，当节点本轮未成功参与共识，则 ΔR 为负数，扣除一定的声誉分数。当节点本轮成功参与共识，则 ΔR 为正数，即给予一定的声誉分数奖励。且 $\ln(1+x)$ 函数的增速逐渐趋于平稳，即函数一开始时增速较高，这对于低完成率的节点，其成功完成一次共识所获得的声誉分数奖励大于高完成率节点成功完成一次共识所获得的声誉分数奖励。目的在于激励共识完成率较低的节点提高其参与共识的积极性，提高系统的整体参与度和公平性。

$$\Delta R = R_{i,k} - R_{i,k-1} = \ln\left(1 + \frac{C_{i,k}}{C_k}\right) - \ln\left(1 + \frac{C_{i,k-1}}{C_{k-1}}\right) \quad (2)$$

式中： $R_{i,k}$ 、 $R_{i,k-1}$ 分别是当前轮次和上一轮次的共识完成率影响因子； C_k 、 C_{k-1} 分别是系统到第 k 和第 $k-1$ 轮总共完成的共识轮次数； $C_{i,k}$ 、 $C_{i,k-1}$ 分别是节点 i 截至第 k 轮和第 $k-1$ 轮成功完成的共识轮次数。 ΔR 是总共识完成率影响因子。

(2) 共识惩罚评分 P_i ：共识节点集中的节点在参与共识时，若因宕机或恶意行为等未成功参与共识，则扣除其声誉分数作为惩罚，并且当节点未成功完成共识的累计次数增多时，对其惩罚力度也逐渐递增。

$$P_i(x) = P_0 + \alpha \cdot \sqrt[3]{x} \quad (3)$$

式中： $P_i(x)$ 是节点 i 第 x 次未成功参与共识的扣分值； P_0 是基础扣分值，本文中为 0； α 为影响分数扣除的系数，本文中为 1； x 是当前轮次节点未成功参加共识的累计次数。

(3) 投票率影响因子 V_i ：指的是某节点在选取主节点时参与投票的次数与系统总发起的投票次数的比值，在共识节点集中，投票率与成功参与共识所得的信誉分数挂钩。候选节点集中根据投票次数进行分数的更新，能够提高所有节点正确行使投票权的积极性。

$$V_i = \frac{N_v}{10 \cdot N_t} \quad (4)$$

式中： N_v 表示节点 i 参与投票的次数； N_t 表示系统总共发起投票的次数。

(4) 共识奖励评分 W ： W 是给予成功完成共识的节点的奖励，根据节点的累计声誉分数将节点的划分为 3 个区间，同样完成一次共识，信誉分值高的节点加分少于信誉分数低的节点，目的在于降低“强者恒强”效应，避免信誉分数较低的节点难以追赶上来，致使节点丧失参与共识和投票的积极性，通过更高的加分才能使得声誉分数低的节点看到提升声誉分数以赶超其他节点的机会，从而积极地参与共识和投票。

$$W = \begin{cases} a \cdot (1 + V_i), & 90 < S_{i,k-1} \\ b \cdot (1 + V_i), & 70 < S_{i,k-1} \leq 90 \\ c \cdot (1 + V_i), & S_{i,k-1} \leq 70 \end{cases} \quad (5)$$

式中： a 、 b 、 c 表示不同的比例，且满足 $0 < a < b < c$ ，在本文中默认 $a=0.5$ 、 $b=1$ 、 $c=2$ 。

2.2.2 候选节点声誉分数评分公式

当候选节点在每轮主节点选取时投出自己的支持票，则被认为是投票成功，给予一定的声誉分数奖励，否则扣除一定的声誉分数作为惩罚。

$$D_{i,k} = D_{i,k-1} + U; U = \begin{cases} d, & \text{节点本轮投票成功} \\ -2d, & \text{节点本轮未投票} \end{cases} \quad (6)$$

式中： $D_{i,k}$ 、 $D_{i,k-1}$ 分别是节点 i 在当前轮次和上一轮次共识结束后更新的声誉分数； U 是候选节点声誉分数更新因子，当节点本轮成功参与投票则给予一定的声誉分数 d 作为奖励，当节点本轮未投票则扣除 $2d$ 的声誉分数作为惩罚，本文默认 d 为 0.5。

2.3 主节点选取方式

在 PBFT 共识算法中，所有节点均参与共识过程且均有当选主节点的机会，而当系统中的拜占庭节点当选主节点后，因无法完成共识而引起视图更换协议更换主节点后，此拜占庭节点依然有机会再次当选主节点，这增加了系统的通信开销，降低了共识的效率同时破坏了系统的安全性。因此本文对主节点的选取做出改进。随着共识过程的进行，节点的声誉分数会根据其在共识过程中的表现动态变化，每轮共识完成后，会按照节点的声誉分数进行排名，排名在前 10% 的共识节点为高声誉分数节点，所有节点对高声誉分数的共识节点进行投票，并按照投票的高低选举出主节点（当前 10% 的高声誉分数节点有多个节点得票数相同且为最高票数时，则此时按照节点声誉分数的高低确定主节点），其中节点 i 的得票数量计算方式为：

$$H_i = \sum_{L=1}^N S_L; S_L = \begin{cases} 1, & \text{节点 } L \text{ 支持节点 } i \\ 0, & \text{节点 } L \text{ 不支持节点 } i \end{cases} \quad (7)$$

本文选取主节点的方式并不是单一的根据信誉分数的高低，而是结合投票的方式，对比于只按照最高信誉分数直接选取主节点的方式，可以打破信誉分数最高节点中心化的垄断，让更多的高声誉分数的节点获得当选主节点的机会，减少某个节点长期主导共识的风险，也不会单一的依赖信誉机制，增加了通过投票动态调整的机会，使主节点的选取更加公平、灵活。

3 实验与分析

本文所提改进算法仿真实验使用设备的处理器为 13th Gen Intel(R) Core(TM)i9-13900K 3.00 GHz，操作系统为 Win-

dows11, 机带 RAW 为 128 GB, 使用 Python3.8 进行代码的编写以及实现。本节通过对比基于声誉分数的改进拜占庭容错算法和传统 PBFT 算法在通信开销、共识实验两部分的实验差异, 验证本文的改进效果。

3.1 通信开销

通信开销的大小是判断共识算法效率高低的一个重要指标, 指节点之间为了达成共识所需的消息传递量。PBFT 算法的通信过程包括 3 个阶段, 假设系统中有 N 个节点, 在预准备阶段, 主节点将预准备消息广播给 $N-1$ 个从节点, 因此此阶段为单向发送消息, 故通信开销为 $N-1$ 。在准备阶段, $N-1$ 个从节点相互发送准备消息, 这其中也包括从节点向主节点发送消息, 此阶段的通信次数为 $(N-1)^2$ 在确认阶段, 系统中的 N 个节点相互发送确认消息, 此阶段的通信次数为 $N \cdot (N-1)$ 。最终得出 PBFT 算法的通信开销 C 为:

$$C = N - 1 + (N - 1) \cdot (N - 1) + N \cdot (N - 1) = 2N^2 - 2N \quad (8)$$

对于本文提出的 FY-PBFT 算法的通信过程同样需要这 3 个阶段。根据方案的设定, 在不断地共识过程中, 根据节点的声誉分数将排名在前 $2N/3$ 的节点划入共识节点集, 参与共识的过程, 其余节点划入候选节点集, 不参与共识的过程, 只是被动的同步在共识节点集中达成一致的消息。因此, 本文在预准备阶段的通信次数为 $(2N/3)-1$, 在准备阶段的通信次数为 $[(2N/3)-1] \times [(2N/3)-1]$, 在确认阶段的通信次数为 $(2N/3) \times [(2N/3)-1]$ 。故得出 FY-PBFT 总的通信开销 C' 为:

$$C' = \frac{8}{9}N^2 - \frac{4}{3}N \quad (9)$$

图 2 为给定节点的情况下本算法与 PBFT 算法通信次数的对比图。

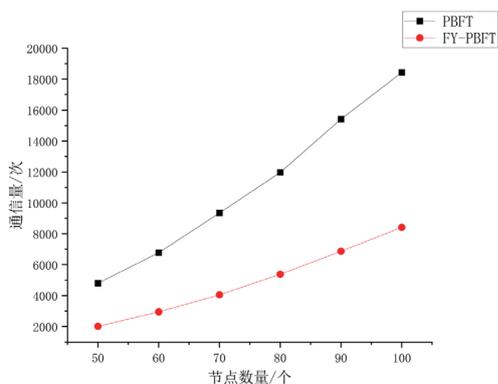


图 2 通信次数对比

图 2 所示在相同节点的情况下 FY-PBFT 算法的通信次数明显低于 PBFT 算法, 且随着节点数量的增加, 两算法之间通信次数的差距越大, 由此可知本算法能够有效减少系统的通信开销。

3.2 共识时延

共识时延是衡量系统性能的重要指标, 它的高低能够直接性的影响系统的效率和用户的体验。共识时延指的是从客户端发送请求到达成请求所花费的时间, 系统中节点的共识时延越低, 则系统达成共识的速度越快。共识时延的计算如下列公式所示。且在给定节点的情况下 FY-PBFT 算法与 PBFT 算法共识时延结果的对比如图 3 所示。

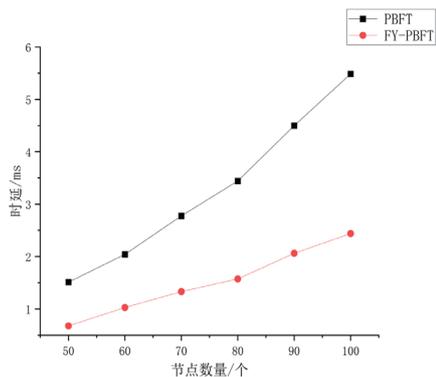


图 3 共识时延对比

$$T = T_F - T_R \quad (10)$$

式中: T_F 表示客户端收到 $f+1$ 个内容相同的回复消息的时间, 此时客户端的请求已被成功执行; T_R 表示的是客户端发送请求给主节点的时间。

当系统中的节点数量分别为 50、60、70、80、90、100 时, 分别对其做 50 次重复实验得出平均值作为实验结果, 最终结果如图 3 所示: 当两算法节点数相同时, PBFT 算法的共识时延明显高于 FY-PBFT 算法的共识时延且随着节点数量的增多这种差距愈加明显, 这是由于 FY-PBFT 算法对主节点、共识节点选取方式的优化降低了视图更换的概率且通过节点的声誉分数选取一部分可靠的节点参与共识, 减少了一部分不必要的通信开销, 从而降低了系统的共识时延。

3.3 节点声誉分数

本部分通过展示系统中 3 个共识节点和 2 个候选节点在 8 轮次的共识过程中其声誉分数的变化, 并以此来验证本文所提声誉分数的合理性和可行性。初始设定时, 节点 1、节点 2 和节点 3 为共识节点, 节点 4、节点 5 为候选节点。其中节点 1 积极参与共识和投票过程, 节点 2 较积极的参与共识和投票过程, 节点 3 消极参与共识但积极参与投票过程。节点 4 积极参与投票, 节点 5 在投票过程中没有有效使用投票权。其在 8 轮次共识过程中声誉分数变化如图 4 所示。由图 4 可知, 节点 1 积极的参与共识与投票过程, 其声誉分数呈上升趋势, 但因本声誉机制在设定时为避免“强者恒强”的现象出现, 故其在突破 70 分以后, 分数的增长速度明显下滑。节点 2 较为积极的参与共识与投票过程, 但能够较为

明显的看出其在第3轮次共识时未成功参与共识而导致分数下降。节点3在初始设定时为共识节点，但因其在第2轮次的共识过程中未成功参与共识，导致分数低于共识节点集的最低阈值故而被踢出共识节点集，在加入到候选节点集后，节点3积极地参与投票过程故分数也呈现较为缓慢的增长趋势。节点4积极地参与投票过程，在第2轮次共识过程结束后，其声誉分数赶超节点3，故而加入到共识节点集参与共识，因其在后续的共识过程中较为积极，故分数呈现总体上升趋势。节点5在投票过程中未能有效使用投票权，根据声誉机制其分数呈现总体下降趋势，若其持续此状态，则难有机会加入到共识节点集中参与共识过程。

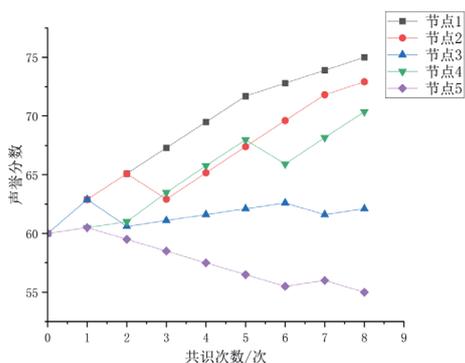


图4 节点声誉分数变化图

4 结语

通过分析 PBFT 共识算法，发现其存在所有节点均参与共识、主节点选取随机并且对恶意节点不进行惩罚等问题，提出了 FY-PBFT 共识算法，本文使用洗牌算法对节点进行分组，并引入声誉奖惩机制，根据节点的历史表现计算出节点的声誉分数，根据声誉分数的高低将节点划入不同的组内，限制了一部分节点参与共识的权利。优化了主节点的选取方式，通过声誉分数以及投票双重评估选取更加安全可靠的主节点，降低了通信复杂度，提高了系统的共识效率。仿真实验表明，相较于 PBFT 算法 PY-PBFT 算法的共识时延更低，通信次数更少且安全性更高。

参考文献:

[1] GADE D S, AITHAL S. Blockchain technology: a driving force in smart cities development[J]. International journal of applied engineering and management letters (IJAEML), 2020, 4(2): 237-252.

[2] GILLPATRICK T, BOĀA S, ALDANMAZ O. How can blockchain contribute to developing country economies? aliterature review on application areas[J]. Economics, 2022, 10(1): 105-128.

[3] SANGEETHA S B, KRISHNA P K. Blockchain based privacy and security across cloud in electric vehicle application for sustainable development in industry, innovation and infrastructure[J]. International journal of management, technology and social sciences (IJMTS), 2022, 7(2): 347-358.

[4] NGUYEN Q K, DANG Q V. Blockchain technology for the advancement of the future[C/OL]//2018 4th International Conference on Green Technology and Sustainable Development (GTSD). Piscataway: IEEE, 2018[2024-05-03]. <https://ieeexplore.ieee.org/document/8595577>.

[5] YUSOFF J, MOHAMAD Z, ANUAR M. A review: consensus algorithms on blockchain[J]. Journal of computer and communications, 2022, 10(9): 37-50.

[6] 武岳, 李军祥. 区块链共识算法演进过程 [J]. 计算机应用研究, 2020, 37(7): 2097-2103.

[7] 刘艺华, 陈康. 区块链共识机制新进展 [J]. 计算机应用研究, 2020, 37(S2): 6-11.

[8] 易黎, 卢新宇, 汤鲲, 等. 区块链共识算法研究综述 [J]. 电子设计工程, 2024, 32(6): 161-170.

[9] 姜义, 吕荣镇. 区块链共识算法综述 [J]. 佳木斯大学学报 (自然科学版), 2021, 39(2): 132-137.

[10] 靳世雄, 张潇丹, 葛敬国, 等. 区块链共识算法研究综述 [J]. 信息安全学报, 2021, 6(2): 85-100.

[11] 甘俊, 李强, 陈子豪, 等. 区块链实用拜占庭容错共识算法的改进 [J]. 计算机应用, 2019, 39(7): 2148-2155.

[12] 唐宏, 刘双, 酒英豪, 等. 实用拜占庭容错算法的改进研究 [J]. 计算机工程与应用, 2022, 58(9): 144-150.

[13] 胡继圆, 于璨. 基于信誉机制分组的改进 PBFT 算法 [J]. 湖北民族大学学报 (自然科学版), 2023, 41(1): 85-89.

[14] 沈学利, 李欣儒. 基于节点动态评分机制的分组共识算法 [J]. 计算机应用研究, 2024, 41(4): 989-994.

【作者简介】

徐莹臣 (1999—), 男, 山东淄博人, 硕士研究生, 研究方向: 区块链, email: 17615351517@163.com。

田野 (1980—), 女, 山西太原人, 博士, 副教授、硕导, 研究方向: 网络信息与安全、区块链。

景莹 (1999—), 女, 山西运城人, 硕士研究生, 研究方向: 深度学习。

袁富江 (1998—) 男, 四川宜宾人, 硕士研究生, 研究方向: 区块链。

(收稿日期: 2024-11-25)