CAGE

Consensus Algorithm Genetically Encouraged

Kamil Maka

Dept. of Computer Science

Middlesex University

Hendon

This thesis is submitted for the degree of

Master of Research

April 2022

Declaration

I, Kamil Maka, declare that this thesis titled, "CAGE" and the work presented in it are my own. I confirm that:

- This work was done wholly for a degree at Middlesex University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Middlesex University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: K. Maka Date: May 2022

Abstract

Blockchain Technology has gone beyond just cryptocurrency. There is a growing need not only for development in blockchain technology to support the needs of Web3.0, but also a need for research into Blockchain Technology. One of the fundamental components of Blockchain Technology is the consensus algorithms used to i) select a node in the network responsible for providing a block added to the blockchain; and, ii) verify that block and ensure trust between the nodes within the system. This research proposes a newly developed consensus algorithm for Blockchain Technology.

This research draws on inspiration from nature and the field of evolutionary computation, and selection methods in particular. The selection method is a mixture of Darwinianism and Fatigue-based systems, used in many evolutionary algorithms. This selection method is applied successfully as a consensus algorithm in a Blockchain Technology Systems. The proposed consensus algorithm is called Consensus Algorithm Genetically Encouraged, or CAGE for short.

An experimental framework was developed in which to test CAGE fairly. In this experimental framework CAGE was then tested and compared to another similar consensus algorithm, Proof-of-Elapsed-Time (PoET), many times. Results and analysis show that as the number of nodes in a blochchain technology increase, CAGE becomes more efficient in latency and throughput of block pro-

duction. Analysis showed that the node distribution of CAGE was not as even as PoET. Some modifications to the algorithm were made and the tests re-run. This proved more successful and improved the distribution of node selection whilst having no effect on throughput and latency. There are some reasons why CAGE outperforms PoET, which are mentioned in the analysis and results chapters.

In summary, this research developed a newly proposed consensus algorithm, CAGE, inspired by the selection methods used in evolutionary computation. CAGE was then tested many times and results show that as the number of nodes in the blockchain technology system increases CAGE outperforms PoET in terms of latency and throughput.

Acknowledgements

I would like to thank the never ending support of my family and friends.

I want to thank my brother for helping me arrange a time for the research and study, by helping out with my kids and around the house.

The completion of this study wouldn't be possible if not for my parents who set challenging standards to live up to and provided me with mental and emotional support not to give up and finish what I started.

I am also thankful to Dr Ian Mitchell for his patience, insight, and genuine desire to help me.

Last, but not least, I would like to thank my wife for her support. She berated me when I wanted to give up, encouraged me on my lazy days, helped me with organising time for the research, and supported me through a hard time when looking for a new job while at the same time working towards the degree.

Thank you all.

Contents

Chapter 1

Introduction

1.1 Introduction to Blockchain Technology

The first implementation of Blockchain Technology was Bitcoin [37]. What emerge from the embryonic stages in 2008 could never have been imagined by its inventors. Blockchain Technology, BCT, is the foundations for Web3.0 [2, ch.1] – for further definitions of Web3.0 see [29]. It is predicted in some text that BCT will become the next General Purpose Technology [20] and therefore, potentially have a positive impact on changing the economy [19]. However, what are Blockchains? To answer this question there is a need to dispel any prejudices, myths and misinformation that comes with BCT. The first myth is that Bitcoin≡BCT, in fact this can be extended to all cryptocurrencies, Cryptocurrencies≡BCT. So, conversely this means that cryptocurrency≠BCT. More accurately Bitcoin is one of many cryptocurrencies and are a subset of BCT, so to start to answer our question, $Cryptocurrencies \subset BCT$.

What are the other members of the BCT set? These are tokenless BCT that do not rely on cryptocurrencies and are often used to exploit the tamper-

resistant properties of BCT for auditing, e.g., [30, 33, 34, 35]. BCT such as Hyperledger [21, 22] grant developers to implement private and permissioned networks that do not rely on cryptocurrency.

So back to the original question, what are Blockchains? Leaving the technical definitions to one side, BCT is a collection of technologies that exploits tamper-resistant properties, promotes trust, and provides solutions in a secure and decentralised manner.

1.2 Motivation

With any success there follows myths and misinformation. Bitcoin is currently the most successful cryptocurrency and comes with one major disadvantage, energy consumption and sustainability [49]. In 2014 energy consumption was estimated at 10GW and is comparable to a small country [40], recent figures put this down to 10TwH per annum, which is equivalent to the consumption of 5 small countries [15]. This energy consumption is often referred to as the Achilles Heel of Bitcoin and 99% of this is caused by the mining and is a consequence of the chosen consensus algorithm, Proof-of-Work, PoW.

This energy consumption is a problem and can be learned from. In the development of CAGE the complexity of the CA should be sustainable or at least have lower energy consumption. Essentially, the energy consumed by the proposed CA should not be deemed a disadvantage.

There is one important distinction to make at this point. Whilst CAs take up huge energy resources in permissionless BCT, this is in complete contrast to permissioned BCT. This thesis looks at the latter permissioned BCT. Therefore, it is imperative that the performance of the proposed CA should be comparative or better than a permissioned CA, such as PoET.

This thesis aims to look at biologically inspired algorithms as an alternative

consensus algorithm. The motivation is to find sustainable consensus algorithms. This may be restrictive and require some context and scoping, which are mentioned in the rest of this section.

1.3 Background

Blockchain Technology, BCT, is built on three main tenets:

1. Encryption: using one-way hash algorithms to create a Blockchain and the ability to communicate and ensure confidentiality, integrity, authenticity and non-repudiation using asymmetric key encryption;

2. Decentralisation: using peer-to-peer (P2P) networks between nodes to facilitate a decentralised approach to management of the creation of blocks; and,

3. Consensus: safeguarding and ensuring the blocks created and the information within are to be trusted.

BCT generates copies of an append-only ledger that is stored on each node within the system. This combined with the consensus algorithm (CA) promotes trust in a trustless system. CAs safeguard the stability of the systems and ensures trust in a system, where trust between all nodes of the network would otherwise be near impossible. Fundamental modifications can be made in BCT, e.g., changing encryption techniques and using different hashing algorithms, but these changes are normally based on some technical or proprietary reason and result in very little difference in performance. The one change that can have the biggest effect on performance is the consensus algorithm. Choosing the right CA, for the right blockchain system is important. Proof-of-work requires effort and can often result in power-consuming mining nodes. PoW would often be deemed inappropriate for some blockchain systems, where this level of trust is

not required. Whilst, Proof-of-Elapse-Time, PoET, could compromise the trust between nodes in a permissionless blockchain system.

So, some gains in performance could be made by choosing different encryption and decentralisation techniques but it is generally agreed that the biggest gain in performance in blockchain systems would be the selection of the CA.

This research will attempt to create a consensus algorithm that encompasses selection methods used in evolutionary computing at its core for a fair distribution of authority to add new blocks to the blockchain. The selection of the node will be based on the node's fitness and its fatigue.

1.3.1 Scope

The danger of merging two subjects is knowing when to stop researching. The research on BCT will focus on canonical BCT and permissioned blockchain. The experimental framework will therefore only look at appropriate CAs. So, comparing CAs PoET and PoW would be unfair since the two CAs are used predominantly in permissioned and permissionless and are fault tolerant and PBFT, respectively. The research on evolutionary computation, another huge area of research, will focus on selection methods and how these can be applied to CAs.

1.4 Aims

The overall aim of this thesis is to develop a CA for a permissioned BCT that is inspired by selection methods found in genetic algorithms, the proposed acronym for this CA is **CAGE**, Consensus Algorithm Genetically Encouraged. The aims of this research are twofold:

1. develop new evolutionary inspired consensus algorithm, using selection

methods usually found in genetic algorithms; and,

2. set up a series of BCT experiments that will compare the proposed CAGE and analyse the results for any improvements in performance, focussing on throughput, latency and the distribution of node selection.

1.4.1 Objectives

The success of the project can be determined by the fulfilment of the following objectives:

- Research blockchain and selection methods in Genetic Algorithms
- Merge two establish technologies: namely selection algorithms in evolutionary computation and CAs in blockchain
- Software development of blockchain system that incorporates different CAs
- Design an experimental framework that will provide impartial and valid testing on CAGE and PoET CAs in a blockchain system.
- Analysis of results and providing summary of improvements and comparative performance of CAGE
- Provide recommendations and guidelines on how selecting a CA is important to the development of BCT

1.5 Research Method

The research method used is predominantly comparative analysis.

1.6 Resources

The following resources were used throughout the completion of this research:

- Computer
  - Processor: AMD Ryzen 7 3700X Eight Core CPU
  - Motherboard: ASUS PRIME B450-PLUS
  - Memory: 16GB Corsair VENGEANCE DDR4 3200MHz
  - Graphics Card: 8GB NVIDIA GEFORCE RTX 2060 SUPER
  - Storage: SSD 512GB PCIe M.2 SSD
  - OS: Windows 10 Home 64bit
- Programming Language: Python
  - **Modules**: time, subprocess, socket, json, threading, operator, math and hashlib
- Other Software
  - Microsoft Office: Word and Excel

1.7 Ethical Approval

BCT death, or the deletion of data on BCT is difficult. This is due to its decentralised nature. This has not been overlooked by the researchers. It is important to know that the data on the developed BCT is randomly selected and does not include personal data, which would be a mistake for any blockchain. Not only is the data random, but also the focus of this research is permissioned BCT, so the nodes can be distributed on the same machine, which is what was done during the design. This makes it easier to delete. Finally, all data is encrypted and hashed so this makes it computationally infeasible to decrypt.

The ethical approval was completed before the experiments were conducted and a copy is found in Appendix A.

The project was deemed minimum risk.

1.8 Planning

The project planning has been affected by the issues caused by COVID and subsequently taken longer than expected. The deliverables for the project are detailed below and relate to the GANTT chart used to track the progress of this project.

1.8.1 Deliverables

Deliverables are a result of actions that complete and attempt to satisfy objectives and can include:

Proposal: Complete proposal

Ethics: Complete Research Ethics approval, see Appendix A

Research: Complete research on Genetic Algorithms and Selection methods

Research: Complete research on Blockchain Technology

Software Development: Complete Software Development for CAGE

Experiments: Complete Experiments based on Experimental Framework

Results: Collate and gather information and data from Experiments

Improvements: Iterate through Software Development, Experiments and Results and make small improvements to the existing proposed algorithm. Stopping when these improvements start to outperform PoET results.

Analysis: Analyse Results and complete write-up of results

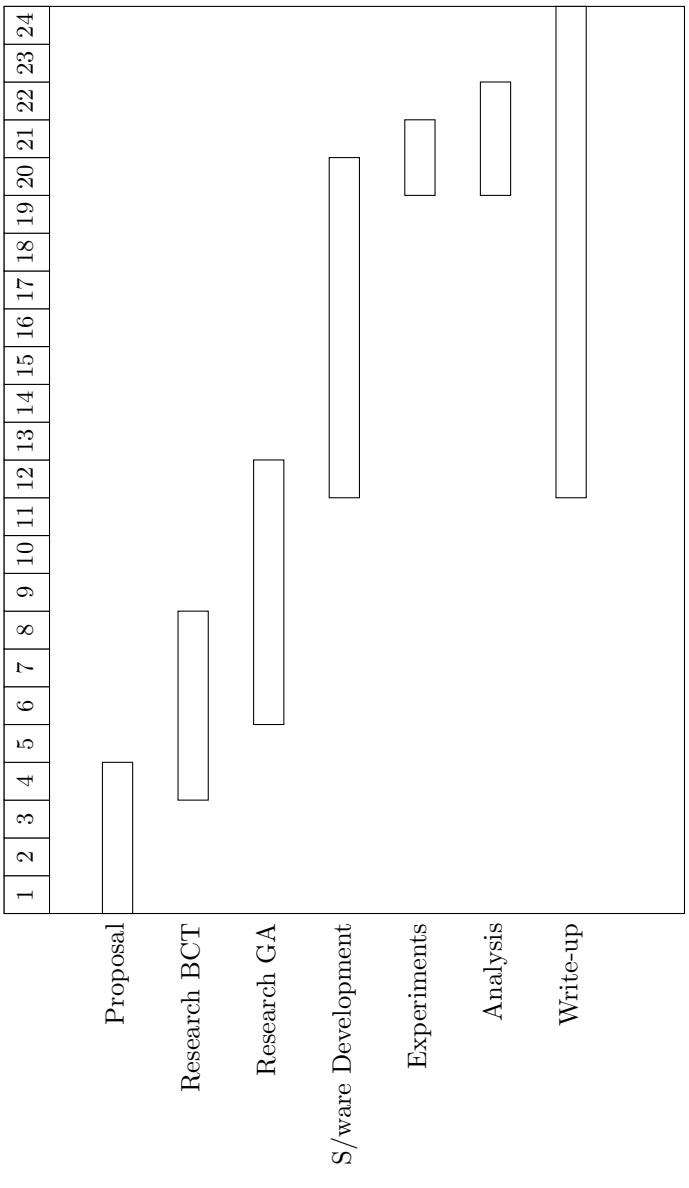

Figure 1.1: Planning for MRes

1.9 Summary

In summary, the research completed in this thesis is the combination of two areas of research: i) Evolutionary Algorithms (EA); and, ii) Blockchain Technology. Research in BCT identifies opportunities for improvements in CAs, whilst research in EAs investigates the best selection method that could be applied to develop a newly proposed CA. The merging of these subject domains results in a proposed Consensus Algorithm entitled, "Consensus Algorithm Genetically Encouraged", abbreviated to **CAGE**.

The software development was completed in Python and an experimental framework was developed to test both CAs in fairly. The results were evaluated and analysed in a comparative way to see how CAGE performed against PoET.

The structure of the rest of this thesis is as follows:

Chapter 2: covers the literature review and current research related to the problem;

Chapter 3: investigates the experiment and rationalises the method undertaken;

Chapter 4: evaluation and analysis the of the results. Improvements to the algorithm; and,

Chapter 5: includes the recommendations and conclusions.

Chapter 2

Literature Review

2.1 Introduction

This research focuses on using selection methods of evolutionary computing in blockchain's consensus algorithm to investigate if this branch of Artificial Intelligence (AI) can assist in blockchain's strive for safer, whether in permissioned or permissionless network, and more efficient consensus mechanisms, that are cost efficient both for the network provider and nodes to maintain continuous use.

2.2 Blockchain

The first work on a secured chain of blocks was done in 1991 by Stuart Haber and W. Scott Stornetta [24]. In 1992, with the addition of Bayer, they have added Merkle trees[31] to their design, which increased its efficiency. The first blockchain was later designed by Satoshi Nakamoto [37]. It had reduced the speed of adding blocks (the reduced speed along with high requirements placed on computing power was arranged to help prevent attacks against the network)

and was improved in the time-stamping area where blocks didn't need to be signed by a trusted party, thus removing the third party from peer-to-peer transactions. The following year design was implemented by Nakamoto to work with a cryptocurrency called bitcoin. [37] suggests blockchain as a secure solution for cash handling networks, that will also remove the necessity to use a trusted third party as a transaction handler, thus removing transaction costs. As a safeguard, Nakamoto proposes using proof-of-work (POW) as a consensus that guards network security and integrity, which along with other Bitcoin's safeguards are not entirely secure and possibly open to advanced adversary operations to exploit weaknesses for profit [12, 41]. POW is CPU heavy, as intended, therefore even though nodes working for network integrity are rewarded for it, they may leave when costs will get too high and not proportional to rewards. The first Bitcoin paper was expanded on in numerous papers, and since then there are a lot more than one blockchain [16], but a more recent paper published by NIST in 2018 [52] is a compendium of accepted knowledge about blockchain to date.

2.3 Blockchain Anatomy

Blockchain is a peer-to-peer network system represented by blocks linked as its immutable database[52]. It works by listening for new transactions from the network of users. Then those transactions are pooled into a block. The number of transaction per block varies and depends on the design of the blockchain system. The blocks are then presented to (miner) nodes that belong to the network. The blocks submitted are then to be approved, the way of approval is different and depends on the consensus algorithm. Once the block is approved, the transactions within are also approved, they then require verification.

The verification stage is where the decentralisation of the network is of importance. The consensus algorithm should be designed so that each independent

node in the network can verify the solution given quickly and efficiently. Once verified the block is then added to the blockchain. To increase the immutability property each block is linked to the previous block, this is completed by including the one-way cryptographic hash, e.g. SHA [1], of the previous block. There is one notable exception that is often referred to as the genesis block. This is the first block created and therefore cannot be linked to a previous block. It is important to distinguish between blocks, blockchains and transactions.

To re-engineer a blockchain becomes computationally infeasible as the number of blocks are added to the blockchain. If a rogue node was to alter one of the transactions, then the verification stage of the consensus algorithm would not be in unilateral agreement, or even a majority of agreement, and the block would not be published.

So, blockchain consists of a chain of blocks, a network of nodes and users, and a consensus algorithm. Firstly, the chain of blocks can be drawn parallels to the database of the blockchain system. Blocks themselves are data objects containing previous block header hash, time-stamp, and transactions in the minimum. Whatever is in blockchain is immutable, the chain is an append only data structure. This provides secure transaction history. Chain of blocks is generally decentralised and stored separately by every user in the network. In certain networks only nodes that have the most up to date chain are allowed to be miners, or verifiers.

Secondly, a network of users that could be further separated into standard nodes and miner nodes. Standard nodes are users who are interested in obtaining and selling the currency either for standard commercial or personal transactions, or for investing. Miner nodes are nodes which purpose is to keep the network running for a reward from network. In POW they are the ones solving difficult equations or in POS they are the ones bidding their stakes. It

is for a privilege to write their blocks that later get added to the chain.

Thirdly, a consensus algorithm which is a safeguard that dictates what are the rules upon which validation happens and who gets to add a new block, as well as, who gets rewarded.

Finally, these three stages are completed using asymmetric encryption resulting in a highly secure system.

These stages are summarised in the flowchart in Fig. 2.3 and each stage is explained as follows:

Pool TX : Pool the transactions, various strategies are used, but there is a link to performance. Grouping many transactions together is a fine balance and many strategies are used. In the canonical blockchain system the number of transactions per block is small and randomly generated.

Selection : The node selection is part of the consensus algorithm. In PoET each node in the network is allocated a random time and the node with the lowest time is selected.

Agreement : The solution from the selected node is verified by the network of nodes, this can be via a majority or unanimous. If consensus is not reached then another node is selected and the process repeated.

Link block : Various information in the block is stored and two further additions are always appended to the block. The first is a time-stamp and the second is a one-way cryptographic hash of the previous block (unless it is the genesis block).

Publish : The block is then published and appended to the blockchain on each of the nodes.

Repeat : The process is repeated whilst there are further transactions to process.

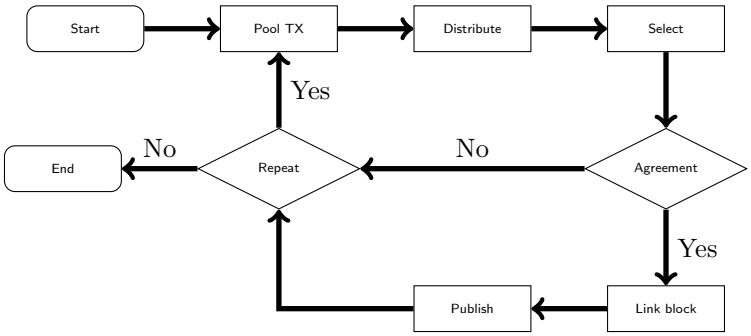

Figure 2.1: Canonical Blockchain Flowchart

There are many other attributes about blockchain that are outside the scope of this thesis, e.g., smart contracts [11], users, wallets and architecture.

The experiments deal with a canonical blockchain system as described above.

2.3.1 Consensus Algorithms

There are various types of consensus algorithms[6, 51]. Proof-of-Work (POW) being one of the first developed has high hardware requirements as it requires computers to solve a difficult task, which differs for every POW blockchain, for example a complicated mathematical puzzle [50]. Rewards, therefore, need to mirror the costs of running high-performance computers, with specialised GPUs and application-specific integrated circuits (ASIC) that can solve the task in a feasible time [47, 25]. Some networks using POW try to defend against ASIC use as to not decentralise the network using multi-hash POW network mechanisms [10]. If rewards were too small, the miners would leave the network and left it exposed to malicious attacks, which could cause monetary loses, private information leaks and finally shutting down the blockchain network. If rewards were too big, users might leave the network as it is them, who pay the reward charge based on the amount of money in their transaction. It is used in one of the biggest and most known blockchains currently running, Bitcoin [37],

and is generally used in permissionless networks.

Other widely known consensus algorithm is proof of stake (PoS). Ethereum, aside of Bitcoin it is also in the top of the most recognisable blockchains, is planning to upgrade to PoS from PoW. PoS works by having its nodes selected depending on stakes they are holding [38]. Nodes stake their tokens to take part in validation. Network selects next validator pseudo-randomly depending on stakes put by validators trying to get the role of the leader. Selected validator proposes block of transactions. Other validators verify and approve the transaction. The block gets added to the existing blockchain. The validator earns a transaction fee. With stake-based selection computational power is no longer an issue, therefore electricity cost is highly reduced to the point it is negligible. This method is used mostly in permissioned networks due to a need for a token, but it can be also used in permissionless networks as in [18].

Proof of Elapsed Time (POET) is another consensus algorithm. It was developed by Intel Corporation. It is used in Hyperledger Sawtooth [13]. It works by assigning to each participating node a wait-time. First node to wake up commits a new block and broadcasts it to the network. Nodes are not rewarded monetarily, but as in a permissioned network [9], all nodes have a benefit and value in making sure that the data shared is verified.

2.3.2 Further research on Blockchain Technology

Blockchains are the target of research for multiple papers which explain different algorithms used [44] and their comparative analysis [39].

In [39] researchers present a survey of CAs used in Blockchain along with their separation into two types, proof-based and voting-based, and advantages and disadvantages for both of those types. It goes over different consensus algorithms that have been already researched and proposed in Blockchain. In

a matter of comparison, the benchmarking framework provided by researchers working on BLOCKBENCH [16] provides tests on the performance of expected CA, which was one of the bases on blockchain performance benchmarking and optimization [46]. [16] is also a vast and comprehensive evaluation of virtual currency systems, which provides evidence of blockchain technological limitations. Some of the limitations lay around scalability or security and them being vulnerable to various types of attacks [16]. Consensus algorithms come to provide safety for the system, but it doesn't mean they are unbreakable. Research around Byzantine fault-tolerant algorithms (BFT)[8] shows weak points and how to counteract them. [7] explores safety measures for an asynchronous environment, like the Internet. The algorithm proposed is also equipped with a proactive recovery mechanism that recovers replicas periodically, even if there is no reason to suspect that they are indeed faulty. Unfortunately, the first work on BFT [7] was proven faulty with more recent research[28]. [28] states that the first BFT [7] contains an inbuilt fault, or error in judgement, with the system being able to detect lack of progress under one leader and choosing a new leader. It is proposed to use a leaderless system with the use of virtual leaders instead, meaning that servers instead of sending the message to the leader, send it to all other servers and then they decide which message should be proposed as the next leader's message. Servers then with help of a synchronous algorithm agree on the vector of proposed messages, a vector containing one proposal for each server. This can still be vulnerable to 51% attacks. 51% attack refers to an attack against blockchain network by a node (or group of nodes) controlling more 51% or more of processing power, hash rate or virtual currency available for stakes [23]. A solution to those attacks is proposed in [5] with an algorithm able to lower the chances of a successful attack in case of an attack where the attacker holds 51% or more of computing power combined in the blockchain.

Another solution, which was introduced in the same year 2015, is proposed in [4]. The paper states that different BFT state-machines were relying on a different set of conditions and during the development of service if the conditions differ even so slightly it may be that choosing one of the existing algorithms may not be the best fit. They propose reconciling existing protocols into one they called "ADAPT" that aims to use a larger set of conditions directed by Machine Learning.

2.3.3 Permissioned and permissionless BCT systems

Permissionless blockchains are blockchains where anyone can log in and become a member node and even a miner node. There is a safeguard system providing trust between transacting parties. This safeguard, consensus algorithm, is generally different to what is used in permissioned network due to a distrust inbuilt in a permissionless network. The most known example of permissionless blockchain is Bitcoin.

Permissioned blockchains usually involve a central overseeing authority or authorities [26]. Transactions are overseen but authorised nodes. Transaction blocks are proposed by authorised miners. This type of blockchain is usually used within a company constrains, to run an efficient supply chain or to securely share information in a network of authorities, like hospitals sharing health records, presented in [33].

2.3.4 Metrics

To compare objectively some metrics are required. In [46] the following metrics are used:

latency : the time difference between the submission of the transaction and the acceptance of the transaction.

throughput : the number of transactions per second, tps.

These are useful metrics that are used to analyse the performance of the consensus algorithm developed. However, these do not take into account two other important features that are also used to analyse the performance of the proposed CA, these are:

Blocks per Second, bps : The number of blocks per second submitted.

Distribution of Node Selection, DoNS : Looks at the distribution of node selected to commit the transaction.

These two other metrics will be incorporated in addition to throughput and latency. With a P2P system if the node selection is biased then there could be an issue with security, fault tolerance and performance. In essence, if the same node is selected than this would no longer be a decentralised system. Also a consensus algorithm that does not have a random distribution of node selection could benefit from performance since there would be less time spent on selecting a node. In fault tolerance, if a system becomes heavily reliant on nodes then it reduces its property to be fault tolerant.

In summary, the consensus algorithm will be inspired by Evolutionary Computation (see the next section) and then tested and compared to PoET. The analysis will look at the latency, throughput, blocks per second and distribution of Node Selection.

2.4 Evolutionary Computation

Evolutionary Computation is a topic that concerns the development of optimisation and search technique that are predominantly inspired by natural selection [27] or other biologically phenomenon, e.g., [17] Ant colony optimisation. Evolutionary Computation covers two main areas: i) Genetic Algorithms, GA; and,

ii) Genetic Programming, GP. Whilst there are fundamental differences between the two, it is selection methods that are of interest.

It is an efficient way of tackling complex problems inspired by Darwinian natural evolution [48]. EA being based around biological evolution uses the same mechanisms: reproduction, mutation, and selection. It works by randomly generating solution candidates (first generation). Then members of this generation are evaluated, and their fitness is measured on how well they perform in the task they were created to solve. Then comes the selection algorithm that using pseudo random method chooses the best candidates to be chosen to have the next generation of genes influenced by them by making crossovers of them. This method uses binary values to determine which values are taken from which "parent", it is more extensively explained in [27, 36]. Then another method, mutation, may be used on those "children" to apply random changes to them to have the population moving forward and possibly induce genes absent in previous generations, but beneficial to next ones. When this new generation is created it is being evaluated, its fitness is being measured and reproduction in being carried away. This cycle is being run usually until either there is no improvement for certain number of iterations, the certain number of generations has been reached, or the pre-defined evaluation or fitness value has been reached [43].

2.5 Selection Methods

Crucial part of EA or Genetic Algorithms (GA) are selection methods. There are various approaches to the subject [3] with proportional, linear, extinctive, preservative, elitist, pure, and more, sometimes different names are used. Those methods can often be used together, one of examples would be combining elitism with extinctive and linear method. This would make sure to keep the strongest

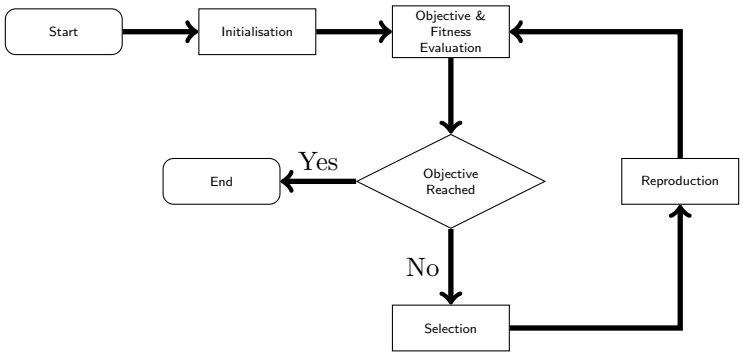

Figure 2.2: Flowchart illustrating the canonical genetic algorithm.

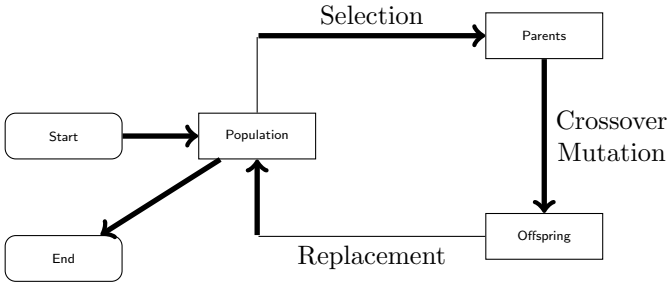

Figure 2.3: Flowchart illustrating the selection process inside the canonical genetic algorithm.

genes, even if they were already used to create offspring. Make sure that weakest genes don't get reproduction rights to the next generation. On the other hand, it would give higher chances to sub-optimal choices, comparing to proportional selection, if they are not at the bottom of fitness ranking.

Fitness-proportionate Selection with Roulette Wheel is one of the most common methods [36]. It uses "proportional" methodology with all individuals evaluated to have their fitness in numerical matter, then fitness of all individuals is summed up together, and then roulette wheel gets "spun". The random number R is generated between 0 and N, with N being the sum of the fitness of all individuals of the given generation. Then the algorithm goes from the beginning of the array of individuals and subtract their fitness from R. When R drops to 0

or below algorithm selects an individual whose fitness caused this. This method prioritises giving the highest chances to the most fit individual proportionally to how fit they are. This method is trained quickly but can often result in getting stuck in the local maximum. In [14] problems of getting stuck in local optima are well documented and methods of escape are proposed.

Rank selection is a selection method that falls under "linear" methodology. In rank selection it is required to evaluate fitness of all participating individuals and then sort them from the best to the worst. Then steps like roulette wheel can be taken to select, using number of individuals in a generation N and individual's location in generation array after sorting I, N-I (as simple example, can be more complicated [31]) parameter in roulette wheel instead of individuals' fitness. Method itself is a bit slower due to the need of sorting the array itself first, but also the training is slower than with fitness fitness-proportionate selection as it gives higher chances for suboptimal individuals to get chosen. On the other hand, this means that it has lower chances of getting stuck in local optima.

Tournament selection is a method combining fairness of rank selection with more computation friendly solution. It randomly selects two individuals from the pool. Then it generates a random number R between 0 and 1. It is then compared to static parameter P (for example 0.75). If R ¡ P, then the fitter individual is chosen. Parameter P in this case has 75% chance of picking the fitter individual and 25% chance to be left with less fit individual. Individuals then get returned to the selection pool and can be chosen again if randomisation falls on them [31]. This can be also altered by changing initial random selection to be rank based, which would reduce chances of unfit individuals chosen, as both would be initially selected pseudo randomly based on their rank.

Other methods worth mentioning mentioned in [36] are Sigma Scaling, Elitism, Boltzmann Selection, Steady-State Selection.

2.6 Summary

As stated in Ch. 1 this thesis is the merging of two technologies. The blockchain review focuses on consensus algorithms; and how they integrate with blockchain technologies. A canonical blockchain was developed that will form the basis of testing of the proposed consensus algorithm, CAGE, discussed in the next chapter.

The genetic algorithm review is less up-to-date since it is only trying to interpret the basic selection algorithms used and insert them into the canonical blockchain developed above. Of course further research into this topic may reveal better selection algorithms that can be adapted and form the basis of future work.

The chosen selection algorithm is based on the roulette wheel. Before looking at the reasons behind choosing roulette wheel, there are a few differences between the objectives of canonical GAs and BTC that influence this decision and are:

Latency : Nature has time on its side, in fact few people can visualise the millions of years evolution takes. Unfortunately, both GAs and BCTs don't have this luxury. To compete with other CAs it is known that the selection method chosen will have to work in milliseconds, see [46], rather than seconds. So, the selection method has to be fast.

Trust : Selecting weak individuals increases the diversity in evolutionary algorithms, sometimes at the expense of optimality. However, BTC there are only nodes that provide wrong and right answers and no in between. So, selecting nodes that are untrustworthy are then penalised heavily to prevent this reoccurring. In GAs there is a constant trade-off between exploration and exploitation, whereas in BCTs there is no exploitation,

only exploration.

Offspring : There is no reproduction, mutation and as a result no generation or offspring necessary in our CA for BCT. The re-evaluation of a new offspring is not necessary. Once the node has been selected, then its characteristics can be altered accordingly.

Decentralisation : Diversity is important to both BCTs and GAs but for different reasons. GAs require diversity to help avoid local optima, however, BCTs want diverse selection of nodes to maintain their decentralisation characteristics. Essentially, have 20 nodes to choose from and only selecting one would essentially make a decentralised system, centralised.

There are a few reasons this was chosen other selection methods, which are as follows.

Speed : Tournament selection relies on tournament size. To select tournament size and then select the node takes precious time. Normally, this method is selected because of its efficiency [42]. However, due to the modifications on proportionate selection there is less time spent on fitness scaling. Therefore this method was considered but not included in CAGE.

Fair : A fair system was needed that did not require the rigorous calculation of the objective and fitness of each node. What was required was a fairness in selection, that can be altered after selection. Unlike nature, the CA requires fairness to some extent to allow decentralisation.

Stamina : To prevent the same node from being selected a stamina parameter was added to each node. Each time the node gets selected this resets the stamina to its weakest and reduces the likelihood of the node being selected in the subsequent transactions. This concept is similar to coin age in Proof-of-Stake [32].

So, the reasons for selecting roulette wheel are a combination of the differences between BCTs and GAs and the above. In the next chapter the details of the modifications to the selection method are explained and the experimental framework is included.

Chapter 3

Method

3.1 Introduction

Testing will be done scientifically using comparative analysis between CAGE and POET. CAGE will be developed having genetic algorithm selection methods in mind. Algorithm will start by checking if there is any new node willing to enter the network. It will then proceed to add it with base of 10 fitness and 100 stamina. It will then refill stamina to all nodes with stamina below 100 according to pre-set rate of recovery. It will then sum up fitness of all nodes with stamina of 100. Following this algorithm will randomise an integer between 0 and the sum of fitness levels. After, it will go through the list of nodes, subtract the node's fitness from the sum, until it gets equal or lower than 0. When it does, the node which fitness caused that, will be selected to provide the block for the chain. This is roulette wheel with fitness selection. There was consideration to use rank selection to reduce monopoly of the best node, but rank selection offers lower performance due to constant need to sort miners by their fitness level, additionally stamina system will make sure that the node

with the highest fitness can't be chosen again until it's stamina regenerates back up to 100.

POET was chosen in the role of a competitor as an already established and recognised consensus algorithm with similar priorities; those are speed and low impact on hardware. Due to limited availability of already established systems that allow plugging of custom consensus algorithms for benchmarking purposes, both will be developed in python. Python was chosen due to large supply of available libraries to support development. Algorithms will be developed in simplified manner to ensure ability of the system to run tests as smoothly as possible. This means one central chain rather than a copy of chain for every node and communication done with central system rather than between all the nodes.

one central blockchain vs "n" amount of chains

"n" amount of communication channels vs "n*(n-1)" communication channels

Those simplifications should allow for the blockchain system to be developed and tested on available setup with various amounts of nodes to test system's scalability.

Developed algorithms will be tested for their "max transactions per second" (MTPS) index and "max blocks per second" by reducing "sleep time between transaction injection" (STBTI) variable to the point of breaking the program, which will happen when generated transaction list gets too long to be passed in a simple message between the server and the node. MTPS will be checked for different corresponding amounts of nodes available in the network that are able to add blocks to the blockchain. Data generated with those test runs will also be used to determine average delay between adding transaction to the list of eligible transactions and adding block containing that transaction to the chain,

as well as delay between blocks being added to the chain.

```
chain.append(EmptyStartBlock)
FitnessStaminArray = []
while (systemIsRunning):
    if( amountOfNodesInSystem > 0 AND amonutOfAvailableTransactions > 0):
        #add fitness and stamina to new nodes
        if( newNodeAvailable):
            FitenssStaminaArray.append(node: avalableNode, stamina: 100, fitness: 10))
        # refill stamina + add up fitness for roulette wheel
        max_fitness = 0
        for node in FitenssStaminArray:
            # refill stamina at a set rate up until 100
            node.stamina == min(
                    node.stamina + (1/math.sqrt(len(FitnessStaminaArray))),
                        100
                        )
            # roulette wheel, add fitness to a total sum
            if(node.stamina == 100):
                max_fitness += node.fitness
            # roulette wheel selection based on finess
            # add (or not) block from the chosen node
            # add/remove fitness to/from chosen node
            # empty stamina from chosen node

            # roulette wheel random integer use for slection
            rouletteRandom = random.randint(0, max_fitness)
            for node in FitnessStaminaArray:
                if (node.stamina == 100):
                    #remove from rouletteRandom until 0, to select node
                    rouletteRandom -= node.fitness
                    if (rouletteRandom <= 0):
                        if(previousBlockHash == node.previousHash):
                            chain.append(node.block)
                            removeFromAvailableTransactions(node.block.transaction)
                            node.fitness += (1 + node.block.transactionCount )
                            node.stamina = 0
                        else:
                            #if block has invalid transactions penalise node, allow min fitness point
                            node.fitness = max(1, node.fitness -1)
                            node.stamina = 0
                    else:
                        #if block doesn't include previous block's hash / has wrong one, penalise node
                        node.fitness = max(1, node.fitness-1)
                        node.stamina = 0
```

Figure 3.1: Coding for CAGE

3.2 CAGE

The algorithm is explained in the code displayed in Fig. 3.1. The selection method is based on fitness-proportionate selection with roulette wheel [45]. The fitness of the individual node depends on the calculations returned. If these calculations are invalid then the node's fitness is penalised and reduces the likelihood of being selected. If the calculations are valid, then the block is appended to the chain, the fitness increased and the stamina reduced. The stamina regulates the number of times a node can be selected. When a node is selected and returns a valid answer, the stamina is reduced to zero. The

stamina helps prevent too-strong selection methods allowing an individual node to dominate, which would lead to centralise-based system with a single node being selected more often. To monitor the node selection the results and analysis not only look at traditional throughput and latency but also node frequency and distribution.

Figure **??** is oversimplified and demonstrates node selection, node selection probability and stamina. This then has to be integrated with a blockchain system responsible for updating in a decentralised manner. The overall view of how CAGE is integrated within blockchain is illustrated in Fig. 3.2.

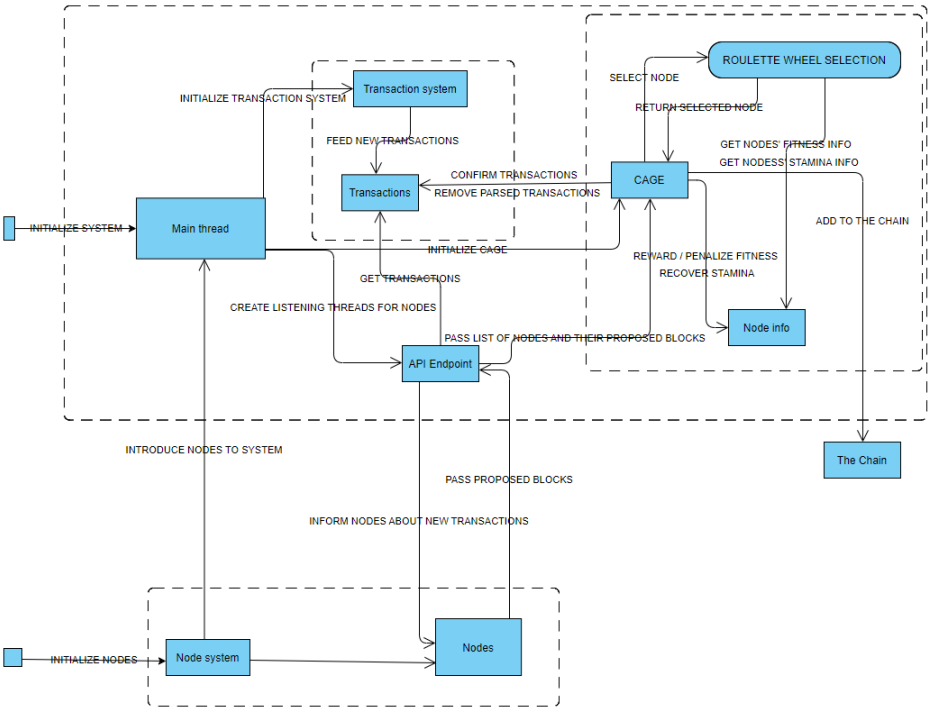

Figure 3.2: Schema diagram for software implementation of CAGE

3.3 Experimental Framework

Experiment will encompass finding MTPS and STBTI for POET and CAGE over a span of different count of working nodes. Higher MTPS means the algorithm is faster and more easily scalable to the industrial sizes. Proposed counts are 3, 10 and 20. Higher may be not possible on a PC used for experiments. Lowering sleep time between injections will increase number of transactions entering the system. Lower injection time will stretch resources available on tested system, which in turn will prove which algorithm can handle more transactions per second. Additionally, experiment will also explore measuring average delay between adding transaction to the list of eligible transactions and adding block containing given transaction to the chain, as well as average delay between inserting blocks to the chain. Those will be done by registering and comparing timestamps of blocks in chain as well as individual transaction timestamps. Tests will be repeated multiple times to ensure that data is as reliable as possible.

Description	Value
Transaction Delay	0, 3, 10, 20(ms)
Number of Nodes	3, 10, 20
Consensus Algorithm	CAGE, PoET

Table 3.1: Parameters for experiments

3.4 Summary

Firstly, experiment will be highly dependent on how fast CA can determine which out of nodes should provide the block this round. POET is expected to perform slightly worse due to inbuilt wait time, which is random but may still be slightly slower than lightweight CAGE algorithm. Adding more nodes, although beneficial for the network in the industrial setting, is not expected to

boost MTPS as this experiment is run in an environment within restrains of one machine. Adding additional nodes is not increasing processing power of the system.

Secondly, experiment is expected to show similar delay times in both algorithms. More nodes may slow down the machine causing the delay to increase. The delay in POET made by sleep timer may cause the average delay between transaction's generation and addition to chain to increase.

Finally, although both methods are highly dependent on limited processing power, the comparative analysis of the two algorithms should still provide valid feedback on speed and required processing power needed for each algorithm.

Chapter 4

Analysis & Results

4.1 Introduction

Setup for experiments:

- Processor (CPU): AMD Ryzen 7 3700X Eight Core CPU (3.6GHz-4.4GHz/36MB CACHE/AM4)
- Motherboard: ASUS PRIME B450-PLUS (DDR4, USB 3.1, 6Gb/s)
- Memory: 16GB Corsair VENGEANCE DDR4 3200MHz (2 x 8GB)
- Graphics Card: 8GB NVIDIA GEFORCE RTX 2060 SUPER
- SSD Drive: 512GB PCIe M.2 SSD (2000 MB/R, 1100 MB/W)
- Operating system: Windows 10 Home 64 Bit

Setup proved to be enough to write the code as well as debug it, but it became apparent that it is not sufficient to emulate the network with hundreds of participating nodes. Twenty nodes working simultaneously along with the main system itself proved to be a challenge and would often crash at the beginning

raw current time in miliseconds	reduced by start time		delay				
1.63525E+12	8348.280029		16				
1.63525E+12	8364.280029		16.01000977		avg delay		min delay
1.63525E+12	8380.290039		16		15.73901099		14.77001953
1.63525E+12	8396.290039		16.01000977				
1.63525E+12	8412.300049		15.14990234				
1.63525E+12	8427.449951		14.86010742				
1.63525E+12	8442.310059		15.95996094				
1.63525E+12	8458.27002		15.43994141				
1.63525E+12	8473.709961		15.57006836				

Parameter	Values (proposed)	values (actual)
TX Delay	0, 3, 10, 20 (ms)	15(ms)
# Nodes	3, 10, 20	3, 10, 20
Consensus Algorithm	CAGE, PoET	CAGE, PoET

Table 4.1: Results

or mid-test. After multitude of attempts to get 10-minute readings, it was decided that even runs that crash will get added to the data, as it was close to impossible to get the full 10-minute reading. Additionally, an issue came up, with using Windows as an OS, as sleep timers that proved to be necessary for creating intervals between adding new transactions to the system became locked at minimum of 15 milliseconds. It is specific to the setup. It was thoroughly tested for the best results and average came out to 15.74 ms of delay, with minimal delay being 14.77 ms. This unfortunately disallowed testing both systems by lowering the sleep timer on transaction injection, that would otherwise allow for additional venue of testing.

Moreover, running initial version of CAGE has shown the fault in the algorithm. Fault was introduced in the numerical interpretation of fitness level. In genetic algorithms, fitness is generated once and then given member of the populations is either selected or not for the next cycle. Next population has fitness calculated anew, which often is represented within specific range. In CAGE, the initial plan was to use fitness in roulette wheel selection to choose the right node to add to the blockchain and then reward selected node with bonus fitness if the node gave the block with transactions that were currently in the list of awaiting transactions. This proved to be perfectly fine with small setup of

three and ten nodes and stamina algorithm that required the node to wait for its stamina to recover. Stamina algorithm by itself proved to be insufficient to provide fair distribution of blocks between nodes. Potential fixes were to either change stamina algorithm to make nodes recover stamina slower, thus giving chance to more nodes before those with high fitness come back to rotation, or to change the selection method that is fairer when the difference between fitness levels gets too big, like rank selection, or finally to tweak roulette wheel selection to be less impacted by constant increases in fitness. At this point the decision fell on the last option, as the first option, changes to the stamina system would only help within test range, and not beyond those twenty potential tests. The second option of changing selection algorithm would lead to the loss of efficiency, as roulette wheel is an efficient algorithm with low complexity. The third option of tweaking the roulette wheel itself seem to be more efficient as it would not increase code's complexity and could have benefits that apply even within large sets of nodes. The decision was to lower the impact of increasing fitness, that is applying modifier of ceiling(sqrt(fitness)) to fitness. Ceiling is to ensure that the result is an integer as well as provide benefit for successful submission of the node a little earlier. Square root (sqrt) of the fitness is to lower the impact of constant increases of fitness. This provides bigger increments at lower ranges of fitness, requiring 7 points of fitness.

Example: Starting 10 points and 7 additional to reach 17, ceiling(sqrt(17)) would give 5, that is 25% more chance to be selected than a node that has not been selected yet with ceiling(sqrt(10))=4. In normal roulette wheel selection 7 points of fitness would increase the chance of being selected by 70%. This reduction in scaling of the impact that fitness has, while also retaining fitness, was enough to ensure fairer environment, while retaining the genetic algorithm principles of selection of the fittest and ensured no raise in algorithm efficiency.

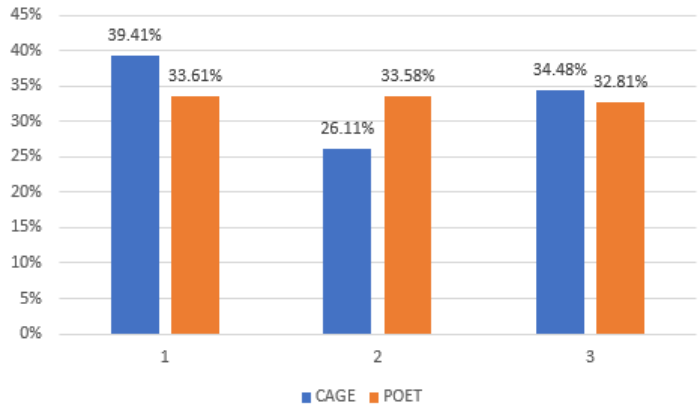

Figure 4.1: Selected Node distribution for 3 node blockchain system. PoET has a fair an equal distribution, CAGE less so. $x - axis$ is nodes, 1-3, and $y - axis$ is frequency in percentage.

Above figures present blocks distribution among nodes for each configuration. In 3-node system, POET gave almost equal results for each node with minimal advantage for the first node in the network as it allowed the first node to add block(s) without contestants. CAGE gave the same benefit of being the first node in the network to node number 1 which in turn allowed it to be selected more often later due to having higher fitness.

In 10-node configuration POET again provides almost equal distribution of blocks with a minimal edge to nodes that are introduced to the chain first. CAGE gives benefit of being first again to the node number 1, which allows it got gain fitness advantage over most of the nodes, while lucky few were also selected enough times to increase their fitness and be selected almost equally to the node number 1.

In 20-node configuration POET started to show differences between the distribution. This is still the same algorithm, but larger pool of nodes to select from, so random algorithm may provide results that are not equal or close to equal. The advantage of being the first node was greatly reduced, but first nodes

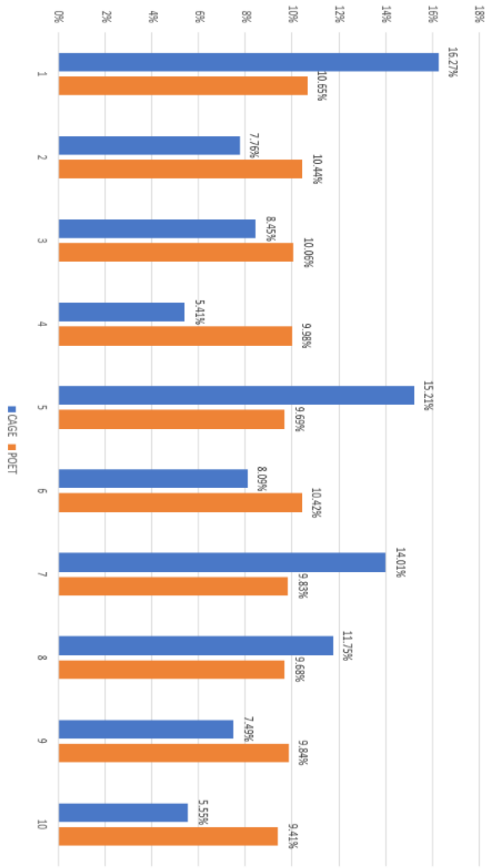

Figure 4.2: Selected Node distribution for 10 node blockchain system. PoET has a fair an equal distribution, CAGE less so. The difference this time is more noticeable with nodes 1, 5 & 7 being selected 44% of the time. $x-axis$ is nodes, 1-10, and $y-axis$ is frequency in percentage.

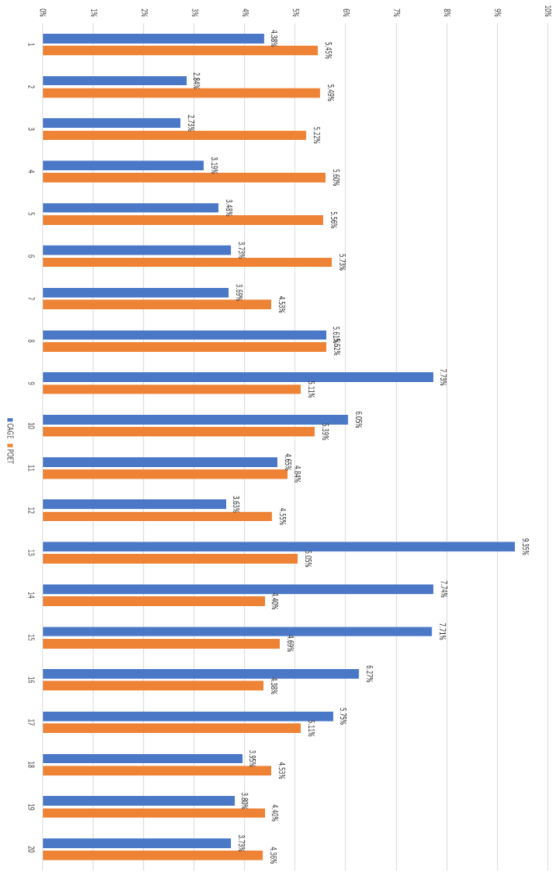

Figure 4.3: Selected Node distribution for 20 node blockchain system. PoET has a fair an equal distribution, CAGE less so. Again, the difference is more noticeable with nodes 9, 13, 14, & 15 having more chance of being selected. $x - axis$ is nodes, 1-20, and $y - axis$ is frequency in percentage.

nodes	3	10	20
	cage 3	cage 10	cage 20
total time	3027421	3012525	2958459
#blocks passed	83961	72571	54709
#blocks per second	27.73351	24.08976	18.4924
#transactions passed	123483	96293	76604
#transactions per second	40.78818	31.96422	25.89321
avg delay between blocks	35.8	41.8	58.8
avg delay transaction to chain	35.8	41.2	66.6
transactions per block	1.470719	1.32688	1.400208
	poet 3	poet 10	poet 20
total time	2605223	2080576	1657417
#blocks passed	43579	23985	9174
#blocks per second	16.72755	11.52806	5.535119
#transactions passed	134830	68260	33421
#transactions per second	51.75373	32.80822	20.16451
avg delay between blocks	60.6	86.2	183.8
avg delay transaction to chain	50.2	98.6	170.4
transactions per block	3.093921	2.845945	3.643013

Table 4.2: Results for experiments using CAGE and PoET consensus algorithms on Canonical Blockchain Systems.

still have a slight edge. CAGE for the first time has shown node number 1 to be outrun by other nodes. It got enough initial push to be able to get advantage over the next few nodes, but it was not enough to beat the 1:20 odds of being selected when more nodes got introduced into the network.

Running experiment five times for each configuration of three, ten and twenty nodes, and POET and CAGE as consensus algorithm came with following collective results:

Time was measured in milliseconds. 3,000,000 milliseconds were the target to get through five runs of ten minutes tests for each configuration. Unfortunately, consistent crashes due to the lack of system resources, made it nearly to impossible to collect the data needed for the 20-nodes configurations.

4.2 Experiment

The result of conducting the experiment shows that CAGE can pass through more blocks per second, than POET, regardless of the number of nodes participating in the blockchain, within the experimental system constraints. Number

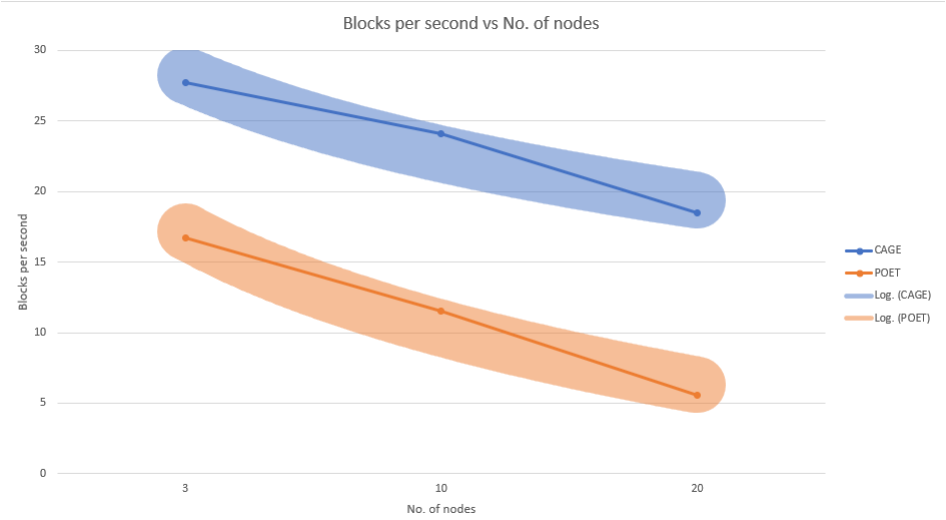

Figure 4.4: Blocks per second plotted against each experiment for 3, 10 & 20 nodes.

of blocks passed per second is vastly higher for CAGE.

Additionally, data gathered suggests that initially, even though POET has substantially lower number of blocks passed, those blocks are tightly packed with transactions and is able to even outperform CAGE. After more nodes is added and systems must pay processing power for the upkeep of those nodes CAGE comes on top with lower processing requirement per node, by just being responsible for adding or removing from fitness levels of selected nodes.

Maximal number of transactions per second assuming setup systems limitations is approx. 63 (1000 ms / 15.74 ms per transaction generation). This means that even 3 node systems lost approx. 35% efficiency for CAGE and approx. 18% for POET, which drops down to approx. 59% loss for CAGE and approx. 68% loss for POET in 20 node systems based on the number of nodes pushed into the system, algorithm efficiency and system limitations.

The experiment shows clearly that delay between blocks is not noticeably in-

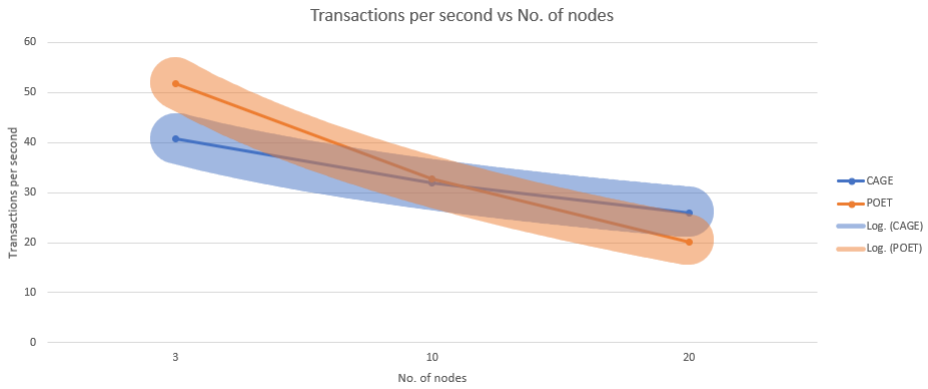

Figure 4.5: Transactions per second plotted against each experiment for 3, 10, & 20 nodes.

creasing in CAGE when adding more nodes, while adding more nodes to POET increases the interval between blocks being added to the chain exponentially. This was the speed limiting factor as adding more transactions per block was a limiter for the setup system that the framework was tested on.

Final figure provides information of average time needed for a transaction to get into the chain since a time of its creation. This is highly dependent on the delay between blocks as higher rate of blocks being included makes sure that as soon as transaction is proposed into the next available block it gets accepted to the chain. The longer the delay between blocks, the longer additional time of delay for the transaction to be accepted into the chain. This suggests high scalability in production environment with thousands of nodes participating.

4.3 Summary

In summary, experiment shows that to get the best results, the testing system should be more powerful and run over a net of connected devices, rather than on one device, either through secure internet framework, or a set of devices con-

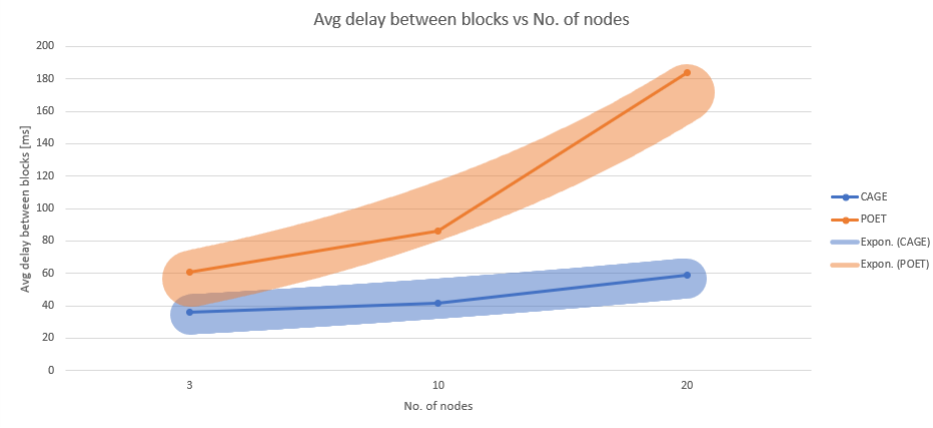

Figure 4.6: Mean delay between blocks (ms) plotted against each experiment for 3, 10, & 20 nodes.

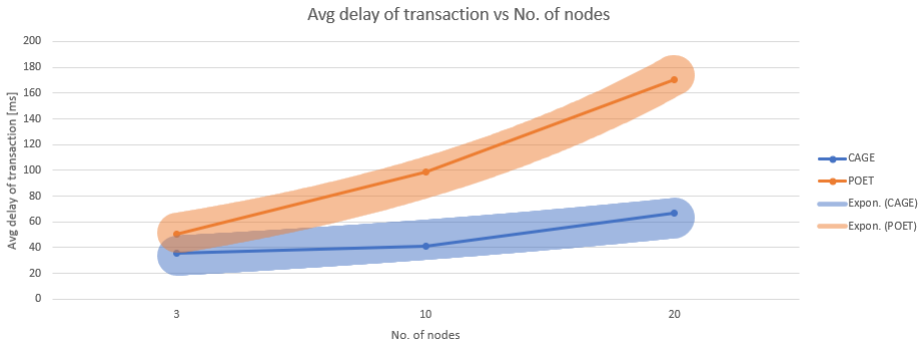

Figure 4.7: Mean delay between blocks (ms) plotted against each experiment for 3, 10, & 20 nodes.

nected through LAN. Which is impossible to do with the available setup for the experiment. This experiment can provide initial information, but unfortunately cannot prove scalability and cannot show what would be the limit in real production environment. The available system can provide comparative analysis on small set of data which can lead to informed judgement about scalability.

Comparatively, CAGE can pass more blocks per second, which consequentially leads to system's ability to sustain more transactions per second. This was proven with the experiment, but results for transactions per second were close for both consensus algorithms. Additionally, the experiment proved that CAGE provides lower delay between introduction of the transaction to the system and addition of said transaction to the chain, which for an end user may mean the delay between attempting to pay and the payment completion. In conclusion, experiment proved CAGE to be a competitive consensus algorithm, worthy of consideration when developing an environment relying on blockchain.

Chapter 5

Conclusions

5.1 Introduction

Following the research, development, and experiments, all the aims and objectives have been reached. Blockchain technologies and selection algorithms in GAs have been researched in chapter 2. The merge of two established technologies has been successful in chapter 3 when presenting the idea of implementation for CAGE. A new consensus algorithm, along with the testing environment, has been developed, with its idea being presented in chapter 3 and the progress of development shown in chapter 4. CAGE was developed using the selection algorithm, the Roulette Wheel, an algorithm in mind, which is generally seen as a selection method in Genetic Algorithms development. It had to be adjusted to work with continuously increasing fitness instead of how it usually is, fitness that is aligned to one population and is evaluated again when the next generation of the population comes. Cage was tested comparatively against PoET and details are shown in chapter 4. CAGE proved to be a successfully developed algorithm able to compete efficiency-wise against PoET, which is known as a lightweight

and fast consensus algorithm that is used commercially in Hyperledger Sawtooth. CAGE proved to be better and has shown to be more promising when it comes to the scalability of the network. Recommendations on selecting a CA depending on the blockchain developed are presented in chapters 2, 4 and 5.

5.2 Recommendations

Development of this project did not only lead to the development of a new consensus algorithm, but also to a realization that when developing a new consensus algorithm or applying custom modifications to an already existing one it is important to not only look at the efficiency and security but also on fairness when choosing a node to provide a block for the chain. It is easy to overlook factors, that in the end could potentially have disastrous repercussions even with impeccable security. When always selecting the same perfect and mostly used node consistently, it stops being a blockchain and starts to be just a ledger for that one node. During the development of this project, it was discovered that the algorithm in a bigger network was prioritising mainly a few nodes that got ahead and the effect of them staying ahead in fitness followed a pattern of a snowball effect. Therefore, after everything was developed and already in a testing phase, changes to the selection algorithm had to be made. Otherwise, the system would be too easy to monopolise by a small subset of nodes, which is not desired even in a permissioned network, where all nodes are trusted by default.

5.3 Future Work

Although the developed algorithm is successful and efficient for the purpose of being used in high traffic permissioned networks it does not have the security

required for permissionless networks. Such lightweight algorithm then would be put into direct competition against POW and other consensus algorithms that might not be as fast as CAGE but are resistant to attacks of malicious nodes. If CAGE was to be used in a permissionless network or in a network that could be susceptible to external infiltrations, the algorithm that governs the penalization of nodes providing incorrect blocks should be polished and balanced properly to discourage and fight off potential threats. Additionally, another layer of protection may be introduced by moving away from the test environment that is developed on a centralised system with its design following permissioned blockchain networks that have a centralised "system entity", rather than on a peer-to-peer network of decentralised nodes, as this could introduce verification of blocks done by other nodes. Therefore, modification of CAGE to be a consensus algorithm able to perform safely in a permissionless network is considered a future work to extend on the ongoing project.

Moreover, it would be highly beneficial to test CAGE in a network of nodes operating on separate machines, which would be a closer representation of how this algorithm would be used in a production environment. The testing done in this project was sufficient, as it was done a comparatively by placing both CAGE and PoET in the same enclosed system to see how they performed in the same situation, but it would be worth seeing how CAGE would perform in a larger and decentralised system developed over a large network of nodes imitating the networks that are used commercially.

Furthermore, as CAGE was compared against PoET, it would be constructive to benchmark it against other algorithms that are used in permissioned networks. Even though PoET is an efficient CA that is used in commercial blockchain, as it is completely unique in how it works, it provides low security with a selection algorithm providing completely random chances of being se-

lected, it is efficient but not secure and completely unsuitable for permissionless networks, which of course it was not built for. Comparing CAGE against one of the voting algorithms would be valuable, as the underlying algorithm of providing the highest chances of being selected to the node that is the fittest for it, is resembling the premise of voting algorithms that vote for the fittest node to provide a block for the chain. Benchmarking CAGE against one of those algorithms would potentially test those not only for performance, but also for security when exposed to attack, or the fairness of giving a chance for all nodes to participate in building the chain and not only the selected subset of nodes in the system.

5.4 Summary

To conclude the research on CAGE, the newly developed consensus algorithm proved to be a success, it works, it uses one of the most common selection algorithms used in Genetic Algorithms, the Roulette Wheel Selection, and the logic of the algorithm follows the principles of genetic evolution by selection of the fittest and raising the chances of the fittest nodes to be the ones being chosen more often. All the aims and objectives have been met. The development and testing environment proved to be sufficient to engage in a comparative analysis, it has provided minor setbacks and several crashes of the system, but the data received was obtained without compromising on lowering the sample size below the original intentions. Comparatively, the new CA has performed better than PoET is a developed test environment and based on the trends, from the data obtained through testing, it has promising scalability for large high traffic permissioned networks. In the end, research was finished successfully even through several setbacks and the developed algorithm managed to be a success.

Bibliography

[1] Federal Information Processing Standards (FIPS) 180-1. Secure hash standard. 1996.

[2] Andreas M. Antonopoulos and Gavin Wood. *Mastering Ethereum*. O'Reilly, 1st edition, 2018.

[3] Thomas Bäck and Frank Hoffmeister. Extended selection mechanisms in genetic algorithms. 1991.

[4] Jean-Paul Bahsoun, Rachid Guerraoui, and Ali Shoker. Making bft protocols really adaptive. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pages 904–913. IEEE, 2015.

[5] Martijn Bastiaan. Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin. 2015. [Accessed: Jan 2022].

[6] Binance.com. What is a blockchain consensus algorithm. `https://academy.binance.com/en/articles/what-is-a-blockchain-consensus-algorithm?ref=AZTKZ9XS`, 2018. [Accessed Sep. 2021].

[7] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.

[8] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[9] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282–297. Springer, 2017.

[10] Hyungmin Cho. Asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols. *IEEE Access*, 6:66210–66222, 2018.

[11] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[12] Mauro Conti, E Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, 2018.

[13] Amie Corso. *Performance analysis of proof-of-elapsed-time (poet) consensus in the sawtooth blockchain framework*. PhD thesis, University of Oregon, 2019.

[14] Duc-Cuong Dang, Tobias Friedrich, Timo Kötzing, Martin S Krejca, Per Kristian Lehre, Pietro S Oliveto, Dirk Sudholt, and Andrew M Sutton. Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation*, 22(3):484–497, 2017.

[15] Debojyoti Das and Anupam Dutta. Bitcoin's energy consumption: Is it the achilles heel to miner's revenue? *Economics Letters*, 186:108530, 2020.

[16] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains.

In *Proceedings of the 2017 ACM international conference on management of data*, pages 1085–1100, 2017.

[17] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[18] Lei Fan, Jonathan Katz, Phuc Thai, and Hong-Sheng Zhou. A permissionless proof-of-stake blockchain with best-possible unpredictability. *Cryptology ePrint Archive, https://eprint.iacr.org/2021/660.pdf*, 2021. [accessed Jan 2022].

[19] Evgeniia Filippova. Empirical evidence and economic implications of blockchain as a general purpose technology. In *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, pages 1–8. IEEE, 2019.

[20] Evgeniia Filippova, Arno Scharl, and Pavel Filippov. Blockchain: an empirical investigation of its scope for improvement. In *International Conference on Blockchain*, pages 1–17. Springer, 2019.

[21] The Linux Foundation. Hyperledger architecture, volume 1. hyperlink, 2017. [Accessed: Jan 2021].

[22] The Linux Foundation. Hyperledger architecture, volume 2. hyperlink, 2018. [Accessed: Jan 2021].

[23] Jake Frankenfield. 51% attack. hyperlink, 2021. [Accessed Sep 2021].

[24] Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

[25] Yoshinori Hashimoto and Shunya Noda. Pricing of mining asic and its implication to the high volatility of cryptocurrency prices. *Available at SSRN 3368286*, 2019.

[26] Christine V Helliar, Louise Crawford, Laura Rocca, Claudio Teodori, and Monica Veneziani. Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, 54:102136, 2020.

[27] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[28] Leslie Lamport. Brief announcement: Leaderless byzantine paxos. In *International Symposium on Distributed Computing*, pages 141–142. Springer, 2011.

[29] Zhuotao Liu, Yangxi Xiang, Jian Shi, Peng Gao, Haoyu Wang, Xusheng Xiao, Bihan Wen, Qi Li, and Yih-Chun Hu. Make web3. 0 connected. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[30] Rosa Lombardi, Charl de Villiers, Nicola Moscariello, and Michele Pizzo. The disruption of blockchain in auditing–a systematic literature review and an agenda for future research. *Accounting, Auditing & Accountability Journal*, 2021.

[31] Ralph C Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

[32] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2567–2572. IEEE, 2017.

[33] I. Mitchell and S. Hara. BMAR - blockchain for medication administration records. *Blockchain and Clinical Trial - Securing Patient Data*, pages 231–248, 2019.

[34] I. Mitchell, M. Sheriff, and S. Hara. DappER: Decentralised Application for Exam Reviews. *Global Security, Safety and Sustainability The Security Challenges of the Connected World*, 2019.

[35] Ian Mitchell, Sukhvinder Hara, Hamid Jahankhani, and David Neilson. Blockchain of custody, boc. In *CYBER SECURITY PRACTITIONER'S GUIDE*, pages 365–397. World Scientific, 2020.

[36] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[37] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. hyperlink, 2008. [Accessed: Jan 2022].

[38] Cong T Nguyen, Dinh Thai Hoang, Diep N Nguyen, Dusit Niyato, Huynh Tuong Nguyen, and Eryk Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. *IEEE Access*, 7:85727–85745, 2019.

[39] Giang-Truong Nguyen and Kyungbaek Kim. A survey about consensus algorithms used in blockchain. *Journal of Information processing systems*, 14(1), 2018.

[40] Karl J O'Dwyer and David Malone. Bitcoin mining and its energy footprint. *Irish Signals & Systems Conf. (ISSC)*, 2014.

[41] Jin Ho Park and Jong Hyuk Park. Blockchain security in cloud computing: Use cases, challenges, and solutions. *Symmetry*, 9(8):164, 2017.

[42] Noraini Mohd Razali, John Geraghty, et al. Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world*

congress on engineering, volume 2, pages 1–6. International Association of Engineers Hong Kong, China, 2011.

[43] Martín Safe, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. On stopping criteria for genetic algorithms. In *Brazilian Symposium on Artificial Intelligence*, pages 405–413. Springer, 2004.

[44] Lakshmi Siva Sankar, M Sindhu, and M Sethumadhavan. Survey of consensus protocols on blockchain applications. In *2017 4th international conference on advanced computing and communication systems (ICACCS)*, pages 1–5. IEEE, 2017.

[45] Anupriya Shukla, Hari Mohan Pandey, and Deepti Mehrotra. Comparative review of selection techniques in genetic algorithm. In *2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*, pages 515–519. IEEE, 2015.

[46] Parth Thakkar, Senthil Nathan, and Balaji Vishwanathan. Performance benchmarking and optimizing hyperledger fabric blockchain platform. *arXiv preprint arXiv:1805.11390*, 2018.

[47] Philip Treleaven, Richard Gendal Brown, and Danny Yang. Blockchain technology in finance. *Computer*, 50(9):14–17, 2017.

[48] Pradnya A Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE, 2016.

[49] Harald Vranken. Sustainability of bitcoin and blockchains. *Current opinion in environmental sustainability*, 28:1–9, 2017.

[50] Brent Waters, Ari Juels, J Alex Halderman, and Edward W Felten. New client puzzle outsourcing techniques for dos resistance. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 246–256, 2004.

[51] Zane Witherspoon. A hitchhiker's guide to consensus algorithms. `https://hackernoon.com/a-hitchhikers-guide-to-consensus-algorithms-d81aae3eb0e3`, 2017. [Accessed Sep. 2020].

[52] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. Technical report, National Institute of Standards and Technology, 2018.

Appendices

Appendix A

Ethical Approval

The project was identified as low risk and did not require any further ethical approval.

Research Ethics Screening Form for Students

Middlesex University is concerned with protecting the rights, health, safety, dignity, and privacy of its research participants. It is also concerned with protecting the health, safety, rights, and academic freedom of its students and with safeguarding its own reputation for conducting high quality, ethical research.

This Research Ethics Screening Form will enable students to self-assess and determine whether the research requires ethical review and approval via the Middlesex Online Research Ethics (MORE) form before commencing the study. Supervisors must approve this form after consultation with students.

Student Name:	Kamil Maka	Email:
Research project title:	CAGE: Consensus Algorithms Genetically Encouraged	
Programme of study/module:	MRes	
Supervisor Name:	Ian Mitchell	Email:

Please answer whether your research/study involves any of the following given below:		
1. [H]ANIMALS or animal parts.	☐ Yes	☒ No
2. [M]CELL LINES (established and commercially available cells - biological research).	☐ Yes	☒ No
3. [H]CELL CULTURE (Primary: from animal/human cells- biological research).	☐ Yes	☒ No
4. [H]CLINICAL Audits or Assessments (e.g. in medical settings).	☐ Yes	☒ No
5. [X]CONFLICT of INTEREST or lack of IMPARTIALITY. If unsure see "Code of Practice for Research" (Sec 3.5) at: https://unihub.mdx.ac.uk/study/spotlights/types/research-at-middlesex/research-ethics	☐ Yes	☒ No
6. [X]DATA to be used that is not freely available (e.g. secondary data needing permission for access or use).	☐ Yes	☒ No
7. [X]DAMAGE (e.g., to precious artefacts or to the environment) or present a significant risk to society).	☐ Yes	☒ No
8. [X]EXTERNAL ORGANISATION – research carried out within an external organisation or your reseach is commissioned by a government (or government body).	☐ Yes	☒ No
9. [M]FIELDWORK (e.g biological research, ethnography studies).	☐ Yes	☒ No
10. [H]GENETICALLTY MODIFIED ORGANISMS (GMOs) (biological research).	☐ Yes	☒ No
11. [H]GENE THERAPY including DNA sequenced data (biological research).	☐ Yes	☒ No
12. [M]HUMAN PARTICIPANTS – ANONYMOUS Questionnaires (participants not identified or identifiable).	☐ Yes	☒ No
13. [X]HUMAN PARTICIPANTS – IDENTIFIABLE (participants are identified or can be identified): survey questionnaire/ INTERVIEWS / focus groups / experiments / observation studies.	☐ Yes	☒ No
14. [H]HUMAN TISSUE (e.g., human relevant material, e.g., blood, saliva, urine, breast milk, faecal material).	☐ Yes	☒ No

15. [H]ILLEGAL/HARMFUL activities research (e.g., development of technology intended to be used in an illegal/harmful context or to breach security systems, searching the internet for information on highly sensitive topics such as child and extreme pornography, terrorism, use of the DARK WEB, research harmful to national security).	☐Yes	☒No
16. [X]PERMISSION is required to access premises or research participants.	☐Yes	☒No
17. [X]PERSONAL DATA PROCESSING (Any activity with data that can directly or indirectly identify a living person). For example data gathered from interviews, databases, digital devices such as mobile phones, social media or internet platforms or apps with or without individuals'/owners' knowledge or consent, and/or could lead to individuals/owners being IDENTIFIED or SPECIAL CATEGORY DATA (GDPR[1]) or CRIMINAL OFFENCE DATA. [1]Special category data (GDPR- Art.9): "personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation".	☐Yes	☒No
18. [X]PUBLIC WORKS DOCTORATES: Evidence of permission is required for use of works/artifacts (that are protected by Intellectual Property (IP) Rights, e.g. copyright, design right) in a doctoral critical commentary when the IP in the work/artifactis jointly prepared/produced or is owned by another body	☐Yes	☒No
19. [H]RISK OF PHYSICAL OR PSYCHOLOGICAL HARM (e.g., TRAVEL to dangerous places in your own country or in a foreign country (see https://www.gov.uk/foreign-travel-advice), research with NGOs/humanitarian groups in conflict/dangerous zones, development of technology/agent/chemical that may be harmful to others, any other foreseeable dangerous risks).	☐Yes	☒No
20. [X]SECURITY CLEARANCE – required for research.	☐Yes	☒No
21. [X]SENSITIVE TOPICS (e.g., anything deeply personal and distressing, taboo, intrusive, stigmatising, sexual in nature, potentially dangerous, etc).	☐Yes	☒No

M – Minimal Risk; X – More than Minimal Risk. H – High Risk

If you have answered 'Yes' to ANY of the items in the table, your application REQUIRES ethical review and approval using the MOREform **BEFORE commencing your research**. Please apply for ethical approval using the MOREform (https://moreform.mdx.ac.uk/).

If you have answered 'No' to ALL of the items in the table, your application is Low Risk and you may NOT require ethical review and approval using the MOREform before commencing your research. Your research supervisor will confirm this below.

Student Signature:…Kamil Maka Date:20/11/2020

To be completed by the supervisor:

Based on the details provided in the self-assesment form, I confirm that:	Insert Y or N
The study is Low Risk and *does not require* ethical review & approval using the MOREform	Yes
The study *requires* ethical review and approval using the MOREform.	No

Superivsor Signature:…Ian Mitchell.. Date: 20/11/2020