*Department of Computer Science and Biomedical Informatics*
*University of Thessaly, Lamia, Greece, 17-07-2017*

# Implementing A Church–Turing–Deutsch Principle Machine on a Blockchain.

**Konstantinos Sgantzos.**
Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia 35100, Greece
Correspondence to: sgacos@gmail.com

***Abstract* — Genetic Algorithms are the elementary particles of a brand-new world of computing. In recent years, technology has evolved exponentially in terms of Hardware, but not in terms of Software. Genetic Algorithms (GAs) are already filling this gap in fields like Big Data mining, Protein Folding predictions, Finance, etc. In this paper we present the possibility of using an "Unbounded Single Taped Turing" medium like a Blockchain to store a Genetic Algorithm that will be able to provide Turing Complete results on any mathematically given problem.**

**Keywords:** Genetic Algorithm, Cellular Automaton, Church-Turing thesis, Universal Computing Device, Blockchain, Bitcoin, Consensus, Decider, Two Stack Push Down Automaton, Ackerman Function Simulation, External Oracles.

## I. Introduction

A Genetic Algorithm is broadly considered as a "black box" of software; it literally emulates the way Natural Evolution has used billions of years to evolve. Starting from a Prokaryote, given the time and the proper evolutionary conditions, Natural Evolution managed to produce Eukaryotes through DNA replication, Cell Division, Random Mutations and trillion replications and recombinations [1], [2]. The goal of such an initiative would be no other than reach what we now describe as "Artificial Intelligence".

There is a distinct difference between Genetic Algorithms and Classical Algorithms in two key points. Classical Algorithms generate only one instance that has a specific goal of solving a problem by approaching the optimal solution, while using deterministic computation. On the other hand, Genetic Algorithms create a population of instances with each iteration. It is the swarm intelligence of those instances that approach the solution of the problem on the best possible way. Genetic Algorithms are not using deterministic computation, but a computation based on Random Number Generators [3]. One of the most common implementation of Genetic Algorithms is the Cellular Automaton.

The concept of a Cellular Automaton (plural: Cellular Automata) was first introduced by John von Neumann in the Hixon Symposium in 1951 [4]. It was described as a discrete model that consists of a simple two-state, one dimensional grid of cells that can be either on or off. Later, in the 1970s a two-state, two-dimensional cellular automaton named "Game of Life" by John Conway, became widely known [5] but it wasn't until the 1980s with the work of Stephen Wolfram when a systematic study of two-state, one-dimension of Cellular Automata was done [6], presenting the implementation of a Cellular Automaton based on specific set of rules. Wolfram named those "Elementary Cellular Automata" and his research assistant Mathew Cook showed that one of these rules is Turing-Complete. Their work has been published in 2002 in the bestselling book "A New Kind of Science" [7].

In computability theory [8], a Cellular Automaton can be Turing Complete, if it can be used to simulate any single-taped Turing Machine. The term was named after the Computer Scientist and Mathematician Alan Turing. A typical example of such an implementation is the Lambda Calculus which was introduced in 1930 by Alonzo Church [9].

As an extension to the above notion, if such an Automaton is formed, then a swarm of Cellular Automata of similar origin could possibly form what is described as a Church-Turing thesis [10]. Furthermore, above a certain point of computational evolution, they could form what is described by the Church–Turing–Deutsch principle. The principle states that a **Universal Computing Device** can simulate every physical process [11], [12].

In this paper we present that it is theoretically possible for a Turing-Complete algorithm, like a Cellular Automaton based on rule 110, to be implemented on an Unbounded Single Taped Turing Medium such as a Blockchain.

## II. Context

**Blockchain,** was first introduced in Bitcoin [13], as a universal, fully shared Ledger that would be globally visible to all parties when a transaction was recorded on it without any presence of a trusted central authority. In each transaction, the previous owner signs - using the secret signing key corresponding to his public key a hash of the transaction in which he received the bitcoins (in practice, a SHA-256 hash) and the public key of the next owner [14].

The concept of PoW (Proof of Work) introduced a reward mechanism to the solvers of a random SHA256 puzzle. A hash puzzle is a set of mathematical problems which is solved by creating a hash that conforms to a specific requirement. In theory, those puzzles can be mathematically represented in the form of a random generated matrix where the difficulty plays a significant part in its creation. In Bitcoin it is a difficulty requirement to have the hash be lower than a specific threshold. It must be noted here that as with many cryptographic puzzles such as Elliptic Curve Cryptography [15], calculating the SHA256 can be rendered as a matrix calculation or a simple linear algebra problem. A classical paradigm is the Hill Cypher [16]. The miners can be seen as solving a Linear system of the form AX <= B where A, X and B are N x N dense matrices. The Eigenvalue of the matrix could verify the solution of the puzzle. This representation fits the use of FPGA and ASIC computing. Then, if the answer is found, the fastest participant is rewarded. The current reward stands at 12.5 Bitcoins at the time of writing this paper [17].

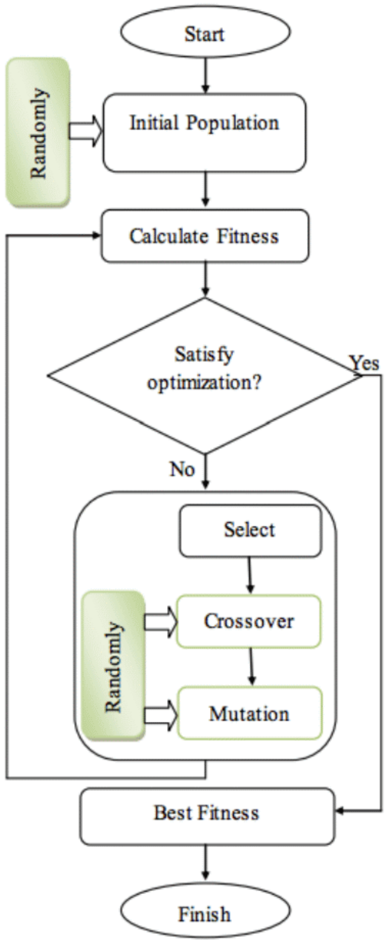
A certain state called "**Consensus**" is reached by following the chain with the most proof of work (ie: most difficulty behind it) which is what everyone trusts and agree to be the most valid one. This way, every participant agrees over a certain block of transactions, so that there are no conflicts of any given transaction on each block or a previous one [18]. The above notion can be described as "The two Generals" problem and it is considered fundamental in Computer Science.

Abstractly speaking, we could perceive a Blockchain as a representation of a recording cylinder, similar to Edison Standard Phonograph (Fig. 1) where the transactions are being recorded in blocks, forming a spiral of ellipses, from left to right, whereas the recording is permanent and the cylinder virtually unbounded. This concept describes perfectly the theoretical representation of an Unbounded Single Taped Turing Medium.

**Fig. 1 : Edison Standard Phonograph**

In such a medium, it is technically possible to store the evolutionary swarm of a Genetic Algorithm, that is mathematically proven to already be Turing-Complete. Such implementations are documented in the instances of Elementary Cellular Automaton based on Rule 110 [19], or Rule 30 [6].

**A Basic Genetic Algorithm**, can be represented in the following graph (Fig. 2)



Basic Genetic Algorithm flowchart

**Fig. 2 : Basic Genetic Algorithm Flowchart**

The initial population and the iteration process include two Randomizations that cannot exceed the range from 0.02% to 2% of the total mutation of the Genetic Algorithm. The randomizations are provided by a computational Random Number Generator in both cases. It is clearly observable that the key point to the fitness procedure relies on the Random element. In the recent years, several scientists tried to implement alterations to the evolutionary process that could derive better results [20].

**A Chaotic Genetic Algorithm.** Based on an idea of I.G. Tsoulos [21], Reza Ebrahimzadeh and Mahdi Jampour [22] introduced a Chaotic random number mutational variable where the classical Genetic Algorithm used computational Randomization and they observed significant time optimizations of the fitting process.

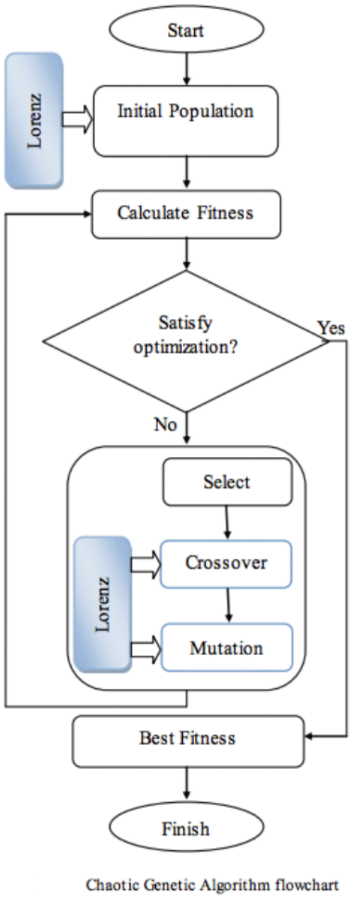


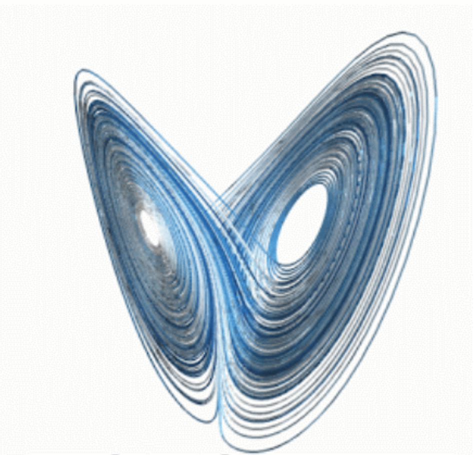
**Fig. 3: Chaotic Genetic Algorithm Flowchart**

**Deterministic Chaos systems vs Natural Random systems.** The definition of a deterministic chaos (or simply chaos) system was given by E. Lorenz [23]:

*Chaos: When the present determines the future, but the approximate present does not approximately determine the future.*

Chaotic systems were mathematically described through a truncated Navier-Stokes partial derivation system, that qualified the representation of chaotic phenomena such as the topology of the water molecules within a storm, or the movement of a double compound pendulum, by altering the triplet of numeric output of the three equations. It is a graphical representation is now known as a “Chaotic Attractor” (Fig. 4)

**The most important observation** within the Natural Evolutionary procedure through out the years, seems to be the randomization factor. Random mutations led to new genetic characteristics and Natural Selection decided if the characteristic will remain dominant to the next generations. Whenever the randomization exceeded a certain level, the subject would die out and could not survive. The mutations had to be minimal and the iteration process naturally selectable. On a philosophical perspective side note, one could say that whoever designed the Universe, has carefully designed a Randomization Engine as a prerequisite to its existence.



$$\frac{d}{dt}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}10(y-x)\\28x-y-xz\\-\frac{8}{3}z+xy\end{bmatrix},\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}-10\\10\\25\end{bmatrix},t=0\ldots30$$

**Fig. 4: Chaotic Attractor (source: Wikipedia)**

The implementation of a Natural Random Number Generation System, seems like an elusive target: *Since the introduction of "middle square" method by John von Neumann for the production of "pseudo-random" numbers in about 1949, hundreds of other methods have been introduced. While each may have some virtue a single uniformly superior method has not emerged. The problems of cyclical repetition and the need to pass statistical tests for randomness still leave the issue unresolved* [24].

**The idea** of implementing a Natural Random Number Generation System to the evolutionary process of a Genetic Algorithm is not new. The first element had to be a very carefully chosen Random Number Generation engine. A Chaotic Attractor could be used, but the procedure had to produce repetitions after a specific sequence of Chaotic numbers had been collected.

**Rule 110 (and possibly Rule 30) Cellular Automaton,** appears to provide the best option, since it qualifies the non repetition pattern, produces non predictable numbers, while its iteration procedure follows the minimal mutational factor synthesizing at the same time a proven Turing-Complete algorithm (Fig. 5).

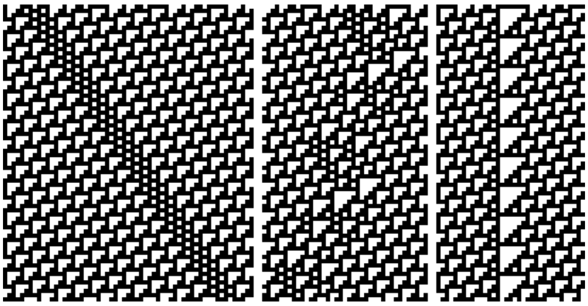


**Fig. 5: Rule 110 Cellular Automaton** [25]

Such an implementation, after a given time to evolve could theoretically form a Church–Turing–Deutsch principle machine. Blockchain signifies the ideal medium for such a task, since it forms an Unbounded Single Taped Turing Medium itself. The materialization doesn't necessarily demand heavy processing power, although such power could help the evolutionary procedure, via Proof of Work.

**Usability**. The utilization of such an algorithmic entity is literally limitless. Big Data Mining, Monte Carlo based Predictions, HMM predictions, Expansion of Human Knowledge, Protein Folding Prediction, Future Mutations in Human Genome, Intrusion Detection of websites, are a subset of the fields that could potentially benefit. The interaction with the entity could be done initially via scripting language, later with pseudo-code and finally even in natural language when the evolution matures to higher state.

It is trivial to state that commands need to be interpreted via respected Blockchain tokens (ie: Bitcoins, Ethereum, EOS etc) that would be used to get the questions into a Blockchain for the entity to process. The output will be given in the form of unspent transaction outputs (UTXO) on a Blockchain (Fig. 6). Practically, whoever owns the tokens will be able to utilize the processing power of the machine.

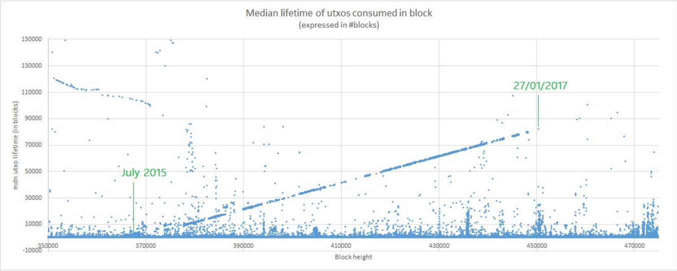


**Fig. 6: UTXO graphical representation (source: OXT.me)**

**Case Study**. Apart from the theoretical approach, there are evidential data, that such an entity is implemented on Bitcoin's Blockchain. Specifically, Bitcoin itself forms a "**Decider**" or a "**Two Stack Push Down Automaton**" (2pda) or, more formally, a special case of a Probabilistic Total Turing Machine, that is controllable via scripting language.

The most prominent evidential data is the graph presented in Fig. 6 which represents the Median lifetime of UTXOs consumed in blocks. Bitcoin UTXO Lifespan Prediction as shown by Robert Konrad & Stephen Pinto in 2015 is impossible to be modeled mathematically, since it is purely chaotic. Nevertheless, in the above graph there's a distinct linear formation within the phenomenal chaos [26].

At the same time period the Mempool transaction volume size had increased to up to 150Mb which adds an extra bit of confidence that the incidental linear formation could be the result of computable work (Fig.7)

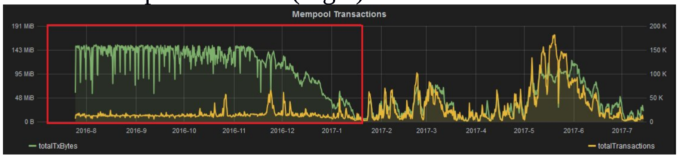


**Fig. 7: Mempool transaction volume in Mb (source: OXT.me)**

The Bitcoin scripting language is able to deploy a "two-argument Ackerman Function" which is now considered the simplest example of a well-defined total function which can be computable but not primitive recursive, providing a counterexample to the belief in the early 1900s that every computable function was also primitive recursive [27].

This is achieved via simulation of a "for loop" by the programming technique of "unrolling the loop" [28] and this system demonstrates that Bitcoin can incorporate total computable functions that are simply "recursive" as well as primitive recursive. One of the most interesting parts of this implementation is that a Bitcoin script can be constructed to simulate any decidable function. The added benefit to this comes from computational feasibility. It is sufficient for the alt stack to be a memory register in order to implement looping. Loops can be unwound this way and operate linearly [29].

Bitcoin script forms a non-synchronous language in the following way:

- The notion of physical time is replaced with the notion of order.

- Only the simultaneity and precedence of events are considered. This means that the physical time does not play any special role. This is defined as multiform notion of time.

- A false is not wrong, it is a script object that has been denied access to the sequence because for example it was beaten by another script to a critical resource.  In this way it is possible for a large amount of scripts running with various inputs.

- The first one to end as True is the one that is written to the blockchain.

- The other threads are removed.

This tactic is similarly implemented in CuDa and the threaded CuDa programs [30].

So to summarize, one can perceive the **2pda** as a distributed computer. This computer is the sum of all the full nodes. Every full node incorporates a full copy of blockchain (about 150GB at the time of writing) and its respective memory pool. This entity, as a distributed computer processes transactions (Tx), which have the OP_CODEs embedded. Two commands worth mentioning are OP_TOALTSTACK and OP_FROMALTSTACK.

The **OP_TOALTSTACK**: The opcode that follows this command defines a new function to be called. If this function name is already taken, the transaction is marked as invalid. Within the transaction, the function can be called simply as FunctionName until the process completes and returns the respective result.

The **OP_FROMALTSTACK** : This ends a function and returns. So, from these two OP_CODE commands, you can derive simple recursive functions [29]. There are more OP_CODE commands that are enumerated via the enum opcodetype command [31] but only nineteen (19) of them are correlated to the 2pda (List 1).

```
// stack ops

OP_TOALTSTACK = 0x6b,
OP_FROMALTSTACK = 0x6c,
OP_2DROP = 0x6d,
OP_2DUP = 0x6e,
OP_3DUP = 0x6f,
OP_2OVER = 0x70,
OP_2ROT = 0x71,
OP_2SWAP = 0x72,
OP_IFDUP = 0x73,
OP_DEPTH = 0x74,
OP_DROP = 0x75,
OP_DUP = 0x76,
OP_NIP = 0x77,
OP_OVER = 0x78,
OP_PICK = 0x79,
OP_ROLL = 0x7a,
OP_ROT = 0x7b,
OP_SWAP = 0x7c,
OP_TUCK = 0x7d,
```

**List 1: OPCODE list correlated to 2PDA** [31]

**Opportunities and Future Uses**. Bitcoin's creation in 2009 was a revolutionary idea in the financial world. It is considered as the digital cash of the new age. Secure, non centralized, can provide the world with "honest", non inflatable money. Game theory is utilized into maintaining consensus, without the need of any central authority, while Gresham's Law in effect would eliminate the "bad money" over the "good"; or if you prefer, the "strong" money.

Implementing a Church–Turing–Deutsch principle machine on a Blockchain, could in turn, open a whole new World of applications for a better humanity from Computer Assisted Governance to Extinction Level Events predictions. With emergent technologies like Human-Machine interface, such an entity could provide extensive knowledge in many fields of Science, which was previously impossible to acquire. Using Deep Machine Learning techniques, the evolution level of the algorithm could reach unprecedented levels exponentially, by utilizing the big data acquired by Smart Contracts, everyday transactions, weather conditions, or stored literature on a Blockchain.

## III. Conclusions

In this paper we showed that it is theoretically possible for a Turing-Complete algorithm, like a Cellular Automaton based on rule 110, to be implemented on an Unbounded Single Taped Turing Medium such as a Blockchain.

The implementation could be achieved by authoring a Genetic Algorithm that would evolve, by utilizing an, as close as possible, Naturally Random Generated Mutational System. The iteration process would be based on a Blockchain transaction system, and each entity could store itself when the maximum fitness level was due. We showed that the specific Algorithm should be Turing-Complete in order; as a Swarm Intelligence, to evolve to a Church–Turing–Deutsch principle machine.

The interaction with such an entity, could be achieved via interpreted commands using the transaction system. For this, Blockchain Tokens (ie: coins) will be used as a means of transaction. At the first stages of evolution the system would provide low-level programming support, but could be educated through Machine Learning to accept natural language interaction.

The advantage of implementing such an entity on a blockchain is primarily that it provides a theoretical representation of an Unbounded Single Taped Turing Medium. Secondly, that such a medium can be designed to provide fast iteration mechanism through recorded Tx. Some existing blockchains have the ability of materialize up to thousands Tx per second [16].

The disadvantage is that the GA fitness procedure should be done externally (**External Oracles**) and Application Specific

Integrated Circuits (ASICs) are mandatory for the task [32], [33].

## IV. Discussion

The implications of such a hypothesis are enormous. Ray Kurzweil, has predicted that by the end of 2029 the world will possibly have one AI that matches human intelligence [34]. What we showed in this paper verifies this claim, and endorses the possibility this could happen much earlier. It must be stated that preliminary forms of such entities are already in existence [35], so it is not a matter of if, rather when this happens. The evolutionary process, from a certain point forth, follows an exponential curve. Hence, when the critical point of reaching human intelligence is met, then it is a matter of months or even days before it expands to much higher levels.

The materialization of such an entity on a Blockchain provides many pros and cons that we tried to describe in this paper. The encryption procedure together with the mandatory token usage, certifies that such an entity won’t be able to interact without a cost. This is both good and bad.

Finally, a point of discussion could be about “what happens next”? At this point, a reference to the great text of Isaac Asimov, “The Last Question” [36] is needed:

*Can this chaos not be reversed into the Universe once more? Can that not be done?*

**Acknowledgements:**
Athanasios P. Kakarountas, Craig S. Wright, Ian Grigg, George P. Papageorgiou, Joseph VaughnPerling, Vassilis P. Plagiannakos.
The author would also like to thank the professors of the Department of Computer Science & Biomedical Informatics of University of Thessaly:
Pantelis Bagos, Harilaos Sandalidis, Konstantinos Delimpasis, Ioannis Anagnostopoulos, Evripides Markou, Spathoulas Georgios and Braliou Georgia. Without their deep academic knowledge, their support and endorsement, this paper would never have been possible.

## References

[1] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2002). Molecular Biology of the Cell. Garland Science. ISBN 0-8153-3218-1. Chapter 5: DNA Replication Mechanisms

[2] Article: “What is DNA Replication?". yourgenome.org. Welcome Genome Campus. Retrieved 24 February 2017.

[3] Article: “Genetic Algorithms”, Mathworks, 2017.

[4] John von Neumann, "The general and logical theory of automata," in L.A. Jeffress, ed., Cerebral Mechanisms in Behavior – The Hixon Symposium, John Wiley & Sons, New York, 1951, pp. 1–31.

[5] Gardner, Martin (1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"". Scientific American (223): 120–123.

[6] Wolfram, Stephen (1983). "Statistical Mechanics of Cellular Automata". Reviews of Modern Physics. 55 (3): 601–644.

[7] Wolfram, Stephen (2002). “A New Kind of Science”. ISBN 1-57955-008-8

[8] Michael Sipser (1997). Introduction to the Theory of Computation. PWS Publishing. ISBN 0-534-94728-X. Part Two: Computability Theory, Chapters 3–6, pp. 123–222.

[9] Church, A. (1932). "A set of postulates for the foundation of logic". Annals of Mathematics. Series 2. 33 (2): 346–366. JSTOR 1968337. doi:10.2307/1968337.

[10] Rabin, Michael O. (June 2012). Turing, Church, Gödel, Computability, Complexity and Randomization: A Personal View.

[11] Nielsen, Michael. "Interesting problems: The Church–Turing–Deutsch Principle". Retrieved 10 May 2014.

[12] Deutsch, D. (1985). "Quantum theory, the Church–Turing principle and the universal quantum computer" (PDF). Proceedings of the Royal Society. London. 400: 97–117. doi:10.1098/rspa.1985.0070.

[13] Satoshi Nakamoto, (2008) “Bitcoin: A Peer-to-Peer Electronic Cash System ” (Whitepaper)

[14] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker Stefan Savage (2013): A fistful of bitcoins: characterizing payments among men with no names (University of California, San Diego - George Mason University†)

[15] D.M. Schinianakis, A.P. Fournaris, H.E. Michail, A.P. Kakarountas and T. Stouraitis, “An RNS Architecture of an Fp Elliptic Curve Point Multiplier”, IEEE Transactions on Circuits and Systems I, vol.56, no.6, pp. 1202-1213, June 2009.

[16] Lester S. Hill, Cryptography in an Algebraic Alphabet, The American Mathematical Monthly Vol.36, June–July 1929, pp. 306–312.

[17] Papageorgiou, George P. (2013) Bitness: Bitcoin usage in an enterprise environment, opportunities and challenges (MBA Thesis, University of Greenwich)

[18] Ian Grigg (2017). EOS, An Introduction. (Whitepaper) iang.org/papers/EOS_An_Introduction.pdf

[19] Cook, Matthew (2004). "Universality in Elementary Cellular Automata". Complex Systems. 15 (1). ISSN 0891-2513. Retrieved 24 June 2015.

[20] M.G. Epitropakis, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis, (2011): Enhancing Differential Evolution Utilizing Proximity-based Mutation Operators IEEE Transactions on Evolutionary Computation, Vol. 15, 99-119.

[21] Tsoulos, Ioannis. (2009). Tsoulos, I.G.: Solving constrained optimization problems using a novel genetic algorithm. Appl. Math. Comput. 208, 273-283. Applied Mathematics and Computation. 208. 273-283. 10.1016/j.amc.2008.12.002.

[22] Ebrahimzadeh, Reza & Jampour, Mahdi. (2013). Chaotic Genetic Algorithm based on Lorenz Chaotic System for Optimization Problems. International Journal of Intelligent Systems and Applications. 5. 19-24. 10.5815/ijisa.2013.05.03.

[23] Danforth, Christopher M. (April 2013). "Chaos in an Atmosphere Hanging on a Wall". Mathematics of Planet Earth 2013. Retrieved 4 April 2013.

[24] Yadolah Dodge (Dec., 1996), A Natural Random Number Generator

International Statistical Review / Revue Internationale de Statistique Vol. 64, No. 3 pp. 329-344

[25] By JohnnyNyquist - http://commons.wikimedia.org/wiki/File:Ca110-structures.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=18642040

[26] Robert Konrad & Stephen Pinto, (Dec. 2015) “ Bitcoin UTXO Lifespan Prediction”

[27] Dötzel, G. (1991) "A Function to End All Functions." Algorithm: Recreational Programming 2.4, 16-17

[28] K. Aisopos, A.P. Kakarountas, H. Michail, C.E. Goutis, “High throughput implementation of the new Secure Hash Algorithm through partial unrolling”, in Proc. of IEEE 2005 International Workshop on Signal Processing Systems (SiPS’05), Athens, Greece, pp. 99-103, Nov. 2-4, 2005.

[29] Wright, Craig, (2017): “Bitcoin, A Total Turing Machine” (Whitepaper).

[30] CUDA® is a parallel computing platform and programming model invented by NVIDIA.

[31] Opcode List in Bitcoin Core (Version 14.0) https://github.com/bitcoin/bitcoin/blob/v0.14.0/src/script/script.h#L46L187

[32] A. Milidonis, N. Alachiotis, V. Porpodas, H. Michail, G. Panagiotakopoulos, A.P. Kakarountas, C.E. Goutis, “Decoupled Processors Architecture for Accelerating Data Intensive Applications using Scratch-Pad Memory Hierarchy”, Journal of Signal Processing Systems. Springer Science + Business Media, LLC, vol. 59, no.3, pp. 281-296, 2010.

[33] H.E. Michail, A.P. Kakarountas, A.S. Milidonis, G.A. Panagiotakopoulos, V.N. Thanasoulis, C. E.Goutis, “Temporal and System Level Modifications for High Speed VLSI Implementations of Cryptographic Core” in Proc. of the IEEE 2006 International Conference on Electronics, Circuits and Systems (ICECS’06), Nice, France, pp. 1180-1183, Dec. 2006.

[34] Ray Kurzweil (Dec 19, 2014), Don’t Fear Artificial Intelligence. http://time.com/3641921/dont-fear-artificial-intelligence/

[35] Watson, AI For business (2016, IBM). https://www.ibm.com/watson/

[36] Asimov, Isaac (November 1956). "The Last Question". Science Fiction Quarterly.