IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier unknown

Computer Science Abstractions To Help Reason About Decentralized Stablecoin Design

BEN CHAROENWONG[1], ROBERT M. KIRBY[2], (SENIOR MEMBER, IEEE) , AND JONATHAN REITER[3]

[1]National University of Singapore, Singapore (e-mail: ben.charoenwong@nus.edu.sg)
[2]School of Computing, University of Utah, Salt Lake City, USA (e-mail: kirby@cs.utah.edu)
[3]Data Finnovation, Singapore (e-mail: jonathan.reiter@datafinnovation.com)

Corresponding author: Ben Charoenwong (e-mail: ben.charoenwong@nus.edu.sg).

ABSTRACT Computer science as a discipline is known for its penchant for using abstractions as a tool for reasoning. It is no surprise that computer science might have something valuable to lend to the world of decentralized stablecoin design, as it is in fact a "computing" problem. In this paper, we examine the possibility of a decentralized and capital-efficient stablecoin using smart contracts that algorithmically trade to maintain stability and study the potential new functionality that smart contracts enable. By exploiting traditional abstractions from computer science, we show that a capital-efficient algorithmic stablecoin cannot be provably stable. Additionally, we provide a formal exposition of the workings of Central Bank Digital Currencies, connecting this to the space of possible stablecoin designs. We then discuss several outstanding conjectures from both academics and practitioners and finally highlight the regulatory similarities between money-market funds and working stablecoins. Our work builds upon the current and growing interplay between the realms of engineering and financial services, and it also demonstrates how ways of thinking as a computer scientist can aid practitioners. We believe this research is vital for understanding and developing the future of financial technology.

Keywords: smart contracts, algorithmic stablecoin, financial stability, DeFi, cryptocurrency

I. INTRODUCTION

As computing becomes more ubiquitous and pervasive, ideas from computer science will increasingly be relevant from Main Street to Wall Street. The power of computer science as a discipline is rooted in its propensity to observe what it thinks of as "specializations" from which it derives abstractions. In the language of object-oriented programming, every instance must derive from some "base class." The world in which we live and interact maintains our wonder through its specializations; however, it often derives its efficiency and predictability through its abstractions.

The field of economics broadly, particularly finance, has always had a close relationship with computing, from optimizing portfolios to the pricing of exotic derivatives using numerical methods. More recently, the rise of blockchain technology and smart contracts has brought a new opportunity for computer science and finance to cooperate.

Advances in financial technology and cryptography have introduced a new asset class called "stablecoins" in decentralized finance (DeFi) [1]. As assets that attempt to connect the DeFi world to the fiat world (and hence indirectly the computer science world to the economic world) by providing a token one can use in DeFi, but with a value pegged to a fiat currency, stablecoins seek to provide an asset in DeFi with a more stable valuation compared to other non-backed, non-managed tokens [2], [3], [4]. There is clearly market demand, as stablecoins collectively have over US$100 billion outstanding as of late 2021 [5].

While stablecoins resemble pegged fiat currencies, there are notable differences. For one, compared to human-controlled monetary authorities, stablecoins seem able to maintain stability algorithmically via a transparent and decentralized smart contract. It is upon hearing the word "algorithmically" that the computer scientists' interests are piqued, and something which we will explore in detail as our work

unfolds.

Before the DeFi revival of stable currency management, the design of managed currency regimes and their failure modes have been well-studied in the international economics literature [6], [7], [8]. The "Mundell-Fleming" framework – the first macroeconomic model to incorporate international capital flows and the foundational model of most modern international economic analyses – states that only arrangements similar to currency boards are stable. A currency board maintains stability and free capital flows but requires a large amount of the reference currency to be locked up forever to back the entire stablecoin monetary base.[1] With over US$100 billion worth of stablecoins circulating today, this approach would represent a significant amount of idle capital in the DeFi space. Therefore, to address this inefficiency, efforts have been made to design stablecoins that are stable without being fully backed by fiat currency reserves [10], [11], [12], but it remains unclear if this is viable in practice.[2]

With the DeFi revolution and renewed interest in the computational aspects of stable currency management and adoption, as surveyed by [15], there is a need for computer science and finance to partner once again to reason about the properties of these systems. In this paper, we explore whether it is possible to design a smart contract capable of maintaining stability without a trusted party. Initial attempts to solve this problem were unsuccessful [16]. However, when one uses abstractions from computer science theory, the problem reduces to reasoning about patterns frequently encountered in other fields (hence highlighting, again, the power of abstractions). We show that a solution for provable stability is possible in general, but the only provably-stable design requires full fiat backing. In other words, no capital-efficient algorithmic stablecoin can ever be impervious to a so-called "death spiral" [17]. This has significant implications for the design of both stablecoins and decentralized payment systems.

While this result may be disappointing in that it rules out extremely low-capital algorithmic stablecoins, it is also empowering in that architectures can now be, provably, maximally efficient. The result establishes clear requirements for many business models that trustless, decentralized protocols cannot reliably transact physical off-chain goods [18]. Our findings show a path for competition among stablecoins in the pricing, development of features, and other unconstrained arenas. Further, our computational approach provides a framework for more efficient and optimized stablecoin architectures to be designed.

Beyond our specific contributions around stablecoins expressed above, our broader contribution is to show how, once again, combining insights from computer science and economics can tackle new and emerging problems in computational finance. We hope that for both the computer scientist and economist – although our results might seem "obvious" as laid out from within the narrow lens of their field – there is value in seeking to cross-fertilize between these fields. Only in doing so can we be prepared for the emerging DeFi world and all its vagaries.

Before giving the caveats of our approach, we acknowledge that this work sits at the interface of computer science and computational finance. As such, there might be concepts or nomenclature that is natural to the computer scientist but opaque to the economist and visa versa. We have attempted to make this work self-contained and readable by both groups, sometimes overemphasizing things at the expense of brevity or under-elaborating at the expense of transparency. The reader is encouraged to maintain a healthy sense of interdisciplinary tension as they engage this work.

A. CAVEATS OF OUR APPROACH

Our work is related to previous work discussing similar issues related to (monetary) stability [19], [20], [21] but we approach the problem from a computer science perspective and address whether a solution can be proved to exist – without the need for fully specifying how economic agents behave nor their beliefs, preferences, or incentives. Additionally, related research has found that stablecoins currently observed in practice contribute to instability in DeFi [22]. Our computability result is consistent with those arguments.

Additionally, a caveat to our results is that the theoretical results pertain to the computational feasibility and provability of the property. It does not imply that a given design will not work for some (possibly very long) time. It simply means we cannot know that it will always maintain stability. In the case of stablecoins, where we have known-working and well-trodden designs available, we believe it is essential to clearly distinguish between things that "do work" and "might work."

B. OUTLINE OF OUR APPROACH

Our approach is as follows. First, we start in computational finance by presenting in Section II a product with definitions taken from an economic context. This product may or may not have some set of desirable properties. Then in Section III, we transition to the language of computer science and cast the economic problems into a computational one, then apply results from computation and graph theory to transform the problem, using various isomorphisms, into common computer science abstracts about which we can reason. Whether the product has the desired property is then formulated into a computer science question. Based upon our transformations and subsequent analysis, we then show that constructing a product that provably satisfies the desired property outlined in Section III is impossible. In Section IV, we present stablecoin designs influenced by our analysis and based on some economic characteristics of interest. In Section V, we use our framework and main results to discuss outstanding conjectures about stablecoin characteristics established by

[1] One example of this is the Hong Kong Monetary Authority (HKMA), which since 1983 has run the Linked Exchange Rate System, fixing the exchange rate between the Hong Kong and United States dollars. The system requires the HKMA to hold over 400 billion USD worth of monetary reserves [9].

[2] The earliest stablecoin designs point to this as a goal [13], [14].

academics and practitioners working on related topics. Finally, in Section VI, we discuss regulatory implications and conclude.

II. SETUP & DEFINITIONS

We start by making two basic assumptions so as to speak to a wide range of potential stablecoin designs. We will couch our nomenclature in the language of computational finance with attempts to connect it to computer science when necessary. The first assumption speaks to the informational efficiency of the stablecoin market, and the second is a simplifying assumption on the market structure, which applies to a broad set of stablecoins.

Assumption 1. *The quantity of risk-free arbitrage profit available in the market at any time is a randomly distributed non-negative number* a. $\forall_{n<\infty} 0 \leq \Pr(a < n) < 1$.

This assumption is weaker than informational efficiency in financial markets [23]. Rather than assuming the market reflects all available public or private information, we simply need that given any positive number n, the probability of this much arbitrage profit existing for a unit length of time right now is < 1.

Assumption 2. *Markets operate as a series of auctions. Each round requires a unit amount of time. Furthermore, parties obey the assumptions of Section 3.3.7 from [24].*

These assumptions lay an intuitive basic framework for how markets function as we expect: participants are at least somewhat rational, have reasonable preferences, have finite budget constraints, and generally look like standard economic actors [25]. The first assumption states that we cannot find unbounded amounts of free money in the market, and the second states that we cannot raise an unbounded amount of financing in a finite time.[3]

These assumptions are generally weaker than those required in standard economic models. Because we are not analyzing stable equilibrium conditions, we do not need to explicitly make assumptions about the expectations or incentives of the market participants. Models requiring economic equilibrium (either implicitly or explicitly) typically acknowledge when the state of the world somehow transitions away from equilibrium – what is technically called off-equilibrium paths or colloquially referred to as a crisis or black swan event – results may change and no longer hold. However, in the economic problems we seek to study – those concerning a trustless, decentralized system – we by definition, cannot rely on exogenous interventions to push things back into stability. As such, none of the policy considerations typically studied in these types of economic models would apply [8]. Most importantly, due to the decentralized nature of the payment system which we consider, these assumptions explicitly rule out fiat or "government monopoly" currencies [26], [27], [28].

We adopt a generic model which imposes few limitations on admissible designs. We follow extant research – taking the stance that "Stablecoins are a class of crypto-assets created to provide the stability money needs to function. As the name implies, they are designed to be price stable with respect to some reference point, such as USD" [29]. As such, without loss of generality, we consider this price to be one against some numeraire. We then define a stablecoin as follows:

Definition 1 (Stablecoin)**.** *A token* s *is a* stablecoin *when there is an associated smart contract* $\mathbb{C}$ *with a treasury* $\mathbb{T}$ *that can always exchange* s *at a fixed rate for some other asset within a finite number of blocks* b*. This same smart contract is also the monopoly issuer of* s*.*

Our setup is consistent with the original intent of decentralized blockchain tokens [30] and intentionally does not admit anything like "quantitative easing" or the monetization of government debt [31]. Instead, we explore the question of whether certain desirable products remain feasible in an environment without these tools.

The requirement to redeem at the correct price within a fixed number of blocks is important. Allowing unbounded time for redemption means everything can qualify as a stablecoin. It is not only important that a token can be redeemed at the correct price, it must also do so "quickly" (within b blocks). Our focus on avoiding unbounded delays in redemptions – possibly through intervention or other extraordinary action – is also explicit in existing financial regulations addressing bank deposit insurance provider resolution planning [32], [33]. A focus on the provision of immediate liquidity in a crisis extends back at least to the 1800s [34].

We further require a single monopoly issuer. If there are several blocks of code that can mint, but they are all controlled by the same issuer, we can take these to be different facets of the same smart contract without loss of generality. The essential feature is that there can only be a single coordinating issuer. Later, we will explore the consequences of potentially-uncontrolled printing of s and see why this restriction is necessary. Then, we define a DeFi stablecoin as:

Definition 2 (DeFi Stablecoin)**.** *A token* s *is a* DeFi stablecoin *when (1) it is a* stablecoin*, (2) it runs on a decentralized, permissionless platform and (3) both the stablecoin and platform are not under the control of any trusted party.*

The treasury $\mathbb{T}$ contains n assets, each with balance T_i for $i \in \{1, \ldots, n\}$. The balance at time t is $T_{n,t}$. The prices of these assets is represented as a vector P with elements p_i. Then, the value of our smart contract's treasury given a vector of prices is $V_t = P_t \cdot T_t$. We denote the number of stablecoin tokens outstanding at time t as s_t. This setup allows us to

[3] As an alternative to this stylized assumption, if one truly wishes to start from first principles, we can instead assume a finite money supply and finite system-wide transaction throughput. As this result already requires synthesizing disparate fields, we have tried to avoid reasoning where we begin in computer science, move to economics, and then back to computer science.

define a particular type of stablecoin with a strong connection to traditional finance:

Definition 3 (Currency Board Stablecoin)**.** *We say a stablecoin is a* currency board *when the reserves consist entirely of the target asset and* $\forall_t\ V_t = T_{target} \geq s_t$.

We use the inequality purposely to admit policies such as those outlined in [9] which seek to provide a buffer over and above a one-to-one backing. This is in line with standard definitions [35].

With this setup in place, we now move the computer science "core" of our paper and see how computer science abstractions can aid us in reasoning about these products.

III. MODELING OF STABLECOINS

An economy is a collection of agents transacting with financial assets.[4] In this section, we will first use results from the theory of computation to show limits on a particular agent and then use network theory to extend those limits to finite networks of agents interacting together.

A. THE SINGLE CONTRACT CASE: A VERTEX IN OUR NETWORK

A smart contract is a single program and is considered a single agent. The stablecoin problem is that an agent can take in x units of asset A and issue $y \geq x$ units of asset B while maintaining a stable price (e.g., that there exists a constant factor α such that $\pi_t(A) = \alpha\pi_t(B)\ \forall\ t$ where π denotes the price of a given A.). The major challenge for the single smart contract is that prior to execution, we must determine the specifiable constraints that guarantee that $\pi(A) = \alpha\pi(B)$. Here α represents the relative price and need not be 1. For example, the Hong Kong Dollar has long been "pegged" to the US Dollar at a rate of approximately 7.75. Without loss of generality, we will consider the $\alpha = 1$ case.

What remains to consider is the relative size of x and y. We will show that in the general case, maintaining a fixed price is an undecidable problem (in the theory of computation sense). For the case $x = y$, we can show that this is decidable. We show that when $y < x$, there are no issues; however, when $y > x$, which is the economically interesting case, it is undecidable.

Undecidability within Smart Contracts

If we show that the issuance is undecidable, it means no algorithm can always determine whether a stablecoin will maintain its peg per the definition above. In other words, it is not provably stable (where stability here is in the monetary finance sense). We will approach undecidability in two steps.

First, we will show how composing the time bound b from our stablecoin definition and our market-structure assumptions place certainty out of reach. We will then remove the market-structure assumptions and provide a more general reduction-to-halting impossibility proof. The idea is to show that the uncertainty in the system cannot be fully removed from the system for computer science reasons. And, further, that even mild economic assumptions can strengthen this impossibility.

Theorem 1. *If a* DeFi stablecoin *is not a* currency board*, then it cannot honor the* $s = 1$ *peg with probability 1 within* b *blocks.*

The proof relies on the market-structure assumptions.

Proof. We are considering whether this DeFi stablecoin can turn $\mathbb{T}$ into $|s_t|$ units of the numeraire within b blocks of time.

Without loss of generality, we take the quantity of arbitrage profit to be 0.[5] This follows the auction set up in [24], where the question is whether the smart contract can raise enough tokens within some constant time b.

Theorem 3.9 from [24] proves that the maximum revenue we can raise from a single round of auction is some function $f(t, x)$ where x contains the preferences and resources of other market participants at time t.

We have no control over $f(t, x)$ because we have a *DeFi stablecoin*. If we run b auctions at successive times, we can at most raise revenue

$$R = \sum_{i=1}^{b} f(t_i, x_i). \tag{1}$$

Our smart contract supports a stablecoin $\iff$ revenue $R \geq |s_t|$. But we do not control R. Therefore, this condition cannot be true with probability 1. □

We now provide a reduction-to-halting process to show how verifying the reliability of more complex constructions is also undecidable. The question is whether the smart contract can raise enough tokens within some constant time b. To that end, modify a proposed DeFi stablecoin smart contract by adding a counter variable c initialized to 0. For each location where an external revenue-raising call is made, increment c. If the counter reaches b, then, after the external call, enter an infinite loop.

Proving that this program halts requires proving the arbitrary code in those "external revenue raising calls" halts. This is the halting problem: we cannot build such a proving agent [39]. Therefore, no such stablecoin will surely work. In other words, can never prove your more complex scheme will always work because that would run afoul of fundamental results in computing.[6]

This does not depend on auction theory at all – it simply relies on the system needing to do something when asked to

[4]We do not take a stance as to whether such assets are necessarily "money" [36], [37], [38], but we only assume the financial assets are traded for goods or services.

[5]An alternative assumption yielding the same results is that the treasury already absorbs all arbitrage profits.

[6]Note that the smart contract modification employed is similar to the proofs of Rice's Theorem given in [40], [41], [42]. This technique is called the "classical proof" in lecture notes such as [43]. A shorter, more casual approach to this result is simply "automatically proving compliance with the time bound is impossible because of Rice's Theorem."

redeem $y > x$ tokens. We still need Assumption 1 to rule out a function that reliably goes out and generates $y - x > 0$ profit within b units of time. However, we need not care what it does instead.

The essential elements here are the time bound, interaction with arbitrary code, and a lack of riskless arbitrage profits. What happens if we remove the arbitrary code? Then we no longer have a halting-like decidability question or a decentralized platform. In this case, we are now reasoning about a known static program and asking whether it can convert x units into $y > x$ units while maintaining $\pi(A) = \pi(B)$. Such a program violates Assumption 1 as it can generate $y - x$ profits with certainty and is therefore not admitted within our framework.

Note this is not a circular argument where our efficiency assumption mechanically yields the conclusion. All it does is convert the assessment of decentralized stablecoins into a satisfiability problem which we can then prove, using standard techniques, is undecidable.

Our assumption purposely rules out a simple centralized algorithm that can generate riskless profits, as that is not the nature of the problem of interest. Instead, we focus on how a dynamic permissionless, decentralized ecosystem need not necessarily provide solutions to age-old problems because such environments are still bound by halting impossibility.

Seen another way, finding a closed, static algorithm that can support $y > x$ tokens is equivalent to finding a riskless source of unbounded trading profits.

B. THE MULTI-CONTRACT CASE: A NETWORKS OF CONTRACTS

The previous section considered whether it is possible to construct a single algorithm – a single "smart contract" in decentralized finance (DeFi) – that can support y stablecoins given only $x < y$ backing units. Now we will show that constructing a network of such contracts does not increase the system's ability to handle the $y > x$ case.

We model assets flowing within an economy as a graph flow problem with instantaneous flows. The reason we view flows as instantaneous as opposed to as a sequential problem is due to the transparency of a public decentralized ledger. In contrast, in the traditional world, where each party has its non-public ledger, we can imagine a process where someone tries to check a reserve quantity by calling or visiting different bank branches sequentially. Each inspection takes time, possibly a different amount of time at each bank, and there is no single way to see everything at once. There is latency, and a clever operator may be able to maintain the illusion of a higher backing for some period by transferring funds over and over. However, this does not apply to a transparent public ledger where we can see the full state of the system all at once. Every system observer can retrieve a complete set of balances and check for themselves. Therefore, asset flows in such a system look more like instantaneously-flowing fluid [44], [45].

The flow network setup includes a source that contains s units as backing, and the question is whether we can extract $l > s$ units from a network of contracts through the sink (or any other network bisection). In practice, sources are minting nodes, and sinks are "burning" (consuming) nodes. Our previous analyses show that individual nodes cannot reliably synthesize flow via some algorithm, raising natural follow-up questions:

- Can we sustain $l > s$ given reserves stored within nodes on the network?
- Can we sustain elevated flow through any useful subset of the network?
- Is there a partition of the network such that flow across the partition boundary exceeds that out of the source?

To study whether we can get more out of a network than was put into it, we must study whether there is any configuration of any network and any cut across such a network where the flow exceeds that out of the source or into the sink. A consequential network flow theorem called the "max-flow min-cut" theorem bounds the maximum flows. Much of our contribution here lies in formalizing the stablecoin problem in a manner amenable to such analysis.

By applying known results in network theory to this new financial market's problem of stablecoin stability, we can evaluate whether there exists a configuration of smart contracts that can collectively service withdrawals of y tokens given only $x < y$ tokens to start.

1) Graph Structure of the Network

In the standard graph formulation of [44], a graph G consists of vertices V and edges E. One vertex s is the "source" and originates all flow. Similarly, t is a sink into which all flow ends. If we denote the flow between u and v as $f(u, v)$ then we have:

$$\forall_{u \in V - \{s,l\}} \sum_{v \in V} f(u, v) = 0. \tag{2}$$

This tells us that, for every vertex except $\{s, t\}$, the flow in equals the flow out. In other words, traded financial assets must be balanced if we do not account for burning or minting. We think of the source as holding our reserves to back the traded financial assets and the sink as withdrawal from the system. Without loss of generality, we consider flow networks where the $f(u, v)$ capacities are equal to the flow required for those vertex pairs to maximize flow from source to sink. We can see from the proof of the max-flow min-cut theorem in [45] that eliminating excess capacity does not reduce the maximum flow through the network.[7]

2) Per-Vertex Reserves

One natural extension to the model is to permit flow to be stored in, or consumed by, vertices other than s, t. Exercises

[7]This sort of excess capacity is called "augmented flow" in [45] and "f-incrementing" in [44].

in both [44] and [45] and elsewhere consider a more general version with positive capacity as:

$$\forall_{u \in V-\{s,l\}} \sum_{v \in V} f(u,v) = u_c < 0. \tag{3}$$

This is called "positive" because there is more flow into u than out of it. We will extend this slightly to allow both positive and negative flows.

$$\forall_{u \in V-\{s,l\}} \sum_{v \in V} f(u,v) = u_c. \tag{4}$$

Negative capacities correspond to flow emerging from the vertex. [8] If $u_c > 0$ the vertex is generating some flow, and if $u_c < 0$ the vertex is storing it. This, it would seem, may provide a solution to our stablecoin problem.

With a small transformation, we see this is simply a sleight of hand on the backing quantity. We can see this adds no power to our graph as a set of these per-vertex constraints is the same as adding edges from $s \rightarrow u$ with capacity $\max(0, u_c)$ and from $u \rightarrow e$ with capacity $\max(0, -u_c)$ and then setting all u_c to 0.[9]

More formally, consider a graph with capacities. We call the flow out of the source:

$$s_0 = \sum_{u \in V-s} f(s,u). \tag{5}$$

Here there might be some way to use negative weights to extract more flow from the network as:

$$\sum_{u \in V-s} f(s,u) + \sum_{u \in V} \max(0, -u_c), \tag{6}$$

Now apply the above procedure to add edges and zero the u_c and call the new flow function $f'(u,v)$. On this new graph, the flow out of the source is:

$$s_1 = \sum_{u \in V-s} f'(s,u), \tag{7}$$

where for this graph, because of the edge-adding procedure which sets $\forall_{u \in V} u_c = 0$:

$$\sum_{u \in V} \max(0, -u_c) = 0, \tag{8}$$

where s_1 is the source flow. It may be possible to set up a configuration where the declared source only contains s_0 units, but we still need to find $s_1 - s_0$ from somewhere. Computationally, we have not achieved anything new.

Vertex capacity can slice up the backing – but we are still constrained to the quantity available out of s. As long as we define the quantum of backing as the total amount in the source, this constraint is binding.

[8] While this is uncommon in physical flow networks, it is of interest when considering financial assets where the "physical" cost of production is 0.

[9] Something like this approach is common as an exercise in graph theory courses. For example, [46], [47]. This is also the same technique employed in [44] to show that it is sufficient to study networks with a single source and sink and not to bother with a separate study of multiple-source or multiple-sink configurations. And we find similar generalizations in related algorithm design work for engineering, such as [48] and [49].

3) Layout of the Network

To develop intuition, lay the network out with the source on the left and the sink on the right. It is a flow network, so all the edges now run left to right. The max-flow min-cut theorem tells us that the flow across any network bisection that separates s from t is the max flow of the network. This means the flow is the same for every top-to-bottom line we can draw through our network.

This applies to any partitioning of the network where s is in one section and t in the other; the flow between sections is the max flow of the network. So there is no clever arrangement of nodes where we can somehow sustain elevated flow locally purely from layout.

We can describe this more formally in the style of [44]. Partition our graph into two subsets S and L where $s \in S$, $l \in L$, $|S| + |L| = |V|$, $S \cap L = \emptyset$ and $S \cup L = V$. From [44] we have that, $\forall_{u \in S}$ and $\forall_{v \in L}$:

$$\sum_{u \in S, v \in L} f(u,v) = \sum_{u \in V-s} f(s,u) = \sum_{u \in V-l} f(u,l) \tag{9}$$

There is no way of partitioning the graph to find more flow.

When we have instantaneous flow, as we do in standard flow network analysis, there is no solution which looks like the process of moving reserves among banks to remain one step ahead of the inspectors.

4) General Networks

Based on the results above, we know that (1) individual vertices cannot be proven to output more tokens than they take in reliably, and (2) that this result applies to networks of multiple vertices regardless of configuration.

It is possible to split up our "source of funds" s into several different pots $s = \{s_0, s_1, s_2, \dots\}$. And we can choose to label one of these as our "treasury." But this is just a semantic game that is easily seen through via the edge-adding procedure described above.

So long as we consider flows of funds to be instantaneous in the way that computer science traditionally treats network flow problems, there is no clever arrangement that solves the stablecoin problem in a more efficient manner than that provided by traditional finance.

This statement can be formalized with a simple proof by induction. The base case is a simple single-contract network with three vertices: a source that holds reserves, a smart contract that operates some stablecoin scheme, and a sink that services withdrawals. The halting-problem-derived proof establishes that no such network can provably always service $l > s$ withdrawals.

Then the induction step considers whether adding contract/vertex increases the quantum of withdrawals that can be handled. We know from our discussion of network theory that, as a consequence of the max-flow min-cut theorem, additional reserves bound additional withdrawal capacity in the guise of negative vertex capacities.

IV. DESIGNING A WORKING STABLECOIN

With our analysis above in place, we now ask the question: how might one design a working stablecoin that respects our results? This section presents one possible implementation of a working stablecoin:

function CREATESTABLE($fiat$)
  DEPOSITINTOBANK($fiat$)
  $balance \leftarrow balance + |fiat|$
  return PRINT($|fiat|$)
end function

function WITHDRAWSTABLE(s)
  BURN($|s|$)
  $balance \leftarrow balance - |s|$
  return WITHDRAWFROMBANK($|s|$)
end function

How do we know this works? And how do these functions interact with the results proved above? First, note that DEPOSITINTOBANK() and WITHDRAWFROMBANK() are not trustless activities. If there is a literal off-chain bank involved, then the system cannot be trustless [18]. For example, if we are operating on-chain with a Central Bank Digital Currency (CBDC), then the backing of the instrument, as discussed above, is dependent on trust in the Central Bank.

Second, there is no open-ended external function call here. To the extent we rely on uncertain activities (e.g., bank deposits or CBDC management), there is no pretense that they are trustless. We can prove this works with the assumption of trust in place. If we try to drop that assumption, the proof falls apart in line with our results above.

This algorithm says nothing about whether a fully-backed stablecoin that makes external function calls can work, or is decidable, or is even a good idea. We make no claims about such a product except to note that the complexity is unnecessary given the known-working simpler scheme presented here.[10]

We show that the traditional currency board is the only workable model to maintain a fiat currency peg, even on a decentralized blockchain with rich, smart contracting capabilities. Such a currency board must, because of our prior results, hold fiat backing in the form of ultimately-government-backed off-chain assets. We see products that are already structurally similar to money-market funds, although they currently operate under different sets of rules [50], [51]. More recently, Japan has passed a law stating that yen stablecoins must be backed only by yen cash and Japanese government bonds.

Stablecoins following different fiat currencies should fall under their respective governmental jurisdictions, curtailing the incentive for regulatory arbitrage. That is not so for many other products. The most common decentralized stablecoin designs in use today relies on a combination of algorithms and crypto-native collateral [52], [53], [54]. Such products are at risk if the backing collateral drops in value too quickly or if some component of an ecosystem goes offline. Other designs involve some combination of backing asset values, fee revenues, and variable funding rates, which can also exhaust resources leading to a failure [55], [56]. These eerily parallel the experience of runs on money-market funds in traditional finance, such as those discussed in [57]. Other variants only try to maintain the peg within a range and explicitly acknowledge failure modes while examining stability properties [58] or provide stability only with a closed ecosystem in a manner reminiscent of a restricted currency [59].

These are hard problems and, even in modern times, authorities sometimes require ad hoc actions through a trusted party [60]. In the case of traditional money-market funds, that trusted party is the government, usually in the form of the central bank or finance ministry such as that outlined in [61]. One implication of our present results is that stablecoins should be treated similarly. Money-market funds are a special collective investment vehicle that, in exchange for a narrow mandate and tight regulatory regime, receive additional support from central authorities.

Further, given the impossibility results presented here, such a framework should only admit working stablecoin designs. Perhaps some limited flexibility, such as the 99.5% requirement of [62], is appropriate. These are practical and fundamentally political questions for regulators beyond the scope of our discussion.

Whatever the correct answer, our results make plain that these questions around stablecoins are but another flavor of money-market fund regulation as truly-novel designs do not provably work, and large quantities of government-bill-like collateral are required. Whether the best design is a government-run CBDC or some money-market-fund-adjacent regulatory regime for private operators, or some hybrid of the two is a complex policy question. It would appear that combining those options is the only choice admitted by the theory.

It is also worth noting that we, as a society, have collective experience managing these problems without a transparent central authority. Before the advent of modern central banking governments were forced to rely on ad hoc deals with wealthy private parties in times of extreme crisis. We already know such arrangements are dangerous and can have unintended consequences for centuries. Reciting a few examples will make clear the trouble: John Jacob Astor and Stephen Girard during the War of 1812 [63], the Rothschild family during the Napoleonic Wars around the same time [64], and J.P. Morgan during the Panic of 1907 [65].

V. EXTENSION & IMPLICATIONS

In this section, we now delve into the DeFi world and discuss additional implications of our results.

[10]External function calls are still subject to both Halting and Rice. But in a fully-backed situation, we may be able to prove such code paths are never followed and the finite time bound is met. One can imagine designs that are undecidable and risky on purpose. It is not clear what such products would be good for. We present one working design and a consistent impossibility result ruling out a large class of other designs.

A. EXISTING IMPLEMENTATIONS

Practitioners have tried to modify basic stablecoin designs in a wide range of ways in pursuit of a working, more efficient product. Here we will explore a few common techniques and show how none truly resolve the issue highlighted above. We can transform the precise nature of the problem; however, because of the constraint of Rice's Theorem, it cannot be removed entirely.

1) Gas Fees

So far, we have ignored smart contract execution costs. Real platforms are not truly Turing-complete in that computation is constrained by a cost known as a "gas fee." If a program consumes all the gas before termination, it exits with an error, and the gas is paid.

While the Halting result does not necessarily apply because of these limitations, the gas fee cost makes the product infeasible. If an attempt to raise revenue fails because of inadequate gas, not only was no revenue raised, but the fee is lost. The value of $\mathbb{T}$ is lower after the attempt, and we have wasted one of our b chances. This is worse.

2) Hybrid Backing and Stop-Loss

One plausible design for a stablecoin is to hold some arbitrary collection of backing assets. Then, so long as their collective market value exceeds the quantity of stablecoin outstanding, it can be converted into the reference asset at a fixed backing ratio. This design follows a trading rule isomorphic to a "stop-loss" strategy [66]. While this approach is appealing, it does not work in general.

To see why, suppose we pursue a simple strategy where once the stop-loss is executed, we never convert back. Essentially, that is, running a currency board with the added risk of failure on stop-loss execution. Sufficient losses on the conversion into a reference asset, even though the trade occurs at most once, can render the stablecoin insolvent. If we clear that risk, the stablecoin is irrevocably a currency board.

Furthermore, the single-stop-loss approach is better than the multiple-trades alternative. A strategy that holds a collection of assets when the market value is above some constant and switches into the reference asset below that price will leak value over time [67]. This sort of "stop-loss start-gain" policy will erode the treasury value below that of the outstanding stablecoins over time. This is true even absent transaction costs. The key insight of the aforementioned paper is that these losses are not due to bid/offer spread or slippage or anything of that kind: they are a built-in feature of stop-loss based trading strategies.

We can also connect the operation of capital-efficient stablecoins to decentralized stop-loss guarantees. Intuitively these concepts come together in products like MakerDAO's DAI [53] and the Hubble Protocol [54]. And there is a discussion of losses-on-repegging [68]. The Tether whitepaper even raises concerns about stop-loss execution [69]:

> In the collateralization method, the market risk exists because the price of the asset being used as collateral can move in an adverse direction to the price of the asset it is backing. This would cause the total value of the collateral to become less than the value of the issued asset and make the system insolvent.

However, there is a deeper computational connection: guaranteeing stop-losses in DeFi is the same problem as designing an algorithmic stablecoin. We will now apply the standard computer science technique of reducing each problem to the other to prove the equivalence.

This connection is made explicit in the MakerDAO whitepaper:

> If the Collateral Auction does not raise enough Dai to cover the Vault,Äôs outstanding obligation, the deficit is converted into Protocol debt. Dai covers protocol debt in the Maker Buffer. If there is not enough Dai in the Buffer, the Protocol triggers a Debt Auction. During a Debt Auction, MKR is minted by the system (increasing the amount of MKR in circulation) and then sold to bidders for Dai. ... MKR dilution could reach extreme levels and still not bring enough liquidity and stability to the market.

This describes a chain of safeguard processes intended to keep the protocol alive. But, in the end, failure is still acknowledged as a possibility. We would also note the final line in that quote is from a section entitled "Unforeseen pricing errors and market irrationality." Irrationality is a well-known and long-studied feature of markets, epitomized in a quote attributed to the economist John Maynard Keynes that "the market can remain irrational longer than you can remain solvent." We surely agree with those authors that one cannot always foresee the drivers of upcoming irrationality — but we know it is coming eventually. This sentiment closely echos our connection between undecidability and risk.

Our reductions employ only a single layer and are considerably less byzantine than the MakerDAO process. A multi-layer scheme may be marginally safer in practice. Our point is rather that no amount of complexity can entirely excise the risk, and a single layer would suffice if one could properly construct the product.

First, assume we have a working algorithmic stablecoin that, with only holdings of some collateral token c, can guarantee a price of 1 USD for the stablecoin s. This kind of promise is made by the stablecoins referenced above (and others), and somewhere on the inside, there is a stop-loss liquidation order to sell c for USD. Then, we construct an algorithm around this product to expose that stop-loss. Consider this process when we wish to construct a long position in c with a guaranteed stop-loss in USD:

function LONGWITHGUARANTEE($c, level$)
  $stablecoin \leftarrow$ NEWSTABLECOIN($level$)
  $s \leftarrow$ DEPOSIT($stable, c$)

```
    while wish to remain long c with a stop-loss do
        if PRICE(c) ≤ level then          ▷ do not sell, redeem
            proceeds ← REDEEMUSD(stable, s)
            return proceeds
        else
            HODL
        end if
    end while
end function
```

This algorithm fails precisely when the stablecoin fails. We are not asserting this algorithm works in practice, but rather that if we could build the stablecoin, then this would work. The equivalence works the other way around as well:

```
function STABLECOINDEPOSIT(c)
    depo ← LONGWITHGUARANTEE2(c, $1)
    ADDTOTREASURY(depo)
    return PRINT(|depo|)
end function
function STABLECOINREDEMPTION(s)
    depo ← EXTRACTDEPOSITS(s)
    if STOPPEDOUT(depo) then          ▷ price below level
        return depo
    else                              ▷ price above level, selling is safe
        proceeds ← MARKETSELL(depo)
        surplus ← proceeds − |s|
        STABLECOINDEPOSIT(surplus)
        return |s|
    end if
end function
```

Note that there is nothing special about USD, and we can replace that symbol with any other asset. In addition, we are not using the LONGWITHGUARANTEE() algorithm written above as that would be circular. We assume there exists some other scheme LONGWITHGUARANTEE2() upon which we can rely. Along this vein, we can see these are the same problem.

We have now established that the only working stablecoin design is fully backed. This means that the second set of functions is not worth considering — we know they cannot be made to work. But what of the first? Unsurprisingly the challenge is the REDEEMUSD() function. We have no idea how to write that, and doubly so in the scenario where the price of c is below the stop-loss level. It is very much a form of financial alchemy, and it would be surprising if this were possible. As we have proven equivalence with a known impossible task, we now know this function cannot be written.

3) Arbitrary Credit Extension

If we have access to any of these functions, we can create arbitrary credit out of a single unit by recursive application of the algorithm. Suppose we have an over-collateralization requirement of ϵ for the backing but wish to achieve $\beta > \epsilon$. Then, consider this deposit mechanism:

```
function RECURSIVEDEPOSIT(c, ε, β)
    Stablecoin ← NEWSTABLECOIN(c)
    if ε ≥ β then                     ▷ enough recursion
        return DEPOSIT(Stablecoin, c)
    else                              ▷ stack another level of stablecoin
        s ← DEPOSIT(Stablecoin, c)
        δ ← ε × (1 + ε)
        return RECURSIVEDEPOSIT(s, δ, β)
    end if
end function
```

This stacks stablecoins until we achieve the desired level of efficiency. Sure, if $\epsilon = 1\%$ and $\beta = 50\%$ we are going to need $\log_{1.01} 1.5 \approx 41$ levels of recursion. But it is a decentralized, permissionless system. Such a thing may grow organically, and there is no way to stop it.

This sort of activity exists in the market today. For example, the Lido protocol is reported to warn users about the dangers of such strategies [70], [71], [72]. Even protocols such as FarmersOnly exist to automate these practices [73].

4) Lender of Last Resort and CBDCs

The concept of a lender of last resort (LOLR) goes back to [34] who was describing the United Kingdom and is a near-ubiquitous feature of banking systems around the world, including Australia [74], Canada [75], the Eurozone [76], Singapore [77], Sweden [78], Thailand [79] and the United States [80]. Such a party is capable of arbitrary fiat credit creation and therefore is in a computational sense equivalent to an algorithmic stablecoin. This is consistent with how a central bank is defined [35] and is far simpler than RECURSIVEDEPOSIT() as presented above. The algorithms are trivial:

```
function DEPOSIT(x)
    BURN(x)                           ▷ burn all incoming fiat
end function
function WITHDRAW(x)
    return PRINT(|x|)                 ▷ print fresh fiat for withdrawal
end function
```

Economic policies implemented in what is a centralized and permissioned environment can "control" how and when these functions are called.

The connection is, in fact, deeper. If a LOLR announces it will buy a certain asset, or class of assets, at a pre-specified price it is also providing the market with a guaranteed stop-loss. LONGWITHGUARANTEE() can be implemented in terms of the fully-implemented WITHDRAW() above as:

```
function LONGWITHGUARANTEELOLR(c, level)
    while wish to remain long c with a stop-loss do
        if PRICE(c) ≤ level then      ▷ if stop is triggered
            s ← WITHDRAW(level)       ▷ print the money
            return BUY(c, s)          ▷ and buy the asset
        else
            HODL
        end if
    end while
end function
```

We know these two problems are the same, so this is not particularly surprising. Rather, the point is that central banks employ this equivalence in the real world. This is one power in various guises. In some sense, different departments within a central bank implement the reduction algorithms presented here.

Through these reductions, we can see that providing LOLR-type functionality is inextricably linked with the existence of a party that can arbitrarily expand credit. And for the same Rice Theorem-based reasons given above, we cannot guarantee to employ these facilities only at most n times in a decentralized, permissionless environment. Consequently, we can have a LOLR if and only if we have possibly-unconstrained credit creation or, equivalently, a possibly-unconstrained money supply. We cannot have one without the other: we either get all of these features or none.

This is why we require a single issuer to coordinate the process. Otherwise, we need to trust another participant in a decentralized, permissionless system not to slip up and bring in all of these problems. Equivalently, we can publish an immutable protocol like Uniswap [81] and construe the original developers, with their code, as the issuer. Or we can trust that different facets of a stablecoin – say the smart contracts on different blockchains – are controlled by the same actors.

Beyond regulatory discussions of whether to treat digital assets as securities (e.g., in India), regulators have started to look into CBDCs and stablecoin designs, in general, [82], [83]. Within this framework, a CBDC is a variant of stablecoin where $\mathbb{T}$ contains an arbitrarily large quantity of fiat backing. That comes straight from the definition of a fiat currency [26]. Such a CBDC is then "stable" to the extent the central bank ensures $V_t \geq |s_t| \; \forall \; t$ in line with Definition 3 and our analysis in Section III.

This exposition makes clear that CBDCs are not fundamentally new. Satisfying that inequality is a policy choice managed off-chain by whoever controls the private keys to $\mathbb{T}$. The CBDC is still entirely under the control of the CB. It does, however, offer the possibility of engineering improvements in settlements, transparency, and other ancillary features [84], [85], [86], [87], [88].

A range of questions are raised in [89] regarding the degree of difference between CBDCs and useful stablecoins. Our results prove that the gap is narrow, with little space to innovate in stablecoin design. Any reliable stablecoin will look like a wrapper around central bank currency or government debt instruments regardless of whether such assets are tokenized on a blockchain, held book entry in a database, or recorded using any other technology.

B. ADDRESSING EXTANT CONJECTURES FROM ACADEMIA AND INDUSTRY

With our analysis from Section III in hand, we can resolve several open questions from the literature. These issues are raised in a wide variety of contexts, so we have divided this section up to break out conjectures from academics, blockchain practitioners, and stablecoin operators. Other sorts of stablecoins may work and may work for a long while. However, there is only one admissible design that always works.

1) Academic Conjectures

There are two conjectures phrased in the language of the long-term stability of an equilibrium [19]. Our present work concerns whether it is possible to design stablecoin systems where $\Pr(\text{failure}) = 0$. So long as that probability is non-zero — and we have proved it must always be non-zero — by Borel's Law of Large Numbers we will eventually see a state of the world which causes failure "in the long-term" [90]. Consequently, we consider that both are resolved through this work. Let us consider them in turn.

Conjecture 1 reads:

> In fully decentralized stablecoins (α = 0) with (i) multiple classes of interested parties (e.g., risk absorbers vs. stablecoin holders) and (ii) a high degree of flexibility in governance design, no equilibrium exists with long-term participation under realistic parameter values

We have proven that there is but a single type of working algorithmic stablecoin: the currency board. These vary only in the degree of their over-collateralization and do not admit multiple classes of interested parties.

Conjecture 2 reads:

> Considering fully decentralized systems (α = 0) with (i) multiple classes of interested parties and (ii) a high degree of flexibility in governance design, DeXs have a wider range of feasible long-term participation equilibria than stablecoins under realistic parameter values

As we have already established, there is but a single workable decentralized stablecoin design. Consequently, it suffices merely to provide two classes of DeX designs to prove the conjecture. Uniswap, and its many forks, qualify as a working DeX [81]. Balancer offers a more flexible framework in the same style [91]. 1Inch employs a somewhat more complex scheme with virtual balances [92]. Serum, a decentralized CLOB, is a completely different type of DeX [93]. Empirically we find the number of different designs is large. For example, TOK stablecoin's whitepaper states:

> The results would mostly confirm our conjecture that some legacy collateral with regulated assets needs to be incorporated in the new collateral pool where actual decomposition over legacy vs. crypto depends on simulation results and user preference [94].

We can resolve this conjecture in the case of a user preference for zero risk: the pool must consist entirely of trusted assets with direct connections to the underlying fiat issuer.

2) Blockchain Practitioner Conjectures

Practitioners have struggled with algorithmic stablecoin design problems for years now. More recently, both in the period leading to and immediately following the Terra collapse, community members have begun to ask questions and pose conjectures that are sufficiently precise to resolve clearly. The practitioner community has the right intuition — but things are somewhat more complex than they first appear.

There are two thought experiments presented in [95]. The results presented here allow us to provide simple, categorical answers to these questions. The first is "can the stablecoin, even in theory, safely 'wind down' to zero users?" We can now answer this question. The answer is: maybe, but the only way to do that with probability 1 is to maintain full backing as a currency board. The second question is "what happens if you try to peg the stablecoin to an index that goes up 20% per year?" We can also answer that question: unless the stablecoin is backed by assets that also go up 20% per year, it will eventually fail.

3) Stablecoin Operator Conjectures

A few similar conjectures are found in the Tether whitepaper:

> Finally, understand that we believe some combination of the above approaches [collateralization and derivatives] may become a secure, reliable, and generally risk-free process for backing/pegging assets; however, at this point in time, this is not a direction we feel is feasible to take to ensure liquidity and price stability. Further, we believe that a reserve-based approach will always be in existence and complement these other approaches as the entire industry grows [69].

Let us take those statements in order. First, we have shown that a combination of those approaches cannot provide the sort of risk-free process desired. However, (and this concerns the design only, not the implementation) they were correct to focus on the reserve-based design and eschew approaches we have proven unreliable. Moreover, they are likely correct that the reserve-based approach will be the last one standing as it is the only one which can offer long-term stability.

VI. DISCUSSION AND CONCLUSIONS

We now discuss the results of our findings from the perspective of computational finance and DeFi, and then conclude with some remarks from the perspective of computer science.

From the computational finance perspective, our finding is that while decentralization may open up certain new properties, something like a currency board is the only available mechanism in DeFi for pegging an exchange rate. Decentralized blockchain-powered smart contracts surely enable new products. Once we accept the requirement for $\geq 100\%$ reserves, we can still construct a reliable, censorship-resistant stablecoin. However, there is little freedom open to us in the design of such a product.

Overcollateralization does not fix this problem. We know from existing derivatives results that a trading strategy with a stop-loss at the level of full-backing will bleed value over time [67]. There is no way with probability 1 to ensure the treasury value is high enough absent a currency board.

Furthermore, this result does not limit product development for "probably stable" coins.[11] We can achieve this by introducing capital-control-adjacent measures, accepting a non-zero probability of the peg failing, burning coins held outside $\mathbb{T}$, or other mechanisms. One can talk about game theory in this context, but only after acknowledging real risks with maintaining stability. We can compare the level of decentralization of such products [99]. Similarly, Rice's Theorem prevents us from quantifying many of these measures when the smart contract language is Turing-complete [42]. We cannot say an algorithmic stablecoin has some known probability of failing because we cannot decide such questions.

Unfortunately, simplifying the smart contracting language does not entirely fix the problem. If our system, by construction, can only ever run b auctions, it can *never* provide a surely-pegged exchange rate unless it is a currency board. We may be able to provide a measure of reliability under certain restricted smart contract language setups as Rice's Theorem does not apply to *all* languages. This means there may be a way to characterize a given simplified language setup as "95% stable." But total reliability remains out of reach independent of language class.

Many groups have tried to design capital-efficient stablecoins. We find that, notwithstanding their engineering prowess, they would not succeed. Anyone interested in decentralized stablecoins would be better served to focus on features while accepting they will be fully reserved. In the same vein, centralized stablecoins can confidently assert their capital efficiency and again compete solely on the rest of their offering. Decentralized stablecoins are possible, but to be truly stable, they must be designed the same way as a traditional pegged currency and will, by induction, eventually rest atop a trusted party. Unfortunately, a blockchain-based smart contract environment does not admit solutions to the problem beyond those in traditional finance.

Lastly, this aversion to stop-losses (i.e. liquidating collateral during a crisis) is present in Bagehot's 1873 writing [34]:

> If it is known that the Bank of England is freely advancing on what in ordinary times is reckoned a good security, on what is then commonly pledged and easily convertible, the alarm of the solvent merchants and bankers will be stayed. But if securities really good and usually convertible, are refused

[11] These are called "metastablecoins" in a nod to physics [96]. To abuse the analogy slightly further: the idea that such products may have local minima above a "ground state" value of 0 is worth further investigation. We note in passing that this ground state in finance is far less sticky than in physics. Examples such as Hertz [97], [98] abound where investors pay well above zero for assets broadly considered worthless – only to find their investment, perhaps temporarily, can pay off. Whether these pre-programmed dynamics within finance could halt a panic remains unclear. However, these novel questions come about only because of permissionless smart contracting. Moreover, there may well be new mechanisms to design here somewhere.

> by the Bank, the alarm will not abate, the other loans made will fail in obtaining their end, and the panic will become worse and worse ...

And the solution then — the strategy to end-run this impossibility — was to employ trust [100]. This is a challenge. And perhaps nothing makes plainer the difficulties in managing this trust than wherein the Bank of England reflects on its performance on the 150^{th} anniversary of the Overend-Gurney crisis, which originally inspired Bagehot's writings [101]. We do not wish to assert that these problems are easy to solve. Rather, we show that decentralized smart contracting platforms do not solve them.

From the computer science perspective, we have demonstrated that the emerging world of DeFi is a rich area for collaboration between the fields of computer science and finance. We hold that many questions arising in the DeFi world might best be examined using computer science abstractions. In this way, the marriage of computer science and finance might beget us with both new opportunities and new challenges for years to come.

REFERENCES

[1] I. Fiedler and L. Ante, "Stablecoins," in The Emerald Handbook on Cryptoassets: Investment Opportunities and Challenges. Emerald Publishing Limited, 2023, pp. 93–105.

[2] M. Mita, K. Ito, S. Ohsawa, and H. Tanaka, "What is stablecoin?: A survey on price stabilization mechanisms for decentralized payment systems," in 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE, 2019, pp. 60–66.

[3] I. G. Pernice, S. Henningsen, R. Proskalovich, M. Florian, H. Elendner, and B. Scheuermann, "Monetary stabilization in cryptocurrencies–design approaches and open questions," in 2019 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, 2019, pp. 47–59.

[4] C. Catalini, A. de Gortari, and N. Shah, "Some simple economics of stablecoins," Annual Review of Financial Economics, vol. 14, pp. 117–135, 2022.

[5] O. President's Working Group On Financial Markets, FDIC, "Report on stablecoins," November 2021.

[6] P.-R. Agenor, J. S. Bhandari, and R. P. Flood, "Speculative attacks and models of balance-of-payments crises," National Bureau of Economic Research, Working Paper 3919, November 1991. [Online]. Available: http://www.nber.org/papers/w3919

[7] P. Krugman, "A model of balance-of-payments crises," Journal of Money, Credit and Banking, vol. 11, no. 3, pp. 311–325, 1979. [Online]. Available: http://www.jstor.org/stable/1991793

[8] R. P. Flood and P. M. Garber, "Collapsing exchange-rate regimes: Some linear examples," Journal of International Economics, vol. 17, no. 1, pp. 1–13, 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022199684900023

[9] HKMA, "2020 annual report," 2021.

[10] H. T. Heinonen, "On creation of a stablecoin based on the morini's scheme of inv&sav wallets and antimoney," in 2021 IEEE International Conference on Blockchain (Blockchain). IEEE, 2021, pp. 409–416.

[11] S. Kazemian, J. Huan, J. Shomroni, and K. Iyer, "Frax: A fractional-algorithmic stablecoin protocol," in 2022 IEEE International Conference on Blockchain (Blockchain). IEEE, 2022, pp. 406–411.

[12] S. Y. Jin, B. T. Xu, P. Intallura, and Y. Xia, "A utxo-based sharding method for stablecoin," in 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA). IEEE, 2022, pp. 195–199.

[13] V. Buterin, "Mastercoin: A second-generation protocol on the bitcoin blockchain," Bitcoin Magazine, November 2013. [Online]. Available: https://bitcoinmagazine.com/technical/mastercoin-a-second-generation-protocol-on-the-bitcoin-blockchain-1383603310

[14] J. R. Willet, "Mastercoin complete specification," 2012. [Online]. Available: https://cryptorating.eu/whitepapers/Omni/MasterCoinSpecification1.1.pdf

[15] L. Ante, I. Fiedler, J. M. Willruth, and F. Steinmetz, "A systematic literature review of empirical research on stablecoins," FinTech, vol. 2, no. 1, pp. 34–47, 2023.

[16] R. H. Sams, "A note on cryptocurrency stabilisation: Seigniorage shares," 2015.

[17] C. Catalini and A. de Gortari, "On the economic design of stablecoins," Available at SSRN 3899499, 2021.

[18] A. K. Goharshady, "Irrationality, extortion, or trusted third-parties: Why it is impossible to buy and sell physical goods securely on the blockchain," CoRR, vol. abs/2110.09857, 2021. [Online]. Available: https://arxiv.org/abs/2110.09857

[19] A. Klages-Mundt, D. Harz, L. Gudgeon, J.-Y. Liu, and A. Minca, "Stablecoins 2.0," Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, Oct 2020. [Online]. Available: http://dx.doi.org/10.1145/3419614.3423261

[20] R. Clements, "Built to fail: The inherent fragility of algorithmic stablecoins," 11 Wake Forest L. Rev. Online 131 (October 2021), 2021. [Online]. Available: http://www.wakeforestlawreview.com/2021/10/built-to-fail-the-inherent-fragility-of-algorithmic-stablecoins/

[21] Y. Li and S. Mayer, "Money creation in decentralized finance: A dynamic model of stablecoin and crypto shadow banking," SSRN, 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3757083

[22] M. Darlin and L. Tassiulas, "Debt-financed collateral and stability risks in the defi ecosystem," 2022. [Online]. Available: https://arxiv.org/abs/2204.11107

[23] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," The Journal of Finance, vol. 25, no. 2, pp. 383–417, 1970. [Online]. Available: http://www.jstor.org/stable/2325486

[24] P. Milgrom, Putting Auction Theory to Work, ser. Churchill Lectures in Economics. Cambridge University Press, 2004.

[25] P. A. Samuelson and W. D. Nordhaus, Economics. New York: McGraw Hill, 2001.

[26] N. Mankiw, Principles of Economics, 5th edition. South-Western Cengage Learning, 2011, the Introductory-Level Textbook. [Online]. Available: http://mankiw.swlearning.com/

[27] F. Hayek, The Denationalization of Money. Institute of Economic Affairs, 1976.

[28] A. J. Schwartz, Money in Historical Perspective. University of Chicago Press, 2009. [Online]. Available: https://doi.org/10.7208/9780226742298

[29] A. Moin, E. G. Sirer, and K. Sekniqi, "A classification framework for stablecoin designs," 2019. [Online]. Available: https://arxiv.org/abs/1910.10098

[30] S. Nakamoto, "A peer-to-peer electronic cash system," Decentralized Business Review, vol. 21260, 2008.

[31] M. Woodford, "Quantitative easing and financial stability," National Bureau of Economic Research, Working Paper 22285, May 2016. [Online]. Available: http://www.nber.org/papers/w22285

[32] FDIC, "Resolution of failed institutions," 2002. [Online]. Available: https://www.fdic.gov/deposit/deposits/international/guidance/unution.pdf

[33] SRB, "The single resolution mechanism: Introduction to resolution planning," 2016.

[34] W. Bagehot, Lombard Street: A Description of the Money Market. McMaster University Archive for the History of Economic Thought, 1873.

[35] P. Krugman and M. Obstfeld, International Economics: Theory and Policy, 4th Edition. Pearson Education, 1997.

[36] F. A. Hayek, The denationalization of money. University of Chicago Press, 1976.

[37] R. D. Wright, "Search costs and the existence of money," Econometrica, vol. 62, no. 5, pp. 1387–1408, 1994.

[38] C. A. E. Goodhart, The evolution of money. Cambridge University Press, 2000.

[39] A. M. Turing, "Computability and λ-Definability," The Journal of Symbolic Logic, vol. 2, no. 4, pp. 153–163, 1937.

[40] B. M. E. B. M. E. Moret, The theory of computation / Bernard M. Moret. Reading, Mass: Addison-Wesley, 1998.

[41] N. D. Jones, Computability and Complexity from a Programming Perspective. Dordrecht: Springer Netherlands, 2002, pp. 79–135. [Online]. Available: https://doi.org/10.1007/978-94-010-0413-8

[42] J. E. Savage, Models of computation. Addison-Wesley Reading, MA, 1998, vol. 136.

[43] A. B. Matos, "Direct proofs of rice's theorem," 2020. [Online]. Available: https://www.dcc.fc.up.pt/~acm/ricep.pdf

[44] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications. New York: Elsevier, 1976.
[45] B. Bollobás, Modern Graph Theory, 1st ed., ser. Graduate Texts in Mathematics 184. Springer-Verlag New York, 1998.
[46] L. Trevisan, "Cs 261: Optimization, lecture 16 , handout 16," March 2011. [Online]. Available: http://theory.stanford.edu/~trevisan/cs261/lecture16.pdf
[47] T. H. Lai, "Cis 6331 homework 9," 2017. [Online]. Available: http://web.cse.ohio-state.edu/~lai.1/6331/au17-hw9.pdf
[48] B. Hong and V. K. Prasanna, "Constrained flow optimization with applications to data gathering in sensor networks," in Algorithmic Aspects of Wireless Sensor Networks, S. E. Nikoletseas and J. D. P. Rolim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 187–200.
[49] S. Khuller and J. Naor, "Flow in planar graphs with vertex capacities," Algorithmica, 1994.
[50] Circle. [Online]. Available: https://www.circle.com/en/usdc
[51] Paxos, "Pax dollar (usdp)," 2021. [Online]. Available: https://insights.paxos.com/hubfs/USDP-whitepaper.pdf
[52] E. Kereiakes, D. Kwon, M. D. Maggio, and N. Platias, "Terra money: Stability and adoption," April 2019.
[53] The Maker Team, "The dai stablecoin system," 2017. [Online]. Available: https://makerdao.com/whitepaper/Dai-Whitepaper-Dec17-en.pdf
[54] Hubble Team, "Hubble protocol official docs," 2021. [Online]. Available: https://docs.hubbleprotocol.io/about-hubble/usdh
[55] K. Inami, "Uxd protocol," 2021. [Online]. Available: https://uxd.fi/static/media/whitepaper.7be6354b.pdf
[56] Y. Cao, M. Dai, S. Kou, L. Li, and C. Yang, "Designing stable coins," April 2021. [Online]. Available: https://rmi.nus.edu.sg/wp-content/uploads/2021/04/RMI-WPS-2021-05.pdf
[57] R. Wermers, "Runs on money market mutual funds," 2012.
[58] J. Zahnentferner, D. Kaidalov, J.-F. Etienne, and J. Díaz, "Djed: A formally verified crypto-backed pegged algorithmic stablecoin," August 2021. [Online]. Available: https://iohk.io/en/research/library/papers/djeda-formally-verified-crypto-backed-pegged-algorithmic-stablecoin/
[59] THORChain, "Thorchain docs," 2022. [Online]. Available: https://docs.thorchain.org/thorchain-finance/composite-model
[60] L. Menand, "Unappropriated dollars: The fed's ad hoc lending facilities and the rules that govern them," 2020.
[61] Federal Reserve, "Money market mutual fund liquidity facility," 2020.
[62] EU, "Regulation (eu) 2017/1131 of the european parliament and of the council of 14 june 2017 on money market funds," 2017.
[63] J. Donald R. Adams, Finance and Enterprise in Early America: A Study of Stephen Girard's Bank, 1812-1831. University of Pennsylvania Press, 2016. [Online]. Available: https://doi.org/10.9783/9781512800012
[64] C. P. Kindleberger, A financial history of Western Europe / Charles P. Kindleberger. Allen & schnee London ; Boston, 1984.
[65] J. R. Moen and E. W. Tallman, "Why didn't the United States establish a central bank until after the panic of 1907?" Federal Reserve Bank of Atlanta, FRB Atlanta Working Paper 99-16, 1999. [Online]. Available: https://ideas.repec.org/p/fip/fedawp/99-16.html
[66] J. Hull, Options, futures, and other derivatives, 6th ed. Upper Saddle River, NJ [u.a.]: Pearson Prentice Hall, 2006.
[67] P. P. Carr and R. A. Jarrow, "The stop-loss start-gain paradox and option valuation: A new decomposition into intrinsic and time value," The Review of Financial Studies, vol. 3, no. 3, pp. 469–492, 1990. [Online]. Available: http://www.jstor.org/stable/2962078
[68] M. Egorov, "Automatic market-making with dynamic peg," 2021. [Online]. Available: https://curve.fi/files/crypto-pools-paper.pdf
[69] Tether Team, "Tether: Fiat currencies on the bitcoin blockchain," 2014. [Online]. Available: https://tether.to/whitepaper
[70] S. Elliott, "Lido warns leveraged traders at risk of liquidation as 'staked ethereum' loses peg," May 2022. [Online]. Available: https://decrypt.co/100375/lido-warns-leveraged-traders-risk-liquidation-staked-ethereum-loses-peg
[71] B. Giove, "Zero degrees celsius [lite]," June 2022. [Online]. Available: https://newsletter.banklesshq.com/p/zero-degrees-celsius-lite
[72] D. Duong, "Liquid staking: Reconciling the steth price gap," June 2022. [Online]. Available: https://www.coinbase.com/institutional/research-insights/research/monthly-outlook/liquid-staking-june-2022
[73] Farmers Only Team, "Farmersonly docs," 2022. [Online]. Available: https://docs.farmersonly.fi/tba/vaults
[74] J. Kearns and P. Lowe, "Promoting liquidity: Why and how?" 2008. [Online]. Available: https://www.rba.gov.au/publications/rdp/2008/pdf/rdp2008-06.pdf
[75] Bank of Canada, "Bank of canada lender-of-last-resort policies," 2010. [Online]. Available: https://www.bankofcanada.ca/wp-content/uploads/2010/07/llr.pdf
[76] ECB, "Emergency liquidity assistance," November 2020.
[77] MAS, "Emergency liquidity assistance in singapore," June 2019. [Online]. Available: https://www.mas.gov.sg/-/media/Emergency-Liquidity-Assistance-Monograph.pdf
[78] L. Nyberg, "The infrastructure of emergency liquidity assistance - what is required in today's financial system?" 2000. [Online]. Available: https://www.bis.org/review/r000523c.pdf
[79] Thailand, "Bank of thailand act," 1945. [Online]. Available: https://www.dpa.or.th/en/file/download/bank-of-thailand-act-be2485
[80] J. Bullard, "President's message: The fed's emergency liquidity facilities: Why they were necessary," 2011. [Online]. Available: https://www.stlouisfed.org/publications/regional-economist/january-2011/the-feds-emergency-liquidity-facilities-why-they-were-necessary
[81] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 core," 2020. [Online]. Available: https://uniswap.org/whitepaper.pdf
[82] U. Bindseil, F. Panetta, and I. Terol, "Central Bank Digital Currency: functional scope, pricing and controls," European Central Bank, Occasional Paper Series 286, Dec. 2021. [Online]. Available: https://ideas.repec.org/p/ecb/ecbops/2021286.html
[83] FRB, "Money and payments: The u.s. dollar in the age of digital transformation," 2022. [Online]. Available: https://www.federalreserve.gov/publications/files/money-and-payments-20220120.pdf
[84] A. Buldas, M. Saarepera, J. Steiner, and D. Draheim, "A unifying theory of electronic money and payment systems," Jul 2021.
[85] R. Bansal and S. Singh, "Advantages of cbdcs in cross-border payments," in China's Digital Yuan: An Alternative to the Dollar-Dominated Financial System. Carnegie Endowment for International Peace, 2021, pp. 7–10. [Online]. Available: https://www.jstor.org/stable/pdf/resrep34852.8.pdf
[86] R. Ali, N. Bilotta, F. Botti, M. Cirasino, C. Lopez, J. Knoerich, T. Masela, F. Passacantando, and S. L. Schwarcz, The (Near) Future of Central Bank Digital Currencies. Risks and Opportunities for the Global Economy and Society, nicola Bilotta and F. Botti, Eds., 2021.
[87] P. Siklos, "Central bank digital currency and governance: fit for purpose?" 2021. [Online]. Available: https://apo.org.au/node/312055
[88] H. Chen and P. L. Siklos, "Central bank digital currency: A review and some macro-financial implications," Journal of Financial Stability, vol. 60, p. 100985, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1572308922000146
[89] B. Eichengreen and G. Viswanath-Natraj, "Stablecoins and central bank digital currencies: Policy and regulatory challenges," 2022. [Online]. Available: https://arxiv.org/abs/2202.07564
[90] G. Shafer and V. Vovk, Borel's Law of Large Numbers, 05 2019, pp. 5–30.
[91] F. Martinelli and N. Mushegian, "Balance whitepaper," 2019. [Online]. Available: https://balancer.fi/whitepaper.pdf
[92] A. Bukov and M. Melnik, "Mooniswap by 1inch.exchange," 2020. [Online]. Available: https://whitepaper.io/document/618/1inch-whitepaper
[93] Serum Foundation, "Serum - white paper," July 2020. [Online]. Available: http://projectserum.com
[94] G. Choi, "Tok stablecoin: A catalyst for defi ecosystem expansion," SSRN Electronic Journal, 2022.
[95] V. Buterin, May 2022. [Online]. Available: https://vitalik.ca/general/2022/05/25/stable.html
[96] D. S. H. Rosenthal, "Metastablecoins," May 2022. [Online]. Available: https://blog.dshr.org/2022/05/metastablecoins.html
[97] J. Sommer, "Hertz: And now for something completely worthless," June 2020. [Online]. Available: https://www.nytimes.com/2020/06/17/business/hertz-bankruptcy-stock-sale.html
[98] K. Dowd, "How hertz went from bankrupt to buying 100,000 teslas," November 2021. [Online]. Available: https://www.forbes.com/sites/kevindowd/2021/11/07/how-hertz-went-from-bankrupt-to-buying-100000-teslas
[99] L. Zhang, X. Ma, and Y. Liu, "Sok: Blockchain decentralization," 2022. [Online]. Available: https://arxiv.org/abs/2205.04256
[100] J. Morgan, "Systemic stablecoin and the defensive case for central bank digital currency: A critique of the bank of england's framing," Research in International Business and Finance, vol. 62, p. 101716, 2022.
[101] R. Sowerbutts, M. Schneebalg, and F. Hubert, "The demise of overend gurney," Bank of England Quarterly Bulletin, vol. 56, no. 2, pp.

94–106, 2016. [Online]. Available: https://EconPapers.repec.org/RePEc:boe:qbullt:0198