Continuous Credit Networks and Layer 2 Blockchains: Monotonicity and Sampling

ASHISH GOEL*, Stanford University
GEOFFREY RAMSEYER*, Stanford University

To improve transaction rates, many cryptocurrencies have implemented so-called "Layer-2" transaction protocols, where payments are routed across networks of private payment channels. However, for a given transaction, not every network state provides a feasible route to perform the payment; in this case, the transaction must be put on the public ledger. The payment channel network thus multiplies the transaction rate of the overall system; the less frequently it fails, the higher the multiplier.

We build on earlier work on credit networks and show that this network liquidity problem is connected to the combinatorics of graphical matroids. Earlier work could only analyze the (unnatural) scenario where transactions had discrete sizes. In this work, we give an analytical framework that removes this assumption. This enables meaningful liquidity analysis for real-world parameter regimes.

Superficially, it might seem like the continuous case would be harder to examine. However, removing this assumption lets us make progress in two important directions. First, we give a partial answer to the "monotonicity conjecture" that previous work left open. This conjecture asks that the network's performance not degrade as capacity on any edge increases. And second, we construct here a network state sampling procedure with much faster asymptotic performance than off-the-shelf Markov chains ($O(|E|\beta(|E|))$, where $\beta(x)$ is the complexity of solving a linear program on x constraints.)

We obtain our results by mapping the underlying graphs to convex bodies and then showing that the liquidity and sampling problems reduce to bounding and computing the volumes of these bodies. The transformation relies crucially on the combinatorial properties of the underlying graphic matroid, as do the proofs of monotonicity and fast sampling.

CCS Concepts: • **Applied computing** → **Electronic funds transfer**; *Digital cash*; • **Mathematics of computing** → Network flows.

Additional Key Words and Phrases: Credit Networks, Monotonicity

1 INTRODUCTION

1.1 Problem Motivation

Recent years have witnessed a dramatic rise in the importance of cryptocurrencies such as Bitcoin [33] and Ethereum [8] as means of exchanging money and storing value [12]. Unfortunately, however, most blockchains can directly process only a small number of transactions per second. Most Bitcoin

---

*Both authors contributed equally to this research.

variants, for example, only support 4-5 transactions per second [29]. One method for improving data throughput is settle transactions privately, using the blockchain as fallback security guarantor. This enables the construction of so-called "off-chain" or "Layer-2" networks. The most famous of these networks is the Lightning network on Bitcoin[34], but Layer 2 protocols have also been implemented in Ethereum and other cryptocurrencies [1–3].

The core idea of these networks is that if two parties frequently transact, they need not record every transaction on the global ledger, but rather, need only record the net total of their transactions at the end of a business relationship. Traditional commercial banks behave in a similar manner; instead of repeatedly transferring cash, they often have accounts with each other, and simply move money in and out of these accounts.

Traditional banks can go to court if a financial transfer never arrives. In an anonymous cryptocurrency network, or when blockchains span political jurisdictions, users may have no such recourse. Instead, to guarantee solvency, two parties can agree to put money into escrow. Transferring money then means privately updating shares of ownership of the escrowed funds. Such an escrow account is typically called a "payment channel." Pairs of agents who lack a direct channel can route funds across a path in the payment channel graph. Figure 1 gives an example payment network on 5 nodes.
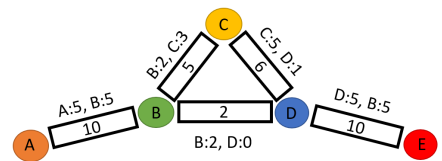
However, if one agent comes to own the entirety of the escrowed funds, it cannot accept additional money on that channel without opening itself to risk of being cheated by its channel counterparty. More generally, no fixed network of payment channels can satisfy every possible sequence of transactions. We give a more detailed description of the network model, based on credit networks [13], in Section 2.

When a payment channel network cannot process a transaction, that transaction falls back to the underlying blockchain. The ratio of the number of transactions successfully settled privately to the number of transactions that a payment channel network cannot settle thus acts as a multiplier on a blockchain's limited transaction rate. This ratio is our object of study; a higher ratio is clearly

Fig. 1. An example payment network on 5 nodes. In this network, A and B have together invested 10 units of capital towards the edge (A, B). From this configuration, A can send up to 5 units of money to B. B could send up to 4 units of money to D through the graph, 2 units directly and 2 via C.

desirable for improving blockchain performance. Higher liquidity can also benefit individual agents directly, in that a higher transaction throughput capacity should mean lower fees per individual transaction, and higher liquidity means higher chance of being able to purchase something online quickly.

Implementing a payment channel network in the real world is a complex systems design problem. For example, agents must have some way of discovering payment routes across the network. Some users might also like to keep their payment patterns private from network intermediaries. However, as we show here, even if a payment network operates perfectly – no hardware ever fails, every node has full information, transactions resolve instantaneously – no payment network can resolve every transaction (under a natural class of transaction models). Therefore, the baseline performance of e.g. a payment routing system is not 100% success, but rather, the transaction success rate in an ideal system. Our goal in this work is to study this baseline performance.

In real-world implementations of payment channel networks, off-chain transactions can settle in seconds, but putting data onto the blockchain or settling on-chain can take 10s of minutes. Because the network's topology itself cannot change fast enough to respond to individual transactions, we study the performance of fixed network topologies.

Of course, placing capital into escrow to secure a link incurs a cost, in terms of the interest rate on the escrowed funds. When agents participate in a payment network, they implicitly choose to pay this cost in exchange for the ability to use the network directly. One goal of this work is to provide a framework for agents to understand the utility they would derive from participating in the network in a particular manner, and thus to evaluate whether this utility outweighs the costs they must pay. Agents must make a strategic choice about how to allocate their escrowed funds. One agent could choose to make one high-capacity link with a central node, or it could choose to make several lower-capacity links with different nodes. An agent might wish to avoid making a link with another agent whose internet connection is unreliable. Given some set of feasible options, agents must implicitly pick a preferred option. One goal of this work is to lay out a framework for agents to evaluate and compare their options.

1.2   Monotonicity

At a high level, a payment network appears similar to a single-commodity flow network when resolving a single transaction. The main difference is that transactions have side effects, and side effects of different transactions can cancel each other out or compound each other. Many network flow problems or queueing problems study the case where the flow across an edge in each direction is independently limited. In a payment network, flow in one direction cancels out flow in the other direction. At any point in time, on each link, the total flow sent since the start of network operation in one direction must be approximately equal to the total flow in the other direction (where approximately means $\pm$ the amount of funds in escrow on that link).

Many seemingly similar network flow problems are subject to what has become known as Braess's paradox [6]. Famously, in car traffic networks, when drivers choose their routes selfishly, total traffic throughput in a road network can be decreased by adding roads. Example 4.6 of [35] gives a simple example of a credit network variant that violates monotonicity. Many network models display similar deleterious effects.

In cryptocurrency networks, where no central party exists to act as a coordinator, if selfishly routing payments had this deleterious effect, there would be no obvious way to coordinate a more efficient routing process. It could also be possible for an adversarial party to add edges to the network to decrease throughput (there might even be a financial incentive to do this – offchain transactions reduce demand for data on the blockchain, which lowers on-chain transaction fees). There naturally arises, then, a question about the "monotonicity" of payment networks – is the fraction of transactions that succeed monotonically non-decreasing as the number and capacity of edges increase?

A Layer-2 payment network closely corresponds to a distributed currency model known as a credit network, first formulated independently by [14, 18, 26]. Dandekar et. al. and Goel et. al. [13, 19] developed the mathematical foundations for the model and analyze transaction success rates, which they call "liquidity". In order to perform their analysis, these authors assume that transactions have discrete sizes. With this assumption, the monotonicity conjecture is equivalent to a conjecture that matroid theory's "negative correlation" conjecture holds on the collection of independent sets of a graphical matroid. Feder and Mihail in [17] showed that if a matroid has this property, one can sample in polynomial time an approximately random basis, but not all matroids have this property. More recently, Anari et. al. [5] gave a polynomial-time approximate sampling algorithm for general matroids.

1.3   Our Results

In this work, we give a general analytical framework for Layer-2 payment networks with arbitrarily sized transactions, eliminating a key assumption needed for [13, 19], and use this toolkit to prove the

monotonicity conjecture for the classes of transactions most relevant to cryptocurrency applications. We also give an efficient network state sampling algorithm.

The detailed results are more understandable with the following two pieces of context.

First, formally defining the "probability that a transaction is successful" requires a small amount of care. We will define it as the probability that a given transaction is feasible in configuration sampled according to some probability measure μ on the set of configurations. This, of course, raises the question of what measures μ arise from real-world behavior.

We model transactions as arising from an exogenous random process. E-commerce transactions, for example, arise largely from demand for consumer goods, not from anything related to the short-term status of a payment processing pipeline. For consistency with prior work [13, 19], we also model transactions as arising in sequence in discrete time. At every timestep, a payment network simply executes the current transaction if it is able in its current escrow-configuration (where a network's escrow-configuration is the ownership status of every edge; henceforth "configuration"). Every successful transaction changes the configuration of the network, so a random transaction model induces a Markov chain on the set of configurations of the network. In the rest of this work, we measure transaction success probability relative to the invariant measure of this Markov chain. There are other similar transaction models that would define natural, induced probability measures; results derived through our analytical framework depend on the probability measure, not the process through which a measure is induced.

Second, the state space of the induced Markov chain is most succinctly described not as a set of escrow-configurations but rather, as a set of classes of functionally-equivalent configurations. An agent can send money to itself along a cycle to change the network configuration – however, every transaction realizable in the network configuration prior to the cyclic flow is realizable in the network configuration after the cyclic flow. Sending flow along cycles gives a "cycle-equivalence" relation between configurations, and for the purpose of executing transactions, representatives of one equivalence class are all functionally equivalent. The induced Markov chain is thus most naturally viewed as acting on the equivalence classes of this equivalence relation.

Prior work primarily analyzed the case where transaction distributions are symmetric, that is to say, agent u is as likely to pay agent v as v is likely to pay u. Such transaction distributions induce uniform probability measures. This assumption may not be realistic in some real-world credit network use cases. For clarity, note that of the below results, result 1 (sections 2-4) does not rely on this assumption, while result 2 (section 5) does, as does Theorem 6.2 of result 3. We give a small comment on this assumption and related open problems in section 7.

Our results are the following:

(1) In sections 2, 3, and 4, we construct a new toolset for analyzing the set of configurations of a credit network. Simply put, we construct a map D (for "Direction" of a transaction) from the set of possible transactions to $\mathbb{R}^n$ and a map S from the set of configurations of a credit network to $\mathbb{R}^n$ that together preserve how transactions and configurations interact. If a transaction τ executed from configuration w_1 sends the network into configuration w_2, then $S(w_1) + D(\tau) = S(w_2)$. The stationary measure of the induced Markov chain is a probability measure on $\mathbb{R}^n$, and analyzing liquidity reduces to analyzing the measures of subsets of $\mathbb{R}^n$. For comparison to prior work, we study a continuous extension of the credit network model of [13, 19]. Specifically, Goel et. al. and Dandekar et. al. [13, 19] measure performance with regard to transactions of size 1, where 1 is some minimum transaction size, while we study transactions of arbitrary real-valued size.

If transactions all have the same size (or integral sizes), then configurations of the credit network can be mapped into forests (acyclic subsets of edges) of related, undirected multigraph

with the same vertex set. They show that the probability that a transaction of size 1 succeeds from agent u to agent v is equal to the probability that a forest sampled uniformly at random puts u and v in the same connected component.

Unfortunately, this correspondence gives no clues as to how to analyze liquidity for transactions of larger size. In real-world blockchains like Bitcoin, one Satoshi – one "smallest unit of money" – is worth at the time of writing roughly 1/10000th of a US dollar, while Lightning channel capacities may run into the hundreds or thousands of dollars. In this kind of parameter regime, a transaction of size 1 succeeds with probability very close to 1 in any network topology, and most transactions sent have much larger sizes. An agent in the network would be interested primarily in transactions of larger, but variable, sizes.

We also study the rate at which transaction success rates decline as transaction sizes increase. Modeling transactions as taking on arbitrary real-valued sizes streamlines this analysis and requires a new set of analytical tools.

As an aside, these analytical frameworks are in fact related mathematically. From the point of view of prior work, we study the success rate of transactions of fixed real-world value, as discretization becomes arbitrarily fine. Informally, analyzing the discrete case compared to the continuous case is like studying the number of points of a lattice within a well-behaved convex body, instead of directly studying the volume of said convex set. Prior work of [13, 19] is able to measure liquidity using forests because of a correspondence between forests and these lattice points ([39], Example 3.1). As a lattice becomes increasingly fine, a (weighted) count of the number of lattice points in a well-behaved convex set approximates the volume of said set; we note without proof that many of our arguments could be adapted to hold if the transaction granularity is sufficiently small using this principle.

(2) In section 5, we prove theorems about the liquidity of general credit networks and a limited version of the monotonicity conjecture.

Relative to a uniform probability measure (i.e. transaction distributions that are symmetric), transaction success probability can be computed via effective resistance calculations.

Payment networks on cryptocurrencies are particularly beneficial for two kinds of transactions. First, they streamline transactions between parties that transact frequently, especially between parties with a direct connection. And second, by increasing transaction throughput, they put downward pressure on transaction fees, which helps make small transactions economically worthwhile. Again relative to a uniform probability measure (i.e. transaction distributions that are symmetric), our analytical toolkit enables a proof of the monotonicity conjecture for transactions of these types.

In real world terms, this means that bad actors cannot add edges to the graph and harm either of these payment network use cases.

(3) Finally, in section 6, we also show that the problem of sampling a random network state reduces to the problem of sampling a random point from a zonotope (that depends on the payment network graph). For many natural distributions, this sampling can be done in practice with reasonably fast Markov chains, such as the well-studied Hit and Run process [38].

Furthermore, relative to a uniform measure on network configurations, we give an algorithm (Theorem 6.2) for exact uniformly random sampling states of a payment network of n agents and m edges in time $O(m\beta(m))$, where $\beta(m)$ is the complexity of solving a linear program on $O(m)$ constraints. For comparison, Hit and Run would require $O(n^3\beta(m))$. Currently, $\beta(m) = m^\omega$ [11].

Informally, if we had instead modeled transactions as occuring rapidly and simultaneously, a natural analogue of the above induced Markov chain is a Reflected Brownian motion on the

same state space. In this model, if the transaction distribution has a bias towards certain vertices (say, one vertex is more likely to be paid than to pay others), then the stationary distribution of the Brownian motion is exponential in the direction of the bias. These distributions can be sampled using Hit and Run.

1.4 Related Work

The concept of the Credit Network that underlies the analysis here was developed independently by [14, 18, 26]. These authors worked in the contexts of informal borrowing networks [26], online reputation systems backed by social relationships [14], and auction systems built on a credit network.

Recently, an analysis of the distribution of network states induced by a distribution on transactions was conducted in [7]. The stated goal of these authors is to understand transaction pricing models, but central to the pricing analysis is an understanding of transaction failure rates in the short term. By constrast, for scaling a network, the key metric is failure rate amortized in the long term. Their analysis, however, only looks at graphs consisting of isolated edges and of stars, and also assumes that the size of a transaction between pairs is fixed.

Most early implementations of payment networks discuss transactions as being routed along single paths (as in e.g. [34]). But for the purpose of actually sending money, one successful \$2 transaction along a single path is functionally equivalent to two separate successful \$1 transactions along two different paths. More generally, a transaction in a payment network can be thought of as a flow through the network.

What we study is also similar to the study of "capacitated transportation networks," as in [15] and [24]. In this model, the capacity of each edge is drawn independently from some random distribution. These capacities are "successful" if they can satisfy some fixed set of flow sources/demands, and the reliability of the network is the chance of success. Similarly, Grimmett and Suen [21] studied the expected size of a maximum flow through a graph, if edge capacities are random. Efforts to approximately compute transportation network reliability include [30].

A key difference from our model, however, stems from the fact that routing flow along a cycle does not change the ability of a payment network to satisfy a flow demand. The set of network states, therefore, is not the product of sets of states on individual edges (unless the network is acyclic).

2 THE NETWORK MODEL

In practice, a payment network consists of a set of agents and a set of edges between agents, where each edge has some associated amount of money in escrow. If an edge runs between two agents A and B, then A and B privately track how much of the escrow on the edge belongs to A and how much belongs to B. For example, Figure 1 gives a simple payment network between a few nodes.

Suppose that agent A wishes to pay one unit of money to agent B. Then the amount of money on the edge between A and B remains constant, but A and B privately decide that of this escrow, A's ownership stake decreases by one, and B's ownership increases by one. Of course, this transaction would fail if and only if A's ownership stake were less than one.

Formally, then, we can model a payment network as the following. As transactions occur, the unchanging components of the system consist of a graph $G = (V, E)$, representing nodes and edges, along with a map $c : E \to \mathbb{R}_{\geq 0}$ that records the capacity of each edge. An escrow configuration of the network, which changes as transactions happen, will be a map $w : (V \times V) \to \mathbb{R}_{\geq 0}$ recording, for each edge, how much of the escrow each agent owns. Specifically, for some edge (u, v), if $w(u, v) = k$, then u owns k units of escrow on edge (u, v), and $w(u, v) + w(v, u) = c(u, v) = c(v, u)$.

Before proceeding, we would like to note that an escrow configuration w in the above sense is *not* the same as the network states discussed in the introduction. For the purpose of sending money,

many escrow configurations can be functionally equivalent, and we will treat equivalent escrow configurations as a single network state.

A simple transaction along a single edge will consist of a sender u, a receiver v, and an amount k, and we will say that the transaction succeeds if and only if $w(u, v) \geq k$. If this transaction succeeds, the network will be left in the state where $w(u, v)$ is decreased by k and $w(v, u)$ is increased by k.

To complete the model, a transaction τ consists of a sender x, a receiver y, and a transaction amount k, and we will say that τ succeeds in a configuration w if and only if there exists a commodity flow $f = \{f(u, v) : (u, v) \in E\}$ taking k units of flow from x to y with $w(u, v) \geq f_{(u,v)}$. Such a transaction change an escrow configuration in the same way that simple transactions of size $f_{(u,v)}$ along each edge (u, v) collectively change an escrow configuration.

2.1 Definitions and Basic Properties

First, note that if agents can successfully perform the same sequences of transactions starting from two different escrow configurations, then in terms of moving money, the configurations are identical. This motivates the following definition.

Definition 2.1 (Transaction Equivalence). Two configurations w_1 and w_2 are *transaction equivalent* if, given any sequence of transactions τ, τ succeeds starting from w_1 if and only if τ succeeds starting from w_2.

Unfortunately, this definition provides no obvious aid in understanding the space of configurations. For that, consider as an example a cycle graph where each edge has capacity 1, and pick one direction around the cycle to be "clockwise". Let w_1 be the configuration where $w_1(u, v) = 1$ if and only if (u, v) points in the clockwise direction, and let w_2 be the configuration where $w_2(u, v) = 1$ if and only if (u, v) points in the counterclockwise direction. Then for any two vertices x and y, x could send at most one unit of payment to y along a counterclockwise path in configuration w_1 and could send at most one unit of payment along a clockwise path in configuration w_2. In this sense, then, w_1 and w_2 are transaction-equivalent.

Observe that one could move from w_1 to w_2 by routing a payment from a vertex to itself in a counterclockwise direction. This motivates the following definition.

Definition 2.2 (Cycle Equivalence (Definition 2, [13])). Two configurations are *cycle-equivalent* if and only if one is reachable from the other by routing a series of payments along cycles.

We will see later that these two definitions are equivalent (Corollary 3.8), and consequently, that the choice of route when making a transaction does not influence the resulting cycle-equivalence class.

Finally, the "liquidity" of the network is the probability that a transaction will succeed from a "random" cycle-equivalence class. We make this definition formal in 4, but intuitively, the distribution of interest is the one that is induced by real-world random processes. We model this as the stationary distribution of a Markov chain that sequentially attempting to execute randomly generated transactions; this Markov chain has a dimple description if it's state space is thought of as cycle-equivalence classes, not as individual configurations.

3 FROM NETWORKS TO CONTINUOUS REPRESENTATIONS

For the rest of the discussion, we will assume that a graph $G = (V, E)$ is connected, and let $n = |V| - 1$ be the size of a spanning tree in G.

At a high level, in this section, we will construct a map S from the set of escrow configurations to a convex subset Z of $\mathbb{R}^n$, and show that sending money in the credit network is equivalent to moving in Z. That is to say, every transaction τ corresponds to some vector $v_\tau \in \mathbb{R}^n$, and if executing τ from

a configuration w results in the configuration w', then $S(w) - v_\tau = S(w')$. Furthermore, the preimage of any point in Z is exactly one cycle-equivalence class of configurations.

This map relies on a representation of the graphic matroid in $\mathbb{R}^n$, subject to the additional constraint that the signed summation of the edges visited while walking in a cycle is 0. This condition lets us discuss the "direction" of money moving from a vertex x to a vertex y in a well-defined manner.

Fix some orientation $s : (V \times V) \to \pm 1$ on every edge.

Definition 3.1. Let D (for "direction") be a map from E to $\mathbb{R}^n$.
The *direction* of a path $p = (v_0, ..., v_k)$ is simply the (signed) summation of D along the path:

$$\sum_{i=0}^{k-1} s(v_i, v_{i+1}) D(v_i, v_{i+1})$$

The direction of a flow f is similarly defined to be $\sum_{(u,v) \in V \times V} f_{(u,v)} s(u,v) D(u,v)$
We will refer to the direction of a path p as $D(p)$ and the direction of a flow f as $D(f)$.

Sending money in a credit network is akin to sending flow through the underlying residual graph. We want the direction of the flow to capture something about how a credit network configuration changes as money is sent. For that, it suffices for a direction map to satisfy the following conditions:

Definition 3.2. A direction map D is a *spanning representation* of the credit network if:
(1) The direction (under D) of any path is 0 if and only if p is a cycle.
(2) For any spanning tree T, the set of vectors $\{D(t)\}_{t \in T}$ spans $\mathbb{R}^n$.

THEOREM 3.3. *Spanning representations exist.*

The proof is in the appendix, but informally, take the edges of a spanning tree, represent them as the standard basis, and represent the other edges so that cycles sum to 0. We will not mention the representation when the choice of representation is clear from context. The utility of a spanning representation is that it naturally induces a map from configurations to $\mathbb{R}^n$ that has several useful properties.

Definition 3.4. The *represented score* of a configuration w with respect to a spanning representation D is $S_D(w) = \sum_{(u,v) \in V \times V : s(u,v)=1} w(u,v) D(u,v)$.[1]

The condition that $s(u,v) = 1$ means that the summation above is over edges pointed in their positive direction.

This should be thought of as a generalization of the "score vectors" as in e.g. [28]. In fact, the represented score map exactly captures the transactional relationships between cycle-equivalence classes.

THEOREM 3.5.
  1 *Two cycle-equivalent configurations have the same represented score.*
  2 *If a transaction along path p of size k takes configuration w_1 to configuration w_2, then $S_D(w_1) - k * D(p) = S_D(w_2)$.*
  3 *If two paths p and q both go from vertex x to vertex y, then $D(p) = D(q)$.*

A transaction of size k from x to y is a special case of a flow demand, and as a corollary of the above properties, any flow that satisfies a given demand will have the same direction. Given some spanning representation, then, it is reasonable to think of transactions not just as paths from x to y but also as flow demands or as vectors in $\mathbb{R}^n$. In particular, we make the following convention:

---

[1]The analysis would work out the same if we simply summed over all directed edges. This definition makes later theorem statements cleaner.

Definition 3.6. A transaction τ sending k units of flow from x to y corresponds to a vector $v \in \mathbb{R}^n$ if, for some path from x to y, $v = kD(p)$.

In other words, if executing τ sends configuration w_1 to w_2, then $S_D(w_1) - v = S_D(w_2)$.

These definitions are sufficient to analyze the space of cycle-equivalence classes.

First, recall that the *Minkowski Sum* of vectors $v_1, ..., v_k$ is the set $\{x : x = \sum_i \lambda_i v_i, \lambda_i \in [0,1]\}$.

THEOREM 3.7. *The image of the set of cycle-equivalence classes, under a spanning representation, is exactly the Minkowski sum of the vectors* $\{v_{(x,y)} = D(x,y) * c(x,y)) : (x,y) \in V \times V, s(x,y) = 1\}$.

As before, the $s(x,y) = 1$ requirement simply means that the summation is over all edges, viewed in their positive direction.

With these definitions, we can finally make the following useful characterization of transaction-equivalence.

COROLLARY 3.8. *Transaction-equivalence is equivalent to cycle-equivalence.*

4 INDUCED MEASURES ON THE CONFIGURATION SPACE

As outlined in the introduction, we would like to understand success probability with respect to a "random" configuration drawn from the distribution induced by real-world transaction activity. For this work, we will think of transactions as arriving from some external process; most transactional contexts like online shopping are driven by forces entirely external to the payment network, such as demand for consumer goods.

Our analysis is one of an idealized payment network, so we will model transactions as happening one at a time, communication links never fail, and transaction fees are zero.[2] In particular, we will model transactions as occuring in sequence, drawn from some distribution. At every timestep, we draw a new transaction by drawing a pair of vertices who will transact and then a transaction size. If the transaction is feasible in the current network configuration, we execute the transaction and move to the resulting state. Otherwise, we disregard the transaction.

Definition 4.1 (Transaction Model). A transaction model for a set of vertices V consists of the following:

(1) The set of pairs of vertices that transact, $X \subset V \times V$.
(2) A distribution ϕ on X.
(3) For each pair $(a,b) \in X$, a continuous probability distribution $k_{a,b}(\cdot)$ on the size of a transaction between a and b.

This transaction model generates a transaction by using ϕ to choose a sender a and a receiver b, and then using $k_{a,b}(\cdot)$ to choose a transaction size.

For technical convenience, we will allow $k_{a,b}(\cdot)$ to output negative values, and sending negative money from a to b will be considered sending money from b to a. It is without loss of generality, then, to assume that only one of (a,b) or (b,a) is in X.

We would like to use a transaction model to induce a distribution on the set of cycle-equivalence classes. In the discrete credit network case of [13, 19], the set of cycle-equivalence classes was finite, and so those authors could simply sum over all classes. However, in our continuous setting, the set of cycle-equivalence classes has infinite cardinality. We would like to analyze the probability mass of a subset of configurations, but it is not clear how to integrate over a set of cycle-equivalence classes without building a lot of measure-theoretic infrastructure. Instead, given a fixed spanning representation, it will be much easier to simply map the set of cycle-equivalence classes into $\mathbb{R}^n$

---

[2]The question of how to optimally set transaction fees is a fascinating open problem for future study.

and analyze subsets of $\mathbb{R}^n$. As a notational shorthand, we will say that the measure of a set of cycle-equivalence classes is simply the measure of the image of the set in $\mathbb{R}^n$ (if said image is measurable).

Given this technical difficulty, when we define the following Markov Chain, we will use as its state space the image in $\mathbb{R}^n$ of the set of all cycle-equivalence classes under a fixed spanning representation.

Definition 4.2 (Markov Chain 1). Let $G = (V, E, c)$ be a payment network, let D be a spanning representation, let $Z \subset \mathbb{R}^n$ be the image under D of the set of cycle-equivalence classes of G, and let Ξ be some transaction model. The induced Markov chain will be defined as follows:

Start at some point $z_0 \in Z$.

At each timestep, draw a transaction between vertices a and b of size k from Ξ. If the transaction is feasible from the current state z_t, then perform the transaction and update the state of the network accordingly ($z_{t+1} = z_t - kD(a,b)$). Otherwise, remain in the same state ($z_{t+1} = z_t$).

The induced distribution on Z is the stationary measure of this Markov chain (if a stationary measure exists and is unique).

THEOREM 4.3. *The above Markov Chain has an invariant probability measure.*

We are primarily interested in the case where a transaction model induces a unique stationary measure. The following condition is sufficient but not necessary; there are many similar sufficient conditions.

THEOREM 4.4. *Let $G = (V, E)$ be a graph, and let $\Xi = (X, \phi, \{k_{a,b}\}_{(a,b) \in X})$ be a transaction model on V.*

Suppose that for each pair $(a,b) \in X$, the support of $k_{a,b}$ contains the interval $[-\varepsilon, \varepsilon]$ for some $\varepsilon > 0$. Then Markov Chain 1 has a unique stationary distribution if the graph (V, X) is connected.

Motivated by Theorem 4.4, we call a transaction model that satisfies these conditions a *connected* transaction model.

In the next section, we will primarily study the case where the unique stationary measure is uniform.

THEOREM 4.5. *Let $\Xi = (X, \phi, \{k_{a,b}\}_{(a,b) \in X})$ be a connected transaction model.*
If each $k_{a,b}$ is symmetric about 0, then the induced stationary measure is uniform.

Motivated by Theorem 4.5, we call a transaction that satisfies these conditions a *symmetric, connected* transaction model.

When the stationary measure is unique, the natural definition of transaction success probability is well-defined.

Definition 4.6. Given a spanning representation D, let Z be the set of configurations represented in $\mathbb{R}^n$ and let π be some density measure over Z. Then the *liquidity* of the network with regard to a transaction v is the relative measure of the image of the set of configurations where v succeeds, $\frac{\int_Z \pi(z) \mathbb{1}\{z - v \in Z\} dz}{\int_Z \pi(z) dz}$.

THEOREM 4.7. *The liquidity of a network with respect to a particular transaction does not depend on the choice of spanning representation.*

These theorems show that under small assumptions on transaction distributions, the distribution on configurations of a payment network is a well-defined topic of discussion.

5 LIQUIDITY ANALYSIS AND MONOTONICITY

One question that an agent operating in a payment network might ask is how adding escrow to one edge will affect transaction success rate across another edge. It would be bad for a payment network if bad actors could add escrow and in doing so decrease liquidity between well-behaved agents.

The authors of [19] were concerned with this question, which they dubbed the "Monotonicity Conjecture." In their work, they observe that this conjecture is equivalent to the well-studied "Negative Correlation" conjecture about the set of forests of a graph. To summarize, a matroid on a set of elements X is said to be negatively correlated if, for any two elements $x, y \in X$, the probability that x is in a random basis is greater than the probability that x is in a random basis that also contains y. [17] showed that this conjecture implies a polynomial time random sampling algorithm for matroid bases.

Not all matroids are negatively correlated, however. And although the set of forests is not a matroid, much intellectual energy has been devoted to the negative correlation of forests question – for example, [10, 20, 36].

We show in this section that for the monotonicity conjecture on credit networks, relaxing the (somewhat unnatural) restriction that transactions must have discrete sizes can actually make the monotonicity question tractable. Below we reproduce this conjecture in the context of our network model.

CONJECTURE 5.1 (MONOTONICITY). *Let $G = (V, E)$ be a credit network with edge capacities $c(\cdot)$, let τ be a transaction from a vertex x to another vertex y of size k, and let π be the uniform stationary measure induced by a symmetric, connected transaction model.*

For any pair of vertices $(w, z) \in V \times V$, increasing $c(w, z)$ in G should not decrease the liquidity of τ with respect to π.

5.1 Liquidity Analysis

To move towards a study of the monotonicity conjecture, we first prove some theorems about general liquidity in credit networks in the continuous model.

As before, to avoid the need to set up measure-theoretic infrastructure on the set of cycle-equivalence classes, we will for the rest of this section measure the probability of a set of configurations as the relative measure of its image in $\mathbb{R}^n$ under the map induced by a spanning representation. When the measure π is uniform, the measure of a set in $\mathbb{R}^n$ is simply its volume.

Therefore, as a notational shorthand, we take the "volume" of a set of cycle-equivalence classes to mean the volume in $\mathbb{R}^n$ of the image of the set of cycle-equivalence classes under the map induced by a fixed spanning representation. As per Theorem 4.7, liquidity metrics are not affected by choice of spanning representation.

As mentioned in Theorem 3.7, the image of the set of configurations is the Minkowski sum of a particular set of vectors. It turns out that the volume of this Minkowski sum is closely related to the undirected graphical matroid of the payment network. The connection comes through the generating polynomial for the matroid, which we now define for completeness.

Definition 5.2. Let $G(V, E)$ be a graph, and let $T(G)$ be the set of all spanning trees of G. Associate with every $e \in E$ a real variable y_e.

The *generating polynomial* of $T(G)$ is $\Gamma(y) = \Sigma_{T \in T(G)} \Pi_{e \in T} y_e$.

We will also denote as $\Gamma_e(y)$ the summation over sets S such that $S \cup \{e\}$ is a spanning tree, or equivalently, $\frac{\partial}{\partial y_e} \Gamma(y)$.

More generally, we will denote with $\Gamma_{a=b}(y)$ the generating polynomial of the graph where vertex a is identified with vertex b. If $e = (a, b)$, then $\Gamma_{a=b} = \Gamma_e$.

THEOREM 5.3. *Let $G = (V, E)$ be a payment network with edge capacities $c(\cdot)$.*

Suppose that a spanning representation $\hat{D}$ maps some tree to the standard basis. Then the volume of the image of the set of cycle-equivalence classes of configurations is equal to $\Gamma(c(e_1),...,c(e_m))$.

The volume of a Minkowski sum of vectors $v_1,...,v_m$ is equal to a summation over subset of vectors $v_{i_1},...,v_{i_n}$ of the determinant of the matrix whose columns are $v_{i_1},...,v_{i_n}$. The result follows from the fact that an edge can only appear once in a simple cycle and the multilinearity of the determinant.

For convenience, in the rest of this section, assume that we have fixed $\hat{D}$ to be such a spanning representation. [3]

To analyze liquidity, we also need a useful analytical description of the volume of set where a transaction v succeeds.

LEMMA 5.4. *Let v be the vector in $\mathbb{R}^n$ corresponding to some transaction τ If Z is the subset of $\mathbb{R}^n$ corresponding to the entire set of cycle-equivalence classes, then $Z \cap \{z : z - v \in Z\}$ (Minkowski subtraction) is the space where τ succeeds.*

PROOF. Follows from the theorems of Section 3. □

We start with the easy case for analysis. Theorems 3.7 and 5.3 shows that the image Z of the set of all cycle-equivalence classes is a zonotope with a convenient structure. When there is an edge already present in the graph between two agents x and y, the area where a transaction from x to y succeeds retains the same kind of structure.

In the language of the above lemma, for a transaction v from x to y, the space corresponding to transactions where v succeeds, $Z \cap \{z : z - v \in Z\}$, is a zonotope. In fact, this zonotope is identical to the zonotope generated by a credit network where the capacity of the edge from x to y is reduced by the size of the transaction.

THEOREM 5.5. *If v corresponds to a transaction between x and y of size k, and $c(x,y) \geq k$, then the volume where v succeeds is equal to $\Gamma(c'(e_1)...,c'(e_m))$ where $c' = c$ except that $c'(x,y) = c(x,y) - k$. The volume of the space where v fails is equal to $k\Gamma_{x=y}(c(e_1)...,c(e_m))$.*

When this holds, then computing a transaction's chance of success is straightforward.

COROLLARY 5.6. *Suppose that v corresponds to a transaction between x and y of size k, and that the graph contains an edge (x,y) with $c(x,y) \geq k$.*

1 *The probability that v fails is simply $k\Gamma_{(x,y)}(c(e_1),...,c(e_m))/\Gamma(c(e_1),...,c(e_m))$.*
2 *Let $G = (V,E)$ be the underlying graph, and let $G' = (V,E')$ be a multigraph where E' is E with (x,y) duplicated to make edges f_1 and f_2. Set $c'(f_1) = k$, $c'(f_2) = c(x,y) - k$, and for all other edges e, $c'(e) = c(e)$.*
   Then the probability that v fails is equal to the probability that f_1 is in a random weighted tree sample from G'.

As an aside, the criterion that x and y have an edge of capacity at least k does not mean that x and y must route all of their transactions explicitly along that edge. The model does not require any particular type of routing algorithm. The rest of the graph is still relevant for x and y when they share an edge, as many configurations will require x and y to use edges other than the one directly between them.

Now for the harder case for analysis. Namely, when $n > 2$ and v does not correspond to an edge in the underlying graph, the space where v succeeds is not necessarily a Minkowski sum of a set of

---

[3]As per Theorem 4.7, choice of spanning representation does not affect liquidity results. This makes the presentation of results simpler.

vectors and thus lacks a concise mathematical characterization. Such examples unfortunately occur in very simple graphs, such as the cycle on four vertices when nonadjacent vertices transact. Figure 6 of [4] gives a visual depiction of such an object.

Nevertheless, lower bounds on liquidity are derivable using the same analytical frameworks as Corollary 5.6.

LEMMA 5.7. *Let x and y be any two vertices, and let v be the direction of a path from x to y.*
Let $f_v(k)$ denote the probability that a transaction size k from x to y fails. Then $0 \leq f_v'(k) \leq \Gamma_{x=y}(c)/\Gamma(c)$. Moreover, $f_v'(0) = \Gamma_{x=y}(c)/\Gamma(c)$.

COROLLARY 5.8. *A lower bound of transaction success probability (i.e. $1 - k * f_v'(0)$) is computable efficiently (via calculating effective resistance, similarly to Theorem 5.6).*

PROOF. Let Z be the image of the set of all cycle-equivalence classes, let P be the projection of Z onto some fixed hyperplane with normal v, and let ρ denote the volume of the projection in $\mathbb{R}^{n-1}$. For every $p \in P$, consider the line extending in direction v from p. Suppose a point q is on this line. Let $d_p(q)$ be the signed distance of q from p, namely, $|p - q| \cdot sgn(v \cdot (q - p))$.

Then for the set of points Z_p on this line and also in Z, let $g_1(p) = \min_{z \in Z_p}(d_p(z))$ and $g_2(p) = \max_{z \in Z_p}(d_p(z))$.

Note that because Z is convex, it can be represented as the space $\cup_{p \in P}\{x : x \in p + [g_1(p), g_2(p)]\}$.

Then the volume of Z where a transaction in the direction of v of size k fails is:

$$\int_0^k \int_P \mathbb{1}(g_2(p) - g_1(p) \geq s)dp \, ds.$$

The derivative of this quantity at k is therefore

$$\int_P \mathbb{1}(g_2(p) - g_1(p) \geq k)dp$$

Clearly this quantity is bounded between ρ and 0, achieves ρ when $s = 0$, and is monotonically decreasing in k.

Note that a projection of a Minkowski sum of a set of vectors is the Minkowski sum of the vectors individually projected. It follows that $\rho = \Gamma_{x=y}(c)$ and that $f_v'(0)$ is $\Gamma_{x=y}(c)/\Gamma(c)$.

$1 - kf_v'(0)$ lower bounds transaction success probability because the volume where a transaction of size k in direction v fails is upper bounded by $k\Gamma_{x=y}(c)$, and thus transaction success probability is at least $(\Gamma(c) - k\Gamma_{x=y}(c))/\Gamma(c) = 1 - kf_v'(0)$.

□

In fact, the Lemma 5.7 directly generalizes 5.6.

COROLLARY 5.9. *The lower bound of Lemma 5.7 is tight if and only if there is an edge of capacity at least k in the direction of the transaction in the credit network graph.*

PROOF. The lower bound is clearly tight if the credit network graph contains an edge in the direction of the transaction, by Theorem 5.6.

Now suppose that the edge does not exist. Specifically, let Z again be the image of the set of all cycle equivalence classes, and let v be the direction of a transaction between vertices x and y. Suppose that the transaction has size k, and that edge from x to y, if it exists, has capacity $c(x,y) < k$.

The proof of Lemma 5.7 shows that volume of the space in Z where a transaction in direction v of size k fails is equal to

$$\int_0^k \int_P \mathbb{1}(g_2(p) - g_1(p) \geq s)dp \, ds.$$

with P, g_1, and g_2 defined as in the proof of Lemma 5.7.

The lower bound of Lemma 5.7 is tight if and only if

$$\int_0^k \int_P \mathbb{1}\left(g_2(p) - g_1(p) \geq s\right) dp\, ds = k \int_P \mathbb{1}(g_2(p) - g_1(p) \geq 0)\, dp \qquad (1)$$
$$= k\Gamma_{x=y}(c)$$

It suffices to find a point $z \in Z$ such that $z + (k/2)v \notin Z$ and $z - (k/2)v \notin Z$. Given such a point z, the point $z + (k/2)v$ is a point from which $z + (k/2)v - kv = z - (k/2)v \notin Z$. Because Z is closed, there is a point $p_0 \in P$ such that $g_2(p_0) - g_1(p_0) < k - \delta$. Hence there exists ε such that for $p \in P$ with $||p - p_0||_2 \leq \varepsilon$, then $g_2(p) - g_1(p) < k - \delta/2$, and thus the first equality of Equation 1 cannot hold.

Consider a credit network configuration defined as follows. Pick some vertex cut $H \subset V$ of the graph separating x from y. That is, a set of vertices such that all paths from x to y flow through H (other than possibly a direct path from x to y). Let m be the number of edges (h, i) such that $h \in H$ and $i \notin H$, and for every such edge (h, i), set $w(i, h)$ such that $w(i, h) \leq (k - c(x, y))/(2m + 1)$. Let $w(x, y) = w(y, x) = c(x, y)/2$. Pick w for every other edge arbitrarily.

Then from configuration w, strictly less than $c(x, y)/2 + m(k - c(x, y))/(2m) = k/2$ money can move from x to y or from y to x. This means that the point $z \in Z$ that is the image of w under our spanning representation has the property that $z + kv/2 \notin Z$ and $z - kv/2 \notin Z$ but $z \in Z$.

□

5.2 Monotonicity

Although we conjecture monotonicity in the general case, the same technical difficulty (Minkowski subtraction does not always produce zonotopes) that obstructed liquidity analysis presents itself when studying monotonicity. Nevertheless, we show that monotonicity holds when transactions are small or between vertices that already share a direct connection in the credit network graph.

THEOREM 5.10. *Suppose that τ is a transaction from x to y of size k and that $c(x, y) \geq k$. Let a and b be any two vertices (not necessarily connected directly, possibly equal to x or y).*
Then increasing $c(a, b)$ does not decrease the success probability of τ.

PROOF. If a and b are disconnected, connecting a to b with an edge of capacity 0 does not change the liquidity of any transaction, so without loss of generality assume a and b are connected by an edge of possibly 0 capacity.

Let Z be the image of the entire space of configurations before adding capacity on edge (a, b), and let $A \subset Z$ be the image of the space where τ succeeds. Let Y be the image of the entire space of configurations after capacity is increased along (a, b), and let $B \subset Y$ be the image of the space where τ succeeds after capacity is increased along (a, b).

For convenience, write $c = (c(e_1), ..., c(e_m))$ and $c' = (c(e_1), ..., c(x, y) - k, ..., c(e_m))$. Let $\mu(\cdot)$ denote the volume of a set in $\mathbb{R}^n$. Finally, let h be the increase in capacity along (a, b).

It suffices to show that $\mu(B)/\mu(Y) \geq \mu(A)/\mu(Z)$.

Note that Z is the Minkowski sum of the vectors $\{c(e)D(e)\}_{e \in E}$ and that A is the Minkowski sum of the vectors $\{c'(e)D(e)\}_{e \in E}$.

Then $\mu(Z) = \Gamma(c)$ and $\mu(A) = \Gamma(c') = \Gamma(c) - k\Gamma_{x=y}(c)$.

Moreover, adding w to A and Z to get B and Y means that B and Y, respectively, are the Minkowski sums of A and Z with the the vector hw. Hence, $\mu(Y) = \mu(Z) + h\Gamma_{a=b}(c)$ and $\mu(B) = \mu(A) + h\Gamma_{a=b}(c') = \Gamma(c) - k\Gamma_{x=y}(c) + h\Gamma_{a=b}(c) - kh\Gamma_{x=y,a=b}(c)$.

Rearranging terms shows that inequality holds if and only if $\Gamma_{x=y,a=b}(c)\Gamma(c) \leq \Gamma_{a=b}(c)\Gamma_{x=y}(c)$, which holds because the tree matroid is Rayleigh [9, 27].

□

More generally, we can prove the monotonicity conjecture if we bound transaction size.

THEOREM 5.11. *For every credit network G, there exists a constant $\hat{k}$ such that G is monotone with regard to transactions of size less than $\hat{k}$.*

PROOF. Let v be the direction of a path from x to y, and let w be the direction of a path from a to b. Let $c = \{c(e_i)\}_{e_i \in E}$. Finally, let $f_v(k, h)$ denote the probability that a transaction sending flow of size k in direction v fails in the graph where capacity h has been added to an edge in direction w.

First, note that by Lemma 5.7, at $k = 0$, $\frac{\partial}{\partial k} f_v(k, h) = (\Gamma_{x=y}(c) + h \Gamma_{x=y,a=b}(c)) / (\Gamma(c) + h \Gamma_{a=b}(c))$.

Because the tree matroid is Rayleigh [9], some computation shows that $\frac{\partial}{\partial k} f_v(k, 0) \leq \frac{\partial}{\partial k} f_v(k, h)$ at $k = 0$ for every $h > 0$. These two quantities are equal if and only if there are no cycles containing the edges (a, b) and (x, y), in which case monotonicity holds trivially.

Then for any fixed h, there exists some $\bar{k} > 0$ such that for all $k < \bar{k}$, $f_v(k, 0) \leq f_v(k, h)$.

Now, let A denote the volume where the transaction succeeds before the capacity addition, and let B denote the volume of the entire space before the capacity addition. Let C_h denote the marginal volume where the transaction succeeds after the capacity addition, and let D_h denote the marginal volume of the entire space after the capacity addition.

Importantly, because the space of configurations is a Minkowski sum, adding capacity simply adds an element to the Minkowski sum. The volume of any convex subset of the Minkowski sum, therefore, is increased simply by h times its projection in the direction w. Thus the ratio C_h/D_h is constant with respect to h.

Note that $f_v(k, h) = 1 - (A + C_h)/(B + D_h)$. Then for any k, if $f_v(k, h) \leq f_v(k, 0)$ for any $h > 0$, then $f_v(k, h) \leq f_v(k, 0)$ for all $h > 0$.

Putting this all together, there exists some $\hat{k} > 0$ such that the credit network is monotone with respect to all transactions along paths with transaction size less than $\hat{k}$.

□

6 EFFICIENT SAMPLING

In Theorem 5.5, we showed that for a narrow class of parameters, agents can compute the probability of transaction success. But this class of parameters was quite narrow, and likely not directly relevant to real-world networks. An agent in a payment network might like to be able to estimate this probability efficiently, so that they could measure the effects of behavior change under a wide variety of distributional assumptions.

In prior work on credit networks that assume that transactions have discrete sizes, estimating this probability is roughly equivalent to sampling a random forest of the underlying graph. A polynomial-time algorithm for this was recently discovered in [5]. This algorithm is quite interesting from a theoretical standpoint, but likely is still too slow to run on real-world graphs.

By contrast, we observe here that removing the discrete size requirement makes sampling a random network state quite efficient for a wide variety of parameter regimes. Where earlier work on credit networks invoked long lines of detailed mathematical machinery, sampling a random network state here simply requires running the well-studied hit-and-run sampling scheme ([38]).

THEOREM 6.1. *If π is a log-concave distribution on the set of cycle-equivalence classes of a credit network, then the hit-and-run sampling algorithm samples a random cycle-equivalence class from π.*

The algorithm requires a preprocessing step of time complexity $\widetilde{O}(n^5)$ preprocessing and (amortized) $\widetilde{O}(n^3 |E|^\omega)$ per sample (hiding log factors and dependencies on $1/\varepsilon$).

PROOF. The zonotope is a convex body, and thus this statement is simply a special case of the results of [31] and [32]. The algorithm, however, requires $\widetilde{O}(n^3)$ zonotope membership oracle calls. Checking membership in a zonotope can be solved with a linear program on $O(|E|)$ variables.

□

When the distribution π is uniform, we can improve our sample time complexity to $O(|E|^{1+\omega})$. The image under a spanning representation of the space of cycle-equivalence classes is not an arbitrary zonotope, but rather, has additional structure that we can exploit.

THEOREM 6.2. *If π is the uniform distribution on cycle-equivalence classes of a credit network, then there exists a sampling algorithm with time complexity $O(|E|^{1+\omega})$.*

This shaves a factor of $|V|^3/|E|$ off of the asymptotic sampling complexity. Note that real-world payment network graphs, like those deployed in Lightning, are typically quite sparse.

PROOF. Let $G = (V, E)$ be a credit network, and fix some ordering on the edges e_i and some spanning representation D. Furthermore, say that the first n elements of the ordering form a spanning tree. Let the edge capacity of edge e_i be $c(e_i)$, and let the Minkowski sum of the first k edges be Z_k. It suffices to sample a uniformly random point within Z_m.

Let e_i be any edge. Let S_i denote the portion of the surface of Z_i that is "visible" from the direction of e_i. (The surface of a zonotope is a collection of faces. A face forms part of S_i if its normal vector n has $n \cdot D(e_i) \geq 0$). Note that Z_i can be decomposed into the almost disjoint union of two pieces. The first is Z_{i-1}, and the second is the remainder, which is the Minkowski sum of the set S_{i-1} and the vector $D(e_i)c(e_i)$. Their intersection is a translation of S_i.

As noted in Corollary 5.6, the ratio of the volume of the second part to the total volume of Z_i is computable via an effective resistance calculation.

Let P_i be the projection of S_i onto the hyperplane through the origin with normal $D(e_i)$. A random point in this second part can be sampled by taking a random point in P_{i-1}, mapping back to the corresponding point in S_{i-1}, and adding $\lambda_i c(e_i)D(e_i)$ for λ_i drawn from $(0, 1]$ uniformly at random.

The mapping from points in P_i back to points in S_i can be computed via a linear program. Specifically, given a point p, maximize α such that $p + \alpha D(e_i)$ is a point within Z_i. This linear program has $O(n+m)$ constraints and m decision variables, so solving this *LP* takes time $O(m^\omega)$, as per the recent algorithm of [11]. Of course, one could substitute in any linear program solver, which may change empirical performance but give a different asymptotic complexity result.

Conveniently, both Z_{i-1} and P_{i-1} are zonotopes, so our sampling process naturally sets itself up for a recursive sampling algorithm.

An algorithm $\mathscr{A}$ for sampling a random point from the Minkowski sum, therefore, could look as follows. Given as input vectors $v_1, ..., v_k \in \mathbb{R}^n$ and capacities $c(e_1), ..., c(e_k)$, $\mathscr{A}$ will return a uniformly random point in the Minkowski sum of the input vectors. For notation, let this Minkowski sum again be Z_k and its surface in the direction $D(e_k)$ be S_k. First, check whether the input set of vectors is linearly independent. If they are, return a random list of k values drawn uniformly at random from $[0, 1]$.

Next, compute the effective resistance r along edge k. Use this to choose between the spaces $Z_k \setminus Z_{k-1}$ and Z_{k-1} randomly proportionally to their volumes.

If the algorithm chooses $Z_k \setminus Z_{k-1}$, the algorithm can invoke $\mathscr{A}$ on the list of vectors $(w_1, ..., w_{k-1})$, where w_i is the component of v_i perpendicular to v_k. It can then find the corresponding point s in S_k via the above linear program.

Finally, the algorithm can sample λ_k uniformly at random from $(0, 1]$ and return $s + \lambda_k D(e_k)$.

If we do not accept e_k, simply return $\mathscr{A}(w_1, ..., w_{k-1}, c(e_1), ..., c(e_{k-1}))$.

Computing a spanning representation takes time $O(nm)$. Each step of this algorithm takes time $O(m^\omega + m)$ for solving the linear program, computing effective resistance, and processing a list of vectors. Since the algorithm recurses m times, the total time complexity is $O(m^{\omega+1})$. □

7 FUTURE WORK, AND A NOTE ON SYMMETRICITY

Our arguments for the monotonicity conjecture require that the Markov-chain-induced probability measure on the set of configurations be uniform over cycle-equivalent classes of configurations. We showed that this occurs when the transaction distribution is symmetric. However, real-world transaction distributions may not be symmetric.

In the real world, transactions on a payment network do not necessarily happen in sequence. We chose to analyze a discrete-time transaction model for conceptual simplicity and to follow the pattern of prior work on credit networks; however, we could have analyzed a continuous-time transaction model where in some window of time, pairs of agents can transact many times. Within a fixed time window, the net transaction balance between every pair of agents u and v might be randomly distributed. We could then sum these biases, weighted by the direction of a transaction from u to v, to get an overall direction of bias $v \in \mathbb{R}^n$. Another possible model for transactions, therefore, is a "Reflected Brownian Motion" with drift vector v (and normal reflection). It follows directly from [23] that the invariant distribution $\mu(x)$ of an RBM with bias v is proportional to $e^{x \cdot v}$.

Of course, in the symmetric distribution case, v is 0 and the distribution is again uniform. If v is nonzero, then the invariant distribution is efficiently sampleable using an algorithm like Hit and Run. Some asymmetric transaction distributions still have a net bias of 0. Finding realistic real-world assumptions that would be sufficient to imply uniformity of the induced stationary measure, or analyzing monotonicity when the induced stationary measure is nonuniform, are interesting open research problems. General models of Brownian motion in a bounded space have been studied in other contexts, such as network queuing theory. A credit network is not exactly a traditional queuing network, but nevertheless, understanding connections with results in queuing theory or the theory of diffusion processes might prove fruitful.

The authors also think that an interesting line of future work would be to study cases where agent behavior changes based on the current configuration of the network. Under mild technical conditions, the notion of an induced invariant measure when agent behavior varies with state is still well-defined, and analogous concepts have been studied in other literature (such as [25]).

8 CONCLUSION

The Credit Network model forms an accurate representation of the combinatorics underlying modern "Layer-2" transaction protocols. The performance of these networks is an important piece of the process of scaling up cryptocurrency deployments.

Prior work was only able to analyze the scenario where transactions have discrete sizes, and could only give meaningful liquidity bounds when transactions had a single fixed size that was not too small relative to edge capacities. Not only is this not a natural assumption, but the resulting analysis left open the important question of monotonicity and the problem of sampling efficiently random network configurations. In this work, we show that not only is the analogous continuous model analytically tractable, but that removing the assumption of discrete transaction sizes actually enables efficient sampling and progress on the monotonicity conjecture.

The results are obtained by exploiting a relationship between the graphs underlying payment networks and particular convex sets in $\mathbb{R}^n$. The problems of monotonicity and sampling thereby transform into problems of analyzing the volume of convex sets. This transformation relies crucially on the combinatorial properties of the underlying graphic matroid.

The authors hope that connections to other branches of mathematics, such as diffusion processes, that the continuous relaxation facilitates could prove fruitful areas of research, such as when studying state-dependent agent behavior or transaction pricing.

ACKNOWLEDGMENTS

REFERENCES

[1] Connext network, https://connext.network/
[2] Introducing outpace: Off-chain unidirectional trustless payment channels, https://medium.com/the-adex-blog/introducing-outpace-off-chain-unidirectional-trustless-payment-channels-243a08e152a
[3] Raiden network, https://raiden.network/
[4] Althoff, M.: On computing the minkowski difference of zonotopes. arXiv preprint arXiv:1512.02794 (2015)
[5] Anari, N., Liu, K., Gharan, S.O., Vinzant, C.: Log-concave polynomials ii: High-dimensional walks and an fpras for counting bases of a matroid. arXiv preprint arXiv:1811.01816 (2018)
[6] Braess, D., Nagurney, A., Wakolbinger, T.: On a paradox of traffic planning. Transportation science **39**(4), 446–450 (2005)
[7] Brânzei, S., Segal-Halevi, E., Zohar, A.: How to charge lightning. arXiv preprint arXiv:1712.10222 (2017)
[8] Buterin, V., et al.: A next-generation smart contract and decentralized application platform
[9] Choe, Y., Wagner, D.G.: Rayleigh matroids. Combinatorics, Probability and Computing **15**(5), 765–781 (2006)
[10] Cocks, C.C.: Correlated matroids. Combinatorics, Probability and Computing **17**(4), 511–518 (2008)
[11] Cohen, M.B., Lee, Y.T., Song, Z.: Solving linear programs in the current matrix multiplication time. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. pp. 938–942. ACM (2019)
[12] Coinbase: Charting the course of bitcoin, 11 years and counting, https://blog.coinbase.com/charting-the-course-of-bitcoin-11-years-and-counting-b4e17969d4e1
[13] Dandekar, P., Goel, A., Govindan, R., Post, I.: Liquidity in credit networks: A little trust goes a long way. In: Proceedings of the 12th ACM conference on Electronic commerce. pp. 147–156. ACM (2011)
[14] DeFigueiredo, D.B., Barr, E.T.: Trustdavis: A non-exploitable online reputation system. In: E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on. pp. 274–283. IEEE (2005)
[15] Doulliez, P., Jamoulle, E.: Transportation networks with random arc capacities. Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle **6**(V3), 45–59 (1972)
[16] Eberle, A.: Markov processes. Lecture Notes at University of Bonn (2017)
[17] as Feder, T., Mihail, M.: Balanced matroids. Citeseer
[18] Ghosh, A., Mahdian, M., Reeves, D.M., Pennock, D.M., Fugger, R.: Mechanism design on trust networks. In: International Workshop on Web and Internet Economics. pp. 257–268. Springer (2007)
[19] Goel, A., Khanna, S., Raghvendra, S., Zhang, H.: Connectivity in random forests and credit networks. In: Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms. pp. 2037–2048. Society for Industrial and Applied Mathematics (2015)
[20] Grimmett, G.R., Winkler, S.: Negative association in uniform forests and connected graphs. Random Structures & Algorithms **24**(4), 444–460 (2004)
[21] Grimmett, G., Suen, W.S.: The maximal flow through a directed graph with random capacities. Stochastics: An International Journal of Probability and Stochastic Processes **8**(2), 153–159 (1982)
[22] Harris, T.E.: The existence of stationary measures for certain markov processes
[23] Harrison, J.M., Williams, R.J.: Multidimensional reflected brownian motions having exponential stationary distributions. The Annals of Probability pp. 115–137 (1987)
[24] Hassin, R., Zemel, E.: Probabilistic analysis of the capacitated transportation problem. Mathematics of operations research **13**(1), 80–89 (1988)
[25] Kang, W., Ramanan, K., et al.: Characterization of stationary distributions of reflected diffusions. The Annals of Applied Probability **24**(4), 1329–1374 (2014)
[26] Karlan, D., Mobius, M., Rosenblat, T., Szeidl, A.: Trust and social collateral. The Quarterly Journal of Economics **124**(3), 1307–1361 (2009)
[27] Kirchhoff, G.: Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. Annalen der Physik **148**(12), 497–508 (1847)
[28] Kleitman, D.J., Winston, K.J.: Forests and score vectors. Combinatorica **1**(1), 49–54 (1981)

[29] Li, K.: The blockchain scalability problem & the race for visa-like transaction speed, https://hackernoon.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44

[30] Lin, J.S., Jane, C.C., Yuan, J.: On reliability evaluation of a capacitated-flow network in terms of minimal pathsets. Networks **25**(3), 131–138 (1995)

[31] Lovász, L., Vempala, S.: Hit-and-run is fast and fun. preprint, Microsoft Research (2003)

[32] Lovász, L., Vempala, S.: The geometry of logconcave functions and sampling algorithms. Random Structures & Algorithms **30**(3), 307–358 (2007)

[33] Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)

[34] Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)

[35] Ramseyer, G., Goel, A., Mazières, D.: Liquidity in credit networks with constrained agents. In: Proceedings of The Web Conference 2020. pp. 2099–2108 (2020)

[36] Semple, C., Welsh, D.: Negative correlation in graphs and matroids. Combinatorics, Probability and Computing **17**(3), 423–435 (2008)

[37] Shephard, G.C.: Combinatorial properties of associated zonotopes. Canadian Journal of Mathematics **26**(2), 302–321 (1974)

[38] Smith, R.L.: Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. Operations Research **32**(6), 1296–1308 (1984)

[39] Stanley, R.P.: Decompositions of rational convex polytopes. Ann. Discrete Math **6**(6), 333–342 (1980)

A SECTION 3 OMITTED PROOFS

A.1 Proof of Theorem 3.3

PROOF. There are, in fact, many spanning representations. This is unsurprising, given that graphical matroids are representable over $\mathbb{R}$, and the dimension of any basis of the graphical matroid is n. It suffices to show that there exists a matroid representation such that summing along a cycle results in 0, not just that the set of vectors corresponding to a cycle forms a linearly dependent set.

As a concrete example, let T be a spanning tree, and let $v_1, ..., v_n$ be a basis for $\mathbb{R}^n$. Correspond the edges in T with $v_1, ..., v_n$. For any other edge (x, y) with $s(x, y) = 1$, let $p_0 = x, ...p_t = y$ be the path from x to y in T, and set $D(x, y) = \sum_{i=0}^{t-1} s(p_i, p_{i+1}) D(p_i, p_{i+1})$.

Then clearly the direction walking along the cycle formed by T with (x, y) is 0.

Note that the direction summation is clearly additive. In other words, if a path p runs from x to y and a path q runs from y to z, then the direction of p plus the direction of q is equal to the direction of the concatenation of p and q.

Observe that walking from x to y, and then back along the same path, gives a path whose sum is 0. In fact, any path from x to y that visits a vertex v twice can be decomposed into a path directly from x to y plus a path from v to itself. Sufficiently repeating this decomposition shows that the sum along every path in T from x to y has the same summation. Note that because $v_1, ..., v_n$ form a basis for $\mathbb{R}^n$, this summation is 0 if and only if $x = y$.

For any other path $p = (p_0, ..., p_t)$, if $(p_i, p_{i+1}) \notin T$, then, by the construction, the summation of the path along p is equal to the summation along the path $p' = (p_0, ..., p_i, q_1, ...q_k, p_{i+1}, ..., p_t)$, where $(p_i, q_1, ...q_t, p_{i+1})$ is a path from p_i to p_{i+1} in T. Repeating this process until all edges are in T shows that p has direction 0 if and only if p is a cycle.

$\square$

A.2 Proof of Theorem 3.5

PROOF.

1 This follows from part 2. Assuming part 2, if transactions along cycles take w_1 to w_2, then, because cycles have direction 0, $S(w_1) = S(w_1) + 0 = S(w_2)$.

2 Suppose that a path $p = (p_0, ..., p_t)$ is feasible in configuration w_1.

By rearranging the summations, we get that

$$S(w_1) - S(w_2) = \Sigma_i s(p_i, p_{i+1})k * D(p_i, p_{i+1})$$
$$= k * D(p)$$

3 Let p' be the reversal of p. Then p concatenated with p' forms a cycle, and q concatenated with p' forms a cycle, so

$$D(p) = -D(p')$$
$$= D(q)$$

$\square$

A.3 Proof of Theorem 3.7

PROOF. This theorem follows almost directly from the construction. If a vector v is expressable as a Minkowski summation with coefficients $\{\lambda_{(x,y)}\}_{(x,y)\in E,s(x,y)=1}$, then the weight map w, defined by $w(x,y) = \lambda_{(x,y)}c(x,y)$ if $s(x,y) = 1$ and $c(x,y) - w(y,x)$ otherwise, forms a feasible network configuration. Conversely, if $S(w) = \sum_{(x,y)\in E:s(x,y)=1} \lambda_{(x,y)}c(x,y)D(x,y)$, then $\lambda_{(x,y)} \in [0,1]$, so $S(w)$ is representable as a Minkowski sum of the vectors of interest. $\square$

A.4 Proof of Corollary 3.8

PROOF. Let $Z \subset \mathbb{R}^n$ be the space of configurations under a spanning representation. Let x_1 and x_2 be in Z. Clearly, if $x_1 = x_2$, that is, x_1 and x_2 are cycle-equivalent, then for any transaction v, $x_1 + v \in Z$ if and only if $x_2 + v \in Z$.

Suppose that x_1 and x_2 are transaction-equivalent but not cycle-equivalent. Then the generalized transaction $v = x_2 - x_1$ is feasible starting from x_1 and takes x_1 to x_2. Let $k = \sup\{t : tv + x_2 \in Z\} + 1/2$. Because Z is convex, the transaction $kv \neq 0$ is feasible starting from x_1 but not from x_2, which gives a contradiction. $\square$

B SECTION 4 OMITTED PROOFS

B.1 Proof of Theorem 4.3

PROOF. Let $\Xi = (X, \phi, \{k_{a,b}\}_{(a,b)\in X})$ be a transaction model on V, and let D be some spanning representation.

Let p be the transition kernel of Markov Chain 1 using transaction distribution Ξ, and let g be any continuous, bounded function on the compact state space Z. By Theorem 1.10 of [16], it suffices to show that if a sequence x_n converges to x, then $\int g(y)p(x_n,dy)$ goes to $\int g(y)p(x,dy)$. But p is the weighted sum of continuous, bounded functions, so this holds.

$\square$

B.2 Proof of Theorem 4.4

PROOF. Let $\Xi = (X, \phi, \{k_{a,b}\}_{(a,b)\in X})$ be a transaction model on V, let p be the transition kernel of Markov Chain 1 using transaction distribution Ξ, and let D be some spanning representation.

Let Z be the image under D of the entire space of configurations, and let μ be some invariant distribution.

Note that it is equivalent to view the set of transacting pairs, X, as a set of vectors X'. Moreover, (V,X) is connected if and only if X' spans Z.

By Theorem 1 of [22], it suffices to show that if a set $E \subset Z$ has nonzero measure, then from any starting point $z \in Z$, the Markov Chain must visit E infinitely many times in expectation.

Note that it suffices to show that for all $z \in Z$, there is some k such that $p^k(z,E) \neq 0$. If this holds for all z, then in expectation, the chain starting at z will visit E in a finite number of steps. After this,

the chain could either stay in E forever (in which case, the chain trivially visits E infinitely many times), or it will leave E to some z'.

Suppose that X' spans Z.

Because each f_v is supported on $[-\varepsilon, \varepsilon]$, for any z, the conditional distribution started at z and running for $|X|$ steps has support on an L_∞ ball (subset of Z) of radius ε around z. By the same argument, after $k|X|$ steps, the conditional distribution started at z has support on an L_∞ ball (subset of Z) of radius $k\varepsilon$. Because Z is bounded, for sufficiently large k, the support of the conditional distribution has measure equal to $\mu(Z)$.

Suppose Markov Chain 1 only returns to some set E finitely many times in expectation. Then E is not in the support of the conditional distribution, and must have measure 0.

$\square$

B.3   Proof of Theorem 4.5

PROOF. Let $\Xi = (X, \phi, \{k_{a,b}\}_{(a,b) \in X})$ be a transaction model on V, let p be the transition kernel of Markov Chain 1 using transaction distribution Ξ, and let D be some spanning representation.

It suffices to show that for any measurable set A, $\mu(A) = \int_Z p(dz, A)$, where $\mu(\cdot)$ is a uniform measure on Z.

A transaction corresponding to a vector v starting from point z lands in A if either $z - v \in A$ – that is, the transaction succeeds and results in a state in A – or if $z \in A$ and $z - v \notin Z$ – that is, the transaction fails and the initial state was already in A. Note also that for any fixed v, these events are mutually exclusive. Hence, the sets $A_v^1 = \{x : x - v \in A\} \cap Z$ and $A_v^2 = \{x : x - v \notin Z\} \cap A$ are disjoint.

Note also that $A_{-v}^1 = \{x : x + v \in A\} \cap Z$.

Translating by $x \to x + v$ gives the set $\{y : y \in A, y - v \in Z\}$. Because μ is uniform on Z and both of these sets are fully contained in Z, we get that

$$\mu(A_{-v}^1) = \mu(\{y : y \in A, y - v \in Z\})$$

Note also that the sets $\{y : y \in A, y - v \in Z\}$ and A_v^2 form a disjoint partition of A. Hence, $\mu(A) = \mu(A_{-v}^1) + \mu(A_v^2)$, and thus

$$2\mu(A) = \mu(A_v^1) + \mu(A_v^2) + \mu(A_{-v}^1) + \mu(A_{-v}^2) \tag{2}$$

Expanding terms thus gives

$$\int_Z p(dZ, A)$$

$$= \sum_{(a,b) \in X} \phi(a,b) \int_Z \left(\int_{-\infty}^{\infty} k_{a,b}(c) \mathbb{1}\{z - cD(a,b) \in A \vee (z - cD(a,b) \notin Z \wedge z \in A\}dc \right) dz$$

Changing order of integration, we get

$$\sum_{(a,b) \in X} \phi(a,b) \int_{-\infty}^{\infty} k_{a,b}(c) \int_Z \mathbb{1}\{(z - cD(a,b) \in A \wedge z \in Z) \vee (z - cD(a,b) \notin Z \wedge z \in A\}dzdc.$$

Because $k_{a,b}$ is symmetric, using equation 2, we can rearrange the integral above into

$$\sum_{(a,b) \in X} \phi(a,b) \int_0^{\infty} k_{a,b}(c)(2\mu(A))dc = \mu(A)$$

Hence, μ is stationary in Markov Chain 1.

$\square$

B.4 Proof of Theorem 4.7

PROOF. First, note that for a spanning tree T, under any spanning representation, the vectors corresponding to T form a basis for $\mathbb{R}^n$. Moreover, the behavior of a spanning representation on one spanning tree uniquely determines its behavior on all other edges. Thus, any spanning representation is the result of a linear transformation of another spanning representation.

Let D_1 and D_2 be two spanning representations, and let M be the (nonsingular) matrix such that for all $e \in E$, $M * D_1(e) = D_2(e)$. Let v_1 be the vector under D_1 corresponding to some transaction, and let $v_2 = Mv_1$ be vector under D_2 for that same transaction. Let Z_1 be the space of represented cycle-equivalent configurations under D_1, and let $Z_2 = MZ_1$ be the space of represented cycle-equivalent configurations under D_2. Finally, let π_1 be a density function on Z_1 and let π_2 be the density function on Z_2 such that $\pi_2(Mx) = \pi_1(x)$ for $x \in Z_1$.

It suffices to show, therefore, that

$$\frac{\int_{Z_1} \pi_1(z)\mathbb{1}\{z - v_1 \in Z_1\}\,dz}{\int_{Z_1} \pi_1(z)dz} = \frac{\int_{Z_2} \pi_2(z)\mathbb{1}\{z - v_2 \in Z_2\}\,dz}{\int_{Z_2} \pi_2(z)dz}$$

The argument follows from a change of variables. We note that stretching the representation space should necessarily stretch the density function. Note that

$$\int_{Z_2} \pi_2(z)\mathbb{1}\{z - v_2 \in Z_2\}dz = \int_{Z_1} \pi_2(Mz)\mathbb{1}\{Mz - Mv_1 \in MZ_1\}|det(M)|dz$$

$$= |det(M)|\int_{Z_1} \pi_1(z)\mathbb{1}\{z - v_1 \in Z_1\}dz$$

and

$$\int_{Z_2} \pi_2(z)dz = \int_{Z_1} \pi_2(Mz)|det(M)|dz$$

$$= |det(M)|\int_{Z_1} \pi_1(z)dz$$

Because M is invertible and $|det(M)| \neq 0$, the equality holds.

$\square$

C   SECTION 5 OMITTED PROOFS

C.1   Proof of Theorem 5.3

PROOF. If A is a set of n vectors in $\mathbb{R}^n$, define $det(A)$ as the determinant of the $n \times n$ matrix where elements of A are interpreted as column vectors.

By (57) of [37], the volume of the Minkowski sum of a set of vectors $S = \{x_1, ..., x_m\} \subset \mathbb{R}^n$ is equal to $\Sigma_{A \subset S:|A|=n}|det(A)|$. The rest of this proof follows from basic facts about the determinant.

Let S be the set of vectors corresponding to the directions under a spanning representation $\hat{D}$ of edges in the underlying payment network G. By assumption, $\hat{D}$ maps some spanning tree to the standard basis $B = \{b_1, ..., b_n\}$.

For any linearly independent set $A \subset S$ of size n, we would like to show that $|det(A)| = |det(B)| = 1$. Note that a set of vectors in S is linearly independent if and only if the set is the set of directions under $\hat{D}$ of a spanning tree in G.

Suppose for induction that the set of vectors A_i is the set of directions of a spanning tree T_{A_i} in G, A_i contains $\{b_1, ..., b_i\}$, $A_i \setminus \{b_1, ..., b_i\} \subset A$, and $|det(A_i)| = |det(A)|$. Set $A_0 = A$.

If $b_{i+1} \in A_i$, then set A_{i+1} to A_i and the induction hypothesis holds. Otherwise, viewed as edges in G, the set $\{b_{i+1}\} \cup A_i$ contains a single simple cycle. Say that the vectors corresponding to edges in this cycle are $b_{i+1}, c_1, ..., c_k$. Because $\hat{D}$ is a spanning representation, $b_{i+1} = \Sigma_{j=1}^{k} \alpha_j c_j$ for $\alpha_j = \pm 1$.

Set $A_{i+1} = (A_i \cup \{b_{i+1}\}) \setminus \{c_k\}$. Then by the linearity of the determinant, we can substitute the column corresponding to c_k with the summation $c_k = 1/\alpha_k \Sigma_{j=1}^{k-1} \alpha_j c_j - b_{i+1}$ and expand to get that $det(A_i) = \Sigma_{j=1}^{k-1} \alpha_j * 0 + 1/\alpha_k det(A_{i+1})$. Hence $|det(A_{i+1})| = |det(A_i)| = 1$, and the induction hypothesis holds.

Since $A_n = B$, the above induction shows that $|det(A)| = |det(B)| = 1$.

As such, the nonzero elements of the summation $\Sigma_{A \subset S: |A|=n} |det(A)|$ correspond exactly to the spanning trees of the underlying graph. This proves the theorem in the case where every edge has capacity 1. The general result follows from the multilinearity of the determinant.

□

C.2   Proof of Theorem 5.5

PROOF. This is a general fact about Minkowski summation, but we include here a proof for completeness.

Let G be the original credit network where $c(x,y) \geq k$, and let H be a credit network where the capacity along (x,y) has been reduced by k (i.e. the capacities of edges in H are c'). Without loss of generality, assume v corresponds to a transaction in the direction along (x,y) that the spanning representation denotes as positive.

Let Z be the image of the set of configurations of G, and let $Z \cap \{z : z - v \in Z$ be the image of the set of configurations of G from which transaction v succeeds.

Let Y be the image of the set of configurations of H (under the same spanning representation).

Clearly $Y \subset Z \cap \{z : z - v \in Z$ (directly as subsets of $\mathbb{R}^n$). If w is a configuration of H, then $w(x,y) \leq c(x,y) - k \leq c(x,y)$, so the configuration $w' = w$ except $w'(y,x) = w(y,x) + k$ is a configuration of G that maps to the same point in $\mathbb{R}^n$.

To show that $Z \cap \{z : z - v \in Z \subset Y$, it suffices to show that each $z \in Z \cap \{z : z - v \in Z$ is the image of some configuration of G where $w(x,y) \leq c(x,y) - k$. Given such a configuration, we can construct a configuration w' as $w' = w$ except that $w'(y,x) = w(y,x) - k$. Then w' is a configuration of H.

Let w be some configuration of G from which transaction v is successful. Then from configuration w, x can send at least k units of money to y. If $w(x,y) > c - k$, then there must be some other paths in w that enable x to send $k - (c - w(x,y))$ units of money to y. But then w is cycle-equivalent to a configuration w' constructed from w by sending $k - (c - w(x,y))$ units of money from x to y through paths not using edge (x,y) and then back to x along edge (y,x).

Hence, $Y = Z \cap \{z : z - v \in Z$.

□