Computers & Industrial Engineering 136 (2019) 160–172

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Equilibria in the tangle

Serguei Popov[a,*], Olivia Saa[b], Paulo Finardi[c]

[a] Department of Statistics, Institute of Mathematics, Statistics and Scientific Computation, University of Campinas – UNICAMP, rua Sérgio Buarque de Holanda 651, 13083–859 Campinas, SP, Brazil
[b] Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo – USP, rua do Matão 1010, 05508–090 São Paulo, SP, Brazil
[c] Institute of Computing – IC, University of Campinas – UNICAMP, av. Albert Einstein 1251, 13083–852 Campinas, SP, Brazil

ARTICLE INFO

Keywords:
Random walk
Nash equilibrium
Directed acyclic graph
Cryptocurrency
Tip selection
IOTA

AMS 2010 subject classifications:
Primary 91A15
Secondary 60J20
68M14

ABSTRACT

We analyse the Tangle — a DAG-valued stochastic process where new vertices get attached to the graph at Poissonian times, and the attachment's locations are chosen by means of random walks on that graph. These new vertices, also thought of as "transactions", are issued by many players (which are the nodes of the network), independently. The main application of this model is that it is used as a base for the IOTA cryptocurrency system.[1] We prove existence of "almost symmetric" Nash equilibria for the system where a part of players tries to optimize their attachment strategies. Then, we also present simulations that show that the "selfish" players will nevertheless cooperate with the network by choosing attachment strategies that are similar to the "recommended" one.

1. Introduction

In this paper we study *the Tangle*, a stochastic process on the space of (rooted) Directed Acyclic Graphs (DAGs). This process "grows" in time, in the sense that new vertices are attached to the graph according to a Poissonian clock, but no vertices/edges are ever deleted. When that clock rings, a new vertex appears and attaches itself to locations that are chosen with the help of certain random walks on the state of the process in the *recent past* (this is to model the network propagation delays); these random walks therefore play the key role in the model.

Random walks on random graphs can be thought of as a particular case of Random Walks in Random Environments: here, the transition probabilities are functions of the graph only, i.e., there are no additional variables, such as conductances[2] etc., attached to the vertices and/or edges of the graph. Still, this subject is very broad, and one can find many related works in the literature. One can mention the internal DLA models (e.g. Jerison et al., 2014 and references therein), random walks on Erdös-Rényi graphs (Cooper et al., 2017; Jerison et al., 2014), or random walks on the preferential attachment graphs (Cooper and Frieze, 2007), which most closely resembles the model of this paper.

The motivation for studying the particular model presented in this paper stems from the fact that it is applied in the IOTA cryptocurrency (Popov, 2015). The IOTA is an ambitious project started in 2015, it aims to provide a globally scalable system capable of processing payments and storing data. One of its distinguishing features is that it uses (nontrivial) DAGs as the primary ledger for the transactions' data.[3] This is different from "traditional" cryptocurrencies such as the Bitcoin, where that data is stored in a sequence of blocks,[4] also known as *blockchain*. An important observation, which motivates the use of more general DAGs instead of blockchains is that the latter *scale* poorly. Indeed, it is not hard to see that the chain of blocks of finite size, which can only be produced at regular discrete time intervals, produces a throughput bottleneck and leads to high transaction fees that need to be paid to the miners (which is by design). Also, when the network is large,

* Corresponding author.
E-mail addresses: popov@ime.unicamp.br, serguei.popov@iota.org (S. Popov), olivia@ime.usp.br, olivia.saa@iota.org (O. Saa), ra144809@ic.unicamp.br (P. Finardi).

[1] http://www.iota.org/.

[2] This refers to the well-known relation between reversible Markov chains and electric networks, see e.g. the classical book Doyle and Snell (1984).

[3] We also cite Baird (2016), Churyumov (2016), Lerner (2015) and Sompolinsky et al. (2016) which deal with other approaches to using DAGs as distributed ledgers.

[4] That is, the underlying graph is essentially $\mathbb{Z}_+$ (after discarding finite forks).

https://doi.org/10.1016/j.cie.2019.07.025

Available online 13 July 2019
0360-8352/

it is difficult for it to achieve consensus on which blocks are "valid" in the situations when the new blocks come too frequently. If one wants to remove the fees and allow the system to scale, the natural idea would thus be to eliminate the bottleneck and the miners.

This is, of course, easier said than done — it raises all sorts of new questions. Where should the next block/transaction/vertex be attached? Who will vet the transactions for consistency and why? How can it be secure against possible attacks? How will consensus be achieved? These questions do not have trivial answers. The paper Popov (2015) presented an idea for an architecture which could potentially resolve these issues. In that system, each transaction, represented by a vertex in the graph, would approve two previous transactions it selects using a particular class of random walks. To eliminate the transaction fees, it was necessary to first eliminate the miners— after all, if one wants to design a feeless system, there cannot be a dichotomy of "miners" who serve the "simple users". This bifurcation of roles between the miners and transactors naturally leads to transaction fees because the miners have some kind of resource that others do not have and they will use this monopoly power to extract rents, in the form of transaction fees, or block rewards, or both. Therefore, to eliminate the fees, all the users would have to fend for themselves. The main principle of such a system would be "help others, and others will help you".

You can help others by approving their transactions; others can help you by approving your transactions. Let us call "tips" the transactions which do not yet have any approvals; all new-coming transactions are tips at first. The idea is that, by approving a transaction, you also indirectly approve all its "predecessors". It is intuitively clear that, to help the system progress, the incoming transactions must approve tips because this adds new information to the system. However, due to the network delays, it is not practical to *impose* that this must happen— how can one be sure that what one believes to be a tip has not already been approved by someone else maybe 0.1 s ago?.

In any case, if everybody collaborates with everybody— only approving recent and "good" (non-contradicting) transactions, then we are in a good shape. On the other hand, for someone who only cares about themself, a natural strategy would just be to choose a couple of old transactions and approve them all the time without having to do the more cumbersome work of checking new transactions for consistency thereby adding new information to the system. If everybody behaves in this way, then no new transactions will be approved, and the network will effectively come to a halt. Thus, if we want it to work, we need to incentivize the participants to collaborate and approve each others recent transactions. Therefore, in some sense, it is all about the incentives. Everybody wants to be helped by others, but, not everybody cares about helping others themselves. To resolve this without having to introduce monetary rewards, we could instead think of a reward as simply not being punished by others. So, we need to slightly amend the above main principle— it now reads: "Help others, and the others will help you; however, if you choose not to help others, others will not help you either". When a new transaction references two previous transactions, it is a statement of "I vouch for these transactions which have not been vouched for before, as well as all their predecessors, and their success is tied to my success". It was suggested in Popov (2015) that the Markov Chain Monte Carlo (MCMC) tip selection algorithm (more precisely, the family of tip selection algorithms) would have these properties.

This paper mainly deals with the following question: what if some participants of the network are trying to minimize their *costs* by adopting a behavior different from the "default" one? How will the system behave in such circumstances? In other words, are there enough incentives for the participants to "behave well"? To address these kinds of questions, we first provide general arguments to prove existence of "almost symmetric" Nash equilibria for the system, see Section 3. Although one can hardly access the explicit form of these equilibria in a purely analytical way, simulations presented in Section 4 show that the "selfish" players will typically still choose attachment strategies that are similar to the default one, meaning that they would prefer *cooperating* with the network rather than simply *using* it.

Let us stress also that, in this paper, we consider only "selfish" players, i.e., those who only care about their own costs but still want to use the network in a legitimate way.[5] We do not consider the case when there are "malicious" ones, i.e., those who want to disrupt the network even at a cost to themselves. We are going to treat several types of attacks against the network in the subsequent papers.

This paper is organized in the following way. In Section 2 we first introduce some notations and define the objects we are working with; then, in Section 2.1 we describe the "recommended" algorithm of how the nodes choose where to attach a new transaction, and then discuss some basic properties of it, also formulating an open problem about the asymptotic behavior of the total number of tips. Section 3 contains the main "theoretical" advances of this paper. There, we first discuss what is a "strategy" that could be used by a selfish player, and then (Section 3) make some further assumptions necessary to formulate our main results (which are placed in Section 3.2). Then, we prove these results in Section 3.3. Section 4 discusses some simulation results, mainly in the case where the selfish players try to use a very natural "greedy" attachment strategy (Section 4.1). In Section 5 one will find conclusions and some final remarks.

2. Description of the model

In the following we introduce the mathematical model describing the Tangle (Popov, 2015).

Let $\mathrm{card}(A)$ stand for the cardinality of (multi) set A. Consider an oriented multigraph $\mathcal{T} = (V, E)$, where V is the set of vertices.[6] and E is the multiset of edges. For $u, v \in V$, we say that u *approves* v, if $(u, v) \in E$. For a vertex $v \in V$, let us denote by

$$\deg_{\mathrm{in}}(v) = \mathrm{card}\{e = (u_1, u_2) \in E\colon u_2 = v\},$$
$$\deg_{\mathrm{out}}(v) = \mathrm{card}\{e = (u_1, u_2) \in E\colon u_1 = v\}$$

the "incoming" and "outgoing" degrees of the vertex v (counting the multiple edges). In the following, we refer to multigraphs simply as graphs. We use the notation $\mathcal{A}(u)$ for the set of the vertices approved by u. We say that $u \in V$ *references* $v \in V$ if there is a sequence of sites $u = x_0, x_1, \ldots, x_k = v$ such that $x_j \in \mathcal{A}(x_{j-1})$ for all $j = 1, \ldots, k$, i.e., there is a directed path from u to v. If $\deg_{\mathrm{in}}(w) = 0$ (i.e., there are no edges pointing to w), then we say that $w \in V$ is a *tip*.Let $\mathcal{G}$ be the set of all directed acyclic graphs (also known as DAGs, that is, oriented graphs without cycles) $G = (V, E)$ with the following properties (see Fig. 1).

- The graph G is finite and the multiplicity of any edge is at most two (i.e., there are at most two edges linking the same vertices).
- There is a distinguished vertex $\wp \in V$ such that $\deg_{\mathrm{out}}(v) = 2$ for all $v \in V \setminus \{\wp\}$, and $\deg_{\mathrm{out}}(\wp) = 0$. This vertex $\wp$ is called *the genesis*.
- Any $v \in V$ such that $v \neq \wp$ references $\wp$; that is, there is an oriented path[7] from v to $\wp$.

We now describe the tangle as a continuous-time Markov process on the space $\mathcal{G}$. The state of the tangle at time $t \geqslant 0$ is a DAG $\mathcal{T}(t) = (V_{\mathcal{T}}(t), E_{\mathcal{T}}(t))$, where $V_{\mathcal{T}}(t)$ is the set of vertices and $E_{\mathcal{T}}(t)$ is the multiset of directed edges at time t. The process's dynamics are described in the following way:

- The initial state of the process is defined by $V_{\mathcal{T}}(0) = \wp$, $E_{\mathcal{T}}(0) = \varnothing$.
- The tangle *grows with time*, that is, $V_{\mathcal{T}}(t_1) \subset V_{\mathcal{T}}(t_2)$ and $E_{\mathcal{T}}(t_1) \subset E_{\mathcal{T}}(t_2)$ whenever $0 \leqslant t_1 < t_2$.
- For a fixed parameter $\lambda > 0$, there is a Poisson process of incoming *transactions*; these transactions then become the vertices of the

[5] i.e., want to issue valid transactions and have them confirmed by the rest of the network.

[6] One can think of vertices as transactions.

[7] Not necessarily unique.

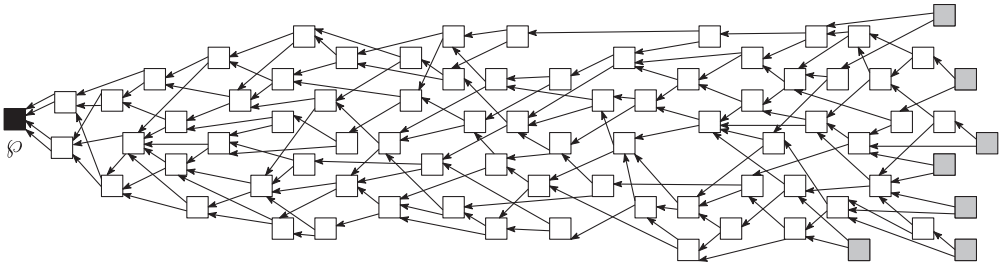

Fig. 1. On the DAGs we are considering: the genesis vertex is on the left, and the tips are grey.

tangle.

- Each incoming transaction chooses[8] two vertices v' and v'' (which, in general, may coincide), and we add the edges (v, v') and (v, v'). We say in this case that this new transaction was *attached* to v' and v'' (equivalently, v *approves* v' and v'').
- Specifically, if a new transaction v arrived at time t', then $V_{\mathcal{T}}(t' +) = V_{\mathcal{T}}(t') \cup \{v\}$, and $E_{\mathcal{T}}(t' +) = E_{\mathcal{T}}(t) \cup \{(v, v'), (v, v'')\}$.

Let us write

$$\mathcal{P}^{(t)}(x) = \{y \in \mathcal{T}(t)\colon y \text{ is referenced by } x\},$$
$$\mathcal{F}^{(t)}(x) = \{z \in \mathcal{T}(t)\colon z \text{ references } x\}$$

for the "past" and the "future" with respect to x (at time t). Note that these introduce a *partial order* structure on the tangle. Observe that, if t_0 is the time moment when x was attached to the tangle, then $\mathcal{P}^{(t)}(x) = \mathcal{P}^{(t_0)}(x)$ for all $t \geqslant t_0$. We also define the *cumulative weight* $\mathcal{H}_x^{(t)}$ of the vertex x at time t by

$$\mathcal{H}_x^{(t)} = 1 + \operatorname{card}(\mathcal{F}^{(t)}(x)); \tag{1}$$

that is, the cumulative weight of x is one[9] plus the number of vertices that reference it. Observe that, for any $t > 0$, if y approves x then $\mathcal{H}_x^{(t)} - \mathcal{H}_y^{(t)} \geqslant 1$, and the inequality is strict if and only if there are vertices different from y which also approve x. Also note that the cumulative weight of any tip is equal to 1.

There is some data associated to each vertex (transaction), created at the moment when that transaction was attached to the tangle. The precise nature of that data is not relevant for the purposes of this paper, so we assume that it is an element of some (unspecified, but finite) set $\mathcal{D}$; what is important, however, is that there is a natural way to say if the set of vertices is *consistent* with respect to the data they contain.[10] When it is necessary to emphasize that the vertices of $G \in \mathcal{G}$ contain some data, we consider the *marked* DAG $G^{[\mathfrak{d}]}$ to be $(G, \mathfrak{d}) = (V, E, \mathfrak{d})$, where $\mathfrak{d}$ is a function $V \to \mathcal{D}$. We define $\mathcal{G}^{[\mathfrak{d}]}$ to be the set of all marked DAGs $(G, \mathfrak{d})$, where $G \in \mathcal{G}$.

A note on terminology: we reserve the term "node" for entities that participate in the system by issuing transactions (which are, by their turn, *vertices* of the tangle graph). That is, the "players" mentioned in Section 1 are nodes.

2.1. On attaching a new transaction to the Tangle

There is one very important detail that has not been explained, namely: how does a newly arrived transaction choose which two vertices in the tangle it will approve, i.e., what is the *attachment strategy*? Notice that, in principle, it would be good[11] for the whole system if the new transactions always prefer to select tips as attachment places, since this way more transactions would be "confirmed".[12] In any case, it is quite clear that the appropriate choice of the attachment strategy is essential for the correct functioning of the system, whatever this could mean.

It is also important to comment that the attachment strategy of a network node is something "internal" to it; what others can see, are the *attachment choices* of the node, but the mechanism behind them need not be publicly known. For this reason, an attachment strategy cannot be *imposed* in the protocol.

We now describe a possible choice of the attachment strategy, used to determine where the incoming transaction will be attached. It is also known as the *recommended tip selection algorithm* (Popov, 2015), since, due to reasons described above, the recommended nodes' behavior is always to try to approve tips. We stress again, however, that approving only tips is not imposed in the protocol, since there is usually no way to know if a node "knew" if the transaction it approved was already approved by someone else before (also, there is no way to know which approving transaction was the first).Let us denote by $\mathcal{L}(t)$ the set of all vertices that are tips at time t, and let $L(t) = \operatorname{card}(\mathcal{L}(t))$. To model the network propagation delays, we introduce a parameter $h > 0$, and assume that at time t only $\mathcal{T}(t - h)$ is known to the entity that issued the incoming transaction. We then define the *tip-selecting random walk*, in the following way. It depends on a parameter q (the backtracking probability) and on a function f. The initial state of the random walk is the genesis $\wp$, [13] and it is stopped upon hitting the set $\mathcal{L}(t - h)$. It is important to observe that $v \in \mathcal{L}(t - h)$ does not necessarily mean that v is still a tip at time t. Let $f\colon \mathbb{R}_+ \to \mathbb{R}_+$ be a monotone non-increasing function. The transition probabilities of the walkers are defined in the following way: the walk *backtracks* (i.e., jumps to a randomly chosen site it approves) with probability $q \in [0, 1/2)$; if y approves $x \neq \wp$, then the transition probability $P_{xy}^{(f)}$ is proportional to $f(\mathcal{H}_x - \mathcal{H}_y)$, that is,

$$P_{xy}^{(f)} = \begin{cases} \dfrac{q}{\operatorname{card}(\mathcal{A}(x))}, & \text{if } y \in \mathcal{A}(x), \\ \dfrac{(1-q) f(\mathcal{H}_x^{(t-h)} - \mathcal{H}_y^{(t-h)})}{\sum\limits_{z : x \in \mathcal{A}(z)} f\left(\mathcal{H}_x^{(t-h)} - \mathcal{H}_z^{(t-h)}\right)}, & \text{if } x \in \mathcal{A}(y), \\ 0, & \text{otherwise;} \end{cases} \tag{2}$$

for $x = \wp$ we define the transition probabilities as above, but with $q = 0$. In words, the walker *backtracks* (i.e., moves one step away from the tips) with (total) probability q, and advances one step towards the tips with (total) probability $(1 - q)$ and relative weights as above. Note that the fact that $q < 1/2$ guarantees that the random walk eventually reaches a tip.[14] almost surely. In what follows, we will mostly assume that $f(s) = \exp(-\alpha s)$ for some $\alpha \geqslant 0$. We use the notation $P^{(\alpha)}$ for the transition probabilities in this case. Intuitively, the smaller is the value

[8] The precise selection mechanism will be described below.

[9] Its "own weight".

[10] One may think that the data refers to value transactions between accounts, and consistency means that no account has negative balance as a result, and/or the total balance has not increased.

[11] Good in the sense described in Section 1.

[12] We discuss the exact meaning of this later; for now, think that "confirmed" means "referenced by many other transactions".

[13] Although in practical implementations one may start it in some place closer to the tips.

[14] More precisely, reaches a vertex that the node *assumes* to be a tip.

of α, the more *random* the walk is.[15] It is worth observing that the case $q = 0$ and $\alpha \to \infty$ corresponds to the GHOST protocol of Sompolinsky and Zohar (2013) (more precisely, to the obvious generalization of the GHOST protocol to the case when a tree is substituted by a DAG).

Now, to select two tips w_1 and w_2 where our transaction will be attached, just run two independent random walks as above, and stop when on the first hit $\mathcal{L}(t-h)$. One can also require that w_1 should be different from w_2; for that, one may re-run the second random walk in the case its exit point happened to be the same as that of the first random walk. Observe that $(\mathcal{T}(t),\, t \geqslant 0)$ is a continuous-time transient Markov process on $\mathcal{G}$; since the state space is quite large, it is difficult to analyse this process. In particular, for a fixed time t, it is not easy to study the above random walk since it takes place on a *random* graph, e.g., can be viewed as a random walk in a random environment; it is common knowledge that random walks in random environments are notoriously hard to deal with.Some motivation for choosing the attachment strategy in the above way is provided in Popov (2015). Very briefly, it encourages the nodes to choose *recent* transactions for approval since a transaction which approved a couple of old transactions, also known as *lazy tip*, is unlikely to be chosen by the above random walk, due to the large difference in cumulative weights in the argument of f in (2)), and also gives protection against certain kinds of attacks (e.g., the double-spending attack).

Let $\gamma_0 \in (0,\, 1)$ be some number, typically close to 1. We say that a transaction is *confirmed with confidence*γ_0 if, with probability at least γ_0, the large-α random walk.[16] ends in a tip which references that transaction. It may happen that a transaction does not get confirmed (even, possible, does not get approved a single time), and becomes orphaned forever. Let us define the event

$$\mathcal{U} = \{\text{every transaction eventually gets approved}\}.$$

We believe that the following statement holds true; however, we have only a heuristical argument in its favor, not a rigorous proof. In any case, it is mostly of theoretical interest, since, as explained below, in practice we will find ourselves in the situation where $\mathbb{P}[\mathcal{U}] = 0$. We therefore state it as.

Conjecture 2.1. It holds that

$$\mathbb{P}[\mathcal{U}] = \begin{cases} 0, & \text{if } \int_0^{+\infty} f(s)ds < \infty, \\ 1, & \text{if } \int_0^{+\infty} f(s)ds = \infty. \end{cases} \tag{3}$$

Explanation. First of all, it should be true that $\mathbb{P}[\mathcal{U}] \in \{0,\, 1\}$ since $\mathcal{U}$ is a *tail event* with respect to the natural filtration; however, it does not seem to be very easy to prove the 0–1 law in this context – recall that we are dealing with a transient Markov process on an infinite state space. Next, consider a tip which got attached to the tangle at time t_0, and assume that it is still a tip at time $t \gg t_0$; also, assume that, among all tips, is "closest", in some suitable sense, to the genesis. Let us now think of the following question: what is the probability that will still be a tip at time $t + 1$?.

Look at Fig. 2: during the time interval $[t,\, t+1)$, $O(1)$ new particles will arrive, and the corresponding walks will travel from the genesis looking for tips. Each of these walks will have to cross the dotted vertical segment on the picture, and with positive probability at least one of them will pass through w_0, one of the vertices approved by . Assume that w_0 was already confirmed, i.e., it is connected to the right end of the tangle via some other transaction u_0 that approves w_0. Then, it is clear (but not easy to prove!) that the cumulative weight of both u_0 and w_0 should be $O(t)$, and so, when in w_0, the walk will jump to the tip with probability $f(O(t))$.

This suggests that the probability that $v_0 \in \mathcal{L}(t+1)$ (i.e., that still is tip at time $t + 1$) is $f(O(t))$, and the Borel-Cantelli lemma[17] gives that the probability that will be eventually approved is less than 1 or equal to 1 depending on whether $\sum_n f(n)$ converges or diverges; the convergence (divergence) of the sum is equivalent to convergence (divergence) of the integral in (3) due to the monotonicity of the function f. A standard probabilistic argument[18] would then imply that if the probability that *a given* tip remains orphaned forever is uniformly positive, then the probability that *at least one* tip remains orphaned forever is equal to 1. □

One may naturally think that it would be better to choose the function f in such a way that, almost surely, every tip eventually gets confirmed. However, as explained in Section 4.1 of Popov (2015), there is a good reason to choose a rapidly decreasing function f, because this defends the system against nodes' misbehavior and attacks. The idea is then to assume that a transaction which did not get confirmed during a sufficiently long period of time is "unlucky", and needs to be re-attached[19] to the tangle. Let us fix some $K > 0$: it stands for the time when an unlucky transaction is reissued (because there is already very little hope that it would be confirmed "naturally"). We call a transaction issued less than K time units ago "unconfirmed", and if a transaction was issued more than K time units ago and was not confirmed, we call it "orphaned". In the following, we assume that the system is *stable*, in the sense that the "recent" unconfirmed transactions do not accumulate and the time until a transaction is confirmed does not depend a lot on the moment when it appeared in the system.[20] We prefer not to elaborate on the exact mathematical definition of stability here, since it requires considering a certain compactification of the space of DAGs (which essentially amounts to considering DAGs with "genesis at minus infinity"), but, hopefully, the idea is intuitively clear anyway.

In that stable regime, let p be the probability that a transaction is confirmed K time units after it was issued for the first time; the number of times a transaction should be issued to achieve confirmation is then a Geometric random variable with parameter p (and, therefore, with expected value p^{-1}); so, the mean time until the transaction is confirmed is K/p. Let us then recall the following remarkable fact belonging to the queuing theory, known as the Little's formula (sometimes also referred to as the Little's theorem or the Little's identity):

Proposition 2.2. Suppose that λ_a is the arrival rate, μ is the mean number of customers in the system, and T is the mean time a customer spends in the system. Then $T = \mu/\lambda_a$.

Proof. See e.g. Section 5.2 of Cooper (1981). To understand intuitively why this fact holds true, one may reason in the following way: assume that, while in the system, each customer pays money to the system with rate 1. Then, at large time t, the total amount of money earned by the system would be (approximately) μt on one hand, and $T\lambda_a t$ on the other hand. Dividing by t and then sending t to infinity, we obtain $\mu = T\lambda_a$.

Little's formula then implies[21] the following (recall that λ is the rate of the incoming transactions flow, not counting reattachments).

Proposition 2.3. The average number of unconfirmed transactions[22] in the system is equal to $p^{-1}\lambda K$.

[15] Physicists would call the case of small α*high temperature regime*, and the case of large α*low temperature regime* (that is, α stands for the inverse temperature).

[16] Recall that the large-α random walk is "more deterministic".

[17] To be precise, a bit more refined argument is needed since the corresponding events are not independent.

[18] Which is also not so easy to formalize in these circumstances.

[19] In fact, the nodes of the network may adopt a rule that instructs to delete the transactions that are older than K and still are tips from their databases.

[20] Simulations indicate that this is indeed the case when α is small (cf. Kuśmierz and Gal, 2018); however, it is not guaranteed to happen for large values of α.

[21] In the language of queuing systems, a reissued transaction is a customer which goes back to the server after an unsuccessful service attempt.

[22] We regard all reattachments as a single trasaction, and if one of the reattachments is confirmed, the transaction is considered confirmed.

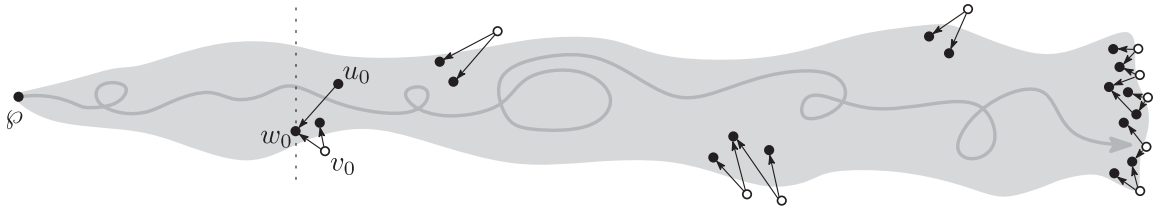

Fig. 2. The walk on the tangle and tip selection. Tips are circles, and transactions which were approved at least once are disks.

Proof. Indeed, apply Proposition 2.2 with $\lambda_a = \lambda$ (think of a transaction which was reattached as a customer which returns to the server after an insuccessful service attempt; this way, the incoming flow of customers still has rate λ). As observed before, the mean time spent by a customer in the system is equal to K/p.

When the tangle contains data, this, in principle, can make transactions incompatible between each other. In this case one may choose more sophisticated methods of tip selection. As we already mentioned,[23] selecting tips with larger values of α provides better defense against attacks and misbehavior; however, smaller values of α make the system more stable with respect to the transactions' confirmation times. An example of "mixed-α" strategy is the following. Define the "model tip" w_0 as a result of the random walk with large α, then select two tips w_1 and w_2 with random walks with small α, but check that

$$\mathcal{P}^{(t-h)}(w_0) \cup \mathcal{P}^{(t-h)}(w_1) \cup \mathcal{P}^{(t-h)}(w_2)$$

is consistent.

3. Selfish nodes and Nash equilibria

Now, we are going to study the situation when some participants of the network are "selfish" and want to use a customized attachment strategy, in order to improve the confirmation time of their transactions (possibly at the expense of the others).

For a finite set A let us denote by $\mathcal{M}(A)$ the set of all probability measures on A, that is

$$\mathcal{M}(A) = \left\{\mu\colon A \to \mathbb{R} \text{ such that } \mu(a) \geqslant 0 \text{forall } a \in A \text{ and } \sum_{a\in A} \mu(a) = 1\right\}$$

.

Let

$$\mathfrak{M} = \bigcup_{G=(V,E)\in\mathcal{G}} \mathcal{M}\left(V \times V\right)$$

be the union of the sets of all probability measures on the pairs of (not necessarily distinct) vertices of DAGs belonging to $\mathcal{G}$. Then, a *general mixed attachment strategy* $\mathcal{S}$ is a map

$$\mathcal{S}\colon \mathcal{G}^{[\mathfrak{d}]} \to \mathfrak{M} \tag{4}$$

with the property $\mathcal{S}(V, E, \mathfrak{d}) \in \mathcal{M}(V \times V)$ for any $G^{[\mathfrak{d}]} = (V, E, \mathfrak{d}) \in \mathcal{G}^{[\mathfrak{d}]}$; that is, for any $G \in \mathcal{G}$ with data attached to the vertices (which corresponds to the state of the tangle at a given time) there is a corresponding probability measure on the set of pairs of the vertices. Note also that in the above we considered *ordered* pairs of vertices, which, of course, does not restrict the generality.

Let $\varkappa > 0$ be a fixed number. We now assume that, for a large N, there are $\varkappa N$ nodes that follow the default tip selection algorithm, and N "selfish" nodes that try to minimize their "cost", whatever it could mean.[24] Assume that all nodes issue transactions with the same rate $\frac{\lambda}{(\varkappa+1)N}$, independently. The overall rate of "honest" transactions in the system is then equal to $\frac{\lambda\varkappa}{\varkappa+1}$, and the overall rate of transactions issued by selfish nodes equals $\frac{\lambda}{\varkappa+1}$. We also justify the assumption that the number of selfish nodes is large by observing that.

- a small number of nodes that do not want to disrupt the system but just want to obtain some advantages for themselves (like e.g. faster confirmations times) are unlikely to "globally" influence the system in any considerable way, even if they do obtain those advantages for themselves;
- however, when it becomes known that it is possible to obtain advantages by deviating from the "recommended" behavior, it is reasonable to expect that a large number of independent entities would try to do it.

3.1. Some further assumptions and definitions

Let us now recall that, in practice, the nodes are computers running a specialized software, so they are selecting the places to attach their transactions in some algorithmic way, using limited physical resources. In such situation, it is unrealistic to assume that a general strategy as in (4) could be implemented "directly", since the space $\mathcal{G}^{[\mathfrak{d}]}$ is infinite; for the same reason, even working with *simple* attachment strategies (which are maps that take an element of $\mathcal{G}^{[\mathfrak{d}]}$ as an input and produce a *deterministic* pair of its vertices as an output) is unrealistic.

Therefore, it looks like a good idea to *restrict* the strategy space we are working with. First, we consider the following simplifying assumption (which is, by the way, also quite reasonable, since, in practice, one would hardly use the genesis as the starting vertex for the random walks due to runtime issues):

Assumption L. There is $n_1 > 0$ such that the attachment strategies of all nodes (including those that use the default attachment strategy) only depend on the restriction of the tangle to the last n_1 transactions that they see.

Observe that, under the above assumption, the set of all such strategies can be thought of as a compact convex subset of $\mathbb{R}^d$, where $d = d(n_1)$ is sufficiently large.

In this section we use a different approach to model the network propagation delays: instead of assuming that an incoming transaction does not have information about the state of the tangle during last h units of time, we rather assume that it does not have information about the last n_0 transactions attached to the tangle, where $n_0 < n_1$ is some fixed positive number (so, effectively, the strategies would depend on subgraphs induced by $n_1 - n_0$ transactions, although the results of this section do not rely on this assumption). Clearly, these two approaches are quite similar in spirit; however, the second one permits us to avoid certain technical difficulties related to randomness of the number of

[23] Recall the discussion around $f(s) = \exp(-\alpha s)$ right after (2).

[24] For example, the cost may be the expected confirmation time of a transaction (conditioned that it is eventually confirmed), the probability that it was not approved during certain (fixed) time interval, etc.; below in (6) we provide the exact definition of the cost function we are working with in this paper.

unseen transactions in the first case. Also, it will be more natural and convenient to pass from continuous to discrete time.

Now, even with the restrictions as above, it is still unrealistic to work with the simple strategies of the sort "choose a fixed pair of transactions for each possible restriction of the tangle to the set of last n_1 transactions", because implementing it in practive would require effectively dealing with sets indexed by all possible restrictions, and the size of the latter set clearly grows exponentially in n_1. Instead, as hinted in the beginning of this subsection, we think of different "attachment methods" as simple strategies. Formally, let $\mathcal{G}_{n_1}^{[\mathfrak{d}]}$ be the set of all possible sub-DAGs of $\mathcal{G}^{[\mathfrak{d}]}$ with n_1 vertices, and $\mathfrak{M}_{n_1}$ be the set of all probability measures on the vertices' pairs of elements of $\mathcal{G}_{n_1}^{[\mathfrak{d}]}$. Clearly, the set $\mathcal{G}_{n_1}^{[\mathfrak{d}]}$ is finite. An *attachment method* is then a map

$$\mathcal{J}\colon \mathcal{G}_{n_1}^{[\mathfrak{d}]} \to \mathfrak{M}_{n_1};$$

it is thought of as a (randomized) polynomial-time polynomial-memory algorithm which takes the last n_1 transactions and returns a pair of those transactions which would serve as attachment's locations. Then, the available simple strategies are attachment methods

$$\{\mathcal{J}_\beta, \beta \in \mathfrak{A}\},$$

where $\mathfrak{A}$ is some (unspecified) index set. It is also important to observe that this approach does not restricts generality. We then denote by Q the set of all *mixed* strategies of the form $\mathcal{J}_\Xi$, where Ξ is a random variable on $\mathfrak{A}$. Observe also that the set of simple strategies can be thought of as a subset of $\mathbb{R}^d$ (which we assume also to be compact), where $d = d(n_1)$ is sufficiently large, and Q would be then its convex hull.

Let $\mathcal{S}_1, \ldots, \mathcal{S}_N \in Q$ be the attachment strategies used by the selfish nodes. To evaluate the "goodness" of a strategy, one has to choose and then optimize some suitable observable (that stands for the "cost"); as usual, there are several "reasonable" ways to do this. We decided to choose the following one, for definiteness and also for technical reasons (to guarantee the continuity of a certain function used below); one can probably extend our arguments to other reasonable cost functions. Assume that a transaction v was attached to the tangle at time t_v, so $v \in V_{\mathcal{T}}(t)$ for all $t \geqslant t_v$. Fix some (typically large) $M_0 \in \mathbb{N}$. Let $t_1^{(v)}, \ldots, t_{M_0}^{(v)}$ be the moments when the subsequent M_0 (after v) transactions were attached to the tangle. For $k = 1\ldots,M_0$ let $R_k^{(v)}$ be the event that the *default* tip-selecting walk[25] on $\mathcal{T}(t_k^{(v)})$ stops in a tip that *does not* reference v. We then define the random variable

$$W(v) = \mathbf{1}_{R_1^{(v)}} + \cdots + \mathbf{1}_{R_{M_0}^{(v)}} \tag{5}$$

to be the number of times that the M_0 "subsequent" tip selection random walks do not reference v (in the above, $\mathbf{1}_A$ is the indicator function of an event A). Intuitively, the smaller is the value of $W(v)/M_0$, the bigger is the chance that v is quickly confirmed.

Next, assume that $(v_j^{(k)}, j \geqslant 1)$ are the transactions issued by the kth (selfish) node. We define

$$\mathfrak{C}^{(k)}\left(\mathcal{S}_1, \ldots, \mathcal{S}_N\right) = M_0^{-1} \lim_{n\to\infty} \frac{W(v_1^{(k)}) + \cdots + W(v_n^{(k)})}{n}, \tag{6}$$

to be the *mean cost* of the kth node given that $\mathcal{S}_1, \ldots, \mathcal{S}_N$ are the attachment strategies of the selfish nodes.

Definition 3.1. We say that a set of strategies $(\mathcal{S}_1, \ldots, \mathcal{S}_N) \in Q^N$ is a *Nash equilibrium* if

$$\mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}, \mathcal{S}_k, \mathcal{S}_{k+1}, \ldots, \mathcal{S}_N) \leqslant \mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}, \mathcal{S}', \mathcal{S}_{k+1}, \ldots, \mathcal{S}_N)$$

for any k and any $\mathcal{S}' \in Q$.

Observe that, since the nodes are indistinguishable, the fact that $(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ is a Nash equilibrium implies that so is $(\mathcal{S}_{\sigma_1}, \ldots, \mathcal{S}_{\sigma(N)})$ for any permutation σ.

[25] i.e., the one used by nodes following the default attachment strategy.

3.2. Main results

From now on, we assume that vertices contain no data, i.e., the set $\mathcal{D}$ is empty; this is not absolutely necessary because, with the data, the proof will be essentially the same; however, the notations would become much more cumbersome. Also, there will be no reattachments; again, this would unnecessarily complicate the proofs (one would have to work with *decorated* Poisson processes). In fact, we are dealing with a so-called *random-turn game* here, see e.g. Chapter 9 of Karlin and Peres (2017) for other examples.

Consider, for the moment, the situation when all nodes use the same attachment strategy (i.e., there are no selfish nodes). The restriction of the tangle on the last n_1 transactions then becomes a Markov chain on the state space $\mathcal{G}_{n_1}$. We now make the following technical assumption on that Markov chain:

Assumption D. The above Markov chain is irreducible and aperiodic.

It is important to observe that Assumption D is *not* guaranteed to hold for *every* natural attachment strategy; however, still, this is not a very restrictive assumption in practice because every finite Markov chain may be turned into an irreducible and aperiodic one by an arbitrarily small perturbation of the transition matrix.

Then, we are able to prove the following.

Theorem 3.2. Under Assumptions L and D, the system has at least one Nash equilibrium.

Symmetric games do not always have symmetric Nash equilibria, as shown in Fey (2012). Also, even when such equilibria exist in the class of mixed strategies, they may be "inferior" to asymmetric pure equilibria; for example, this happens in the classical "Battle of the sexes" game (see e.g. Section 7.2 of Karlin and Peres, 2017).

Now, the goal is to prove that, if the number of selfish nodes N is large, then for *any* equilibrium state the costs of distinct nodes cannot be significantly different. Let us recall the notations we use: $\mathcal{S}_1, \ldots, \mathcal{S}_N$ are the strategies of the N selfish nodes, and $\mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_N)$, $k = 1, \ldots, N$, are the mean costs of the selfish nodes, defined in (6). Now, we have the following.

Theorem 3.3. For any $\varepsilon > 0$ there exists N_0 (depending on the default attachment strategy) such that, for all $N \geqslant N_0$ and any Nash equilibrium $(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ it holds that

$$|\mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_N) - \mathfrak{C}^{(j)}(\mathcal{S}_1, \ldots, \mathcal{S}_N)| < \varepsilon \tag{7}$$

for all $k, j \in \{1, \ldots, N\}$.

Now, let us define the notion of *approximate* Nash equilibrium:

Definition 3.4. For a fixed $\varepsilon > 0$, we say that a set of strategies $(\mathcal{S}_1, \ldots, \mathcal{S}_N) \in Q^N$ is an ε-equilibrium if

$$\mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}, \mathcal{S}_k, \mathcal{S}_{k+1}, \ldots, \mathcal{S}_N) \leqslant \mathfrak{C}^{(k)}(\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}, \mathcal{S}', \mathcal{S}_{k+1}, \ldots, \mathcal{S}_N) + \varepsilon$$

for any k and any $\mathcal{S}' \in Q$.

The motivation for introducing this notion is that, if ε is very small, then, in practice, ε-equilibria are essentially indistinguishable from the "true" Nash equilibria.

Theorem 3.5. For any $\varepsilon > 0$ there exists N_0 (depending on the default attachment strategy) such that, for all $N \geqslant N_0$ and any Nash equilibrium $(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ it holds that $(\mathcal{S}, \ldots, \mathcal{S})$ is an ε-equilibrium, where

$$\mathcal{S} = \frac{1}{N} \sum_{k=1}^{N} \mathcal{S}^{(k)} \tag{8}$$

(that is, all selfish nodes use the same "averaged" strategy defined above). The costs of all selfish nodes are then equal to

$$\frac{1}{N}\sum_{k=1}^{N}\mathfrak{C}^{(k)}\Big(\mathcal{S}_1,\ ...,\mathcal{S}_N\Big),$$

that is, the average cost in the Nash equilibrium.

In other words, for large N one can essentially assume that all selfish nodes follow the same attachment strategy. This result will be important in Section 4, because it makes it possible to use (practical) simulations in order to find the Nash equilibria of systems with large number of selfish players.

3.3. Proofs

First, we need the following technical result:

Lemma 3.6. Let P be the transition matrix of an irreducible and aperiodic discrete-time Markov chain on a finite state space E. Let $\widehat{P}$ be a continuous map from a compact set $F \subset \mathbb{R}^d$ to the set of all stochastic matrices on E (equipped by the distance inherited from the usual matrix norm on the space of all matrices on E). Fix $\theta \in (0, 1)$, denote $\widetilde{P}(s) = \theta P + (1-\theta)\widehat{P}(s)$, and let π_s be the (unique) stationary measure of $\widetilde{P}(s)$. Then π_s is also continuous (as a function of s).

Proof. In the following we give a (rather) probabilistic proof of this fact via the Kac's lemma, although, of course, a purely analytic proof is also possible. Irreducibility and aperiodicity of P imply that, for some $m_0 \in \mathbb{N}$ and $\varepsilon_0 > 0$

$$P_{xy}^{m_0} \geqslant \varepsilon_0 \tag{9}$$

for all $x, y \in E$, where $P^{m_0} = (P_{xy}^{m_0},\ x, y \in E)$ is the transition matrix in m_0 steps. Now, (9) implies that

$$\widetilde{P}_{xy}^{m_0}(s) \geqslant \theta^{m_0}\varepsilon_0 \tag{10}$$

for all $x, y \in E$ and all $s \in F$.

Being $(X_n,\ n \geqslant 0)$ a stochastic process on E, let us define

$$\tau(x) = \min\{k \geqslant 1\colon X_k = x\}$$

(with the convention $\min\varnothing = \infty$) to be the *hitting time* of the site $x \in E$ by the stochastic process X. Now, let $\mathbb{P}_x^{(s)}$ and $\mathbb{E}_x^{(s)}$ be the probability and the expectation with respect to the Markov chain with transition matrix $\widetilde{P}(s)$ starting from $x \in E$. We now recall the Kac's lemma (cf. e.g. Theorem 1.22 of Durrett, 2012): for all $x \in E$ it holds that

$$\pi_s(x) = \frac{1}{\mathbb{E}_x^{(s)}\tau(x)}. \tag{11}$$

Now, (10) readily implies that, for all $x \in E$ and $n \in \mathbb{N}$,

$$\mathbb{P}_x^{(s)}[\tau(x) \geqslant n] \leqslant c_1 e^{-c_2 n} \tag{12}$$

for some positive constants $c_{1,2}$ which do not depend on s. This in its turn implies that the series

$$\mathbb{E}_x^{(s)}\tau(x) = \sum_{n=1}^{\infty}\mathbb{P}_x^{(s)}\Big[\tau(x) \geqslant n\Big]$$

converges uniformly in s and so $\mathbb{E}_x^{(s)}\tau(x)$ is uniformly bounded from above[26]; also, the Uniform Limit Theorem (see e.g. Section D.6.2 of Ok, 2007) implies that $\mathbb{E}_x^{(s)}\tau(x)$ is continuous in s. Therefore, for any $x \in E$, (11) implies that $\pi_s(x)$ is also a continuous function of s.

Proof of Theorem 3.2. The authors were unable to find a result available in the literature that implies Theorem 3.2 directly; nevertheless, its proof is quite standard and essentially follows Nash's original paper (Nash, 1950) (see also Fink, 1964). There is only one technical difficulty, which we intend to address via the above preparatory steps: one needs to prove the continuity of the cost function.

[26] and, of course, it is also bounded from below by 1.

Denote by π_S the invariant measure of the Markov chain given that the (selfish) nodes use the "strategy vector" $\mathbf{s} = (\mathcal{S}_1,\ ...,\mathcal{S}_N)$. Then, the idea is to use Lemma 3.6 with $\theta = \frac{\kappa}{\kappa+1}$, P the transition matrix obtained from the default attachment strategy, and $\widehat{P}(s)$ is the transition matrix obtained from the strategy $\mathcal{S}' = N^{-1}\sum_{k=1}^{N}\mathcal{S}_k$ (observe that N nodes using the strategies $\mathcal{S}_1,\ ...,\mathcal{S}_N$, is the same as one node with strategy $\mathcal{S}'$ issuing transactions N times faster). Assumption D together with Lemma 3.6 then imply that $\pi_{\mathbf{s}} := \pi_{\mathcal{S}'}$ is a continuous function of $\mathbf{s}$.

Let $\mathbb{E}_{\pi_{\mathcal{S}'}}^{\mathcal{S},\widehat{\mathcal{S}}}$ be the expectation with respect to the following procedure: take the "starting" graph according to $\pi_{\mathcal{S}'}$, then attach to it a transaction according to the strategy $\mathcal{S}$, and then keep attaching subsequent transactions according to the strategy $\widehat{\mathcal{S}}$ (instead of $\mathcal{S}'$ and $\widehat{\mathcal{S}}$ we may also use the strategy vectors; $\mathcal{S}'$ and $\widehat{\mathcal{S}}$ would be then their averages). Let also $W^{(k)}$ be the random variable defined as in (5) for an arbitrary transaction v issued by the kth node. Then, the Ergodic Theorem for Markov chains (see e.g. Theorem 1.23 of Durrett, 2012) implies that

$$\mathfrak{C}^{(k)}(\mathcal{S}) = \mathbb{E}_{\pi_{\mathcal{S}'}}^{\mathcal{S}_k,\mathcal{S}'}W^{(k)}. \tag{13}$$

It is not difficult to see that the above expression is a polynomial of the $\mathcal{S}$'s coefficients (i.e., the corresponding probabilities) and $\pi_{\mathcal{S}'}$-values, and hence it is a continuous function on the space of strategies $\mathfrak{M}_{n_1}$. Using this, the rest of the proof is standard, it is obtained as a consequence of the Kakutani's fixed point theorem (Kakutani, 1941), also with the help of the Berge's Maximum Theorem (see e.g. Chapter E.3 of Ok, 2007).

Proof of Theorem 3.3. Without restricting generality we may assume that

$$\mathfrak{C}^{(1)}\Big(\mathcal{S}_1,...,\mathcal{S}_N\Big) = \max_{k=1,\ ...,\ N}\mathfrak{C}^{(k)}\Big(\mathcal{S}_1,...,\mathcal{S}_N\Big),$$

$$\mathfrak{C}^{(2)}\Big(\mathcal{S}_1,...,\mathcal{S}_N\Big) = \min_{k=1,\ ...,\ N}\mathfrak{C}^{(k)}\Big(\mathcal{S}_1,...,\mathcal{S}_N\Big),$$

so we then need to proof that $\mathfrak{C}^{(1)}(\mathbf{s}) - \mathfrak{C}^{(2)}(\mathbf{s}) < \varepsilon$, where $\mathbf{s} = (\mathcal{S}_1,\ ...,\mathcal{S}_N)$. Now, the main idea of the proof is the following: if $\mathfrak{C}^{(1)}(\mathbf{s})$ is considerably larger than $\mathfrak{C}^{(2)}(\mathbf{s})$, then the owner of the first node may decide to adopt the strategy used by the second one. This would not necessarily decrease his costs to the former costs of the second node since a change in an individual strategy leads to changes in *all* costs; however, when N is large, the effects of changing the strategy of only one node would be small, and (if the difference of $\mathfrak{C}^{(1)}(\mathbf{s})$ and $\mathfrak{C}^{(2)}(\mathbf{s})$ were not small) this would lead to a contradiction to the assumption that $\mathbf{s}$ was a Nash equilibrium.

So, let us denote $\mathbf{s}' = (\mathcal{S}_2,\ \mathcal{S}_2,\ \mathcal{S}_3,\ ...,\mathcal{S}_N)$, the strategy vector after the first node adopted the strategy of its "more successful" colleague, see Fig. 3. Let

$$\mathcal{S} = \frac{1}{N}(\mathcal{S}_1+\cdots+\mathcal{S}_N) \text{ and } \mathcal{S}' = \frac{1}{N}(2\mathcal{S}_2 + \mathcal{S}_3+\cdots+\mathcal{S}_N)$$

be the two "averaged" strategies. In the following, we are going to compare $\mathfrak{C}^{(2)}(\mathbf{s}) = \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)}$ (the "old" cost of the second node) with $\mathfrak{C}^{(1)}(\mathbf{s}') = \mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(1)}$ (the "new" cost of the first node, after it adopted the second node's strategy). We need the following

Lemma 3.7. For any measure π on $\mathcal{G}_{n_1}$ and any strategy vectors $\mathbf{s} = (\mathcal{S}_1,\ ...,\mathcal{S}_N)$ and $\mathbf{s}' = (\mathcal{S}_1',\ ...,\mathcal{S}_N')$ such that $\mathcal{S}_k = \mathcal{S}_k'$ for all $k = 2,\ ...,N$, we have

$$|\mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}W^{(j)} - \mathbb{E}_{\pi}^{(\mathcal{S}_j',\mathcal{S}')}W^{(j)}| \leqslant \frac{M_0}{N} \tag{14}$$

for all $j = 2,\ ...,N$.

Proof. Let us define the event

$$A = \{\text{among the } M_0 \text{ transactions there is at least one issued by the first node}\},$$

and observe that, by the union bound, the probability that it occurs is at most M_0/N. Then, using the fact that $\mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}(W^{(j)}\mathbf{1}_{A^c}) = \mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S}')}(W^{(j)}\mathbf{1}_{A^c})$

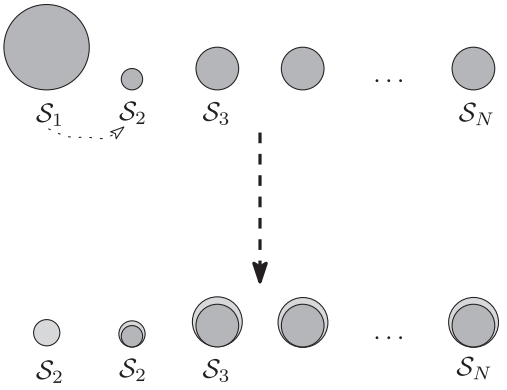

Fig. 3. On the main idea of the proof of Theorem 3.3. The node with the highest cost will switch to the strategy of the node with the lowest cost. That will not guarantee exactly that same cost to the former node, but the difference will be rather small since N is large (so the change in one component of the strategy vector will not influence a lot the outcome).

(since, on A^c, the first node does not "contribute" to $W^{(j)}$), write

$$\begin{aligned}&|\mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}W^{(j)} - \mathbb{E}_{\pi}^{\left(\mathcal{S}'_j,\mathcal{S}'\right)}W^{(j)}|\\ &= |\mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}(W^{(j)}\mathbf{1}_A) + \mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}(W^{(j)}\mathbf{1}_{A^c}) - \mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S}')}(W^{(j)}\mathbf{1}_A) - \mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S}')}(W^{(j)}\mathbf{1}_{A^c})|\\ &= |\mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S})}(W^{(j)}\mathbf{1}_A) - \mathbb{E}_{\pi}^{(\mathcal{S}_j,\mathcal{S}')}(W^{(j)}\mathbf{1}_A)|\\ &\leqslant \tfrac{M_0}{N},\end{aligned}$$

where we also used that $W^{(j)} \leqslant 1$. This concludes the proof of Lemma 3.7.

We continue proving Theorem 3.3. First, by symmetry, we have

$$\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(1)} = \mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(2)}. \tag{15}$$

Also, it holds that

$$|\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(2)} - \mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)}| \leqslant \frac{M_0}{N} \tag{16}$$

by Lemma 3.7. Then, similarly to the proof of Theorem 3.2, we can obtain that the function

$$(\mathcal{S}, \mathcal{S}', \mathcal{S}'') \mapsto \mathbb{E}_{\pi_{\mathcal{S}''}}^{(\mathcal{S},\mathcal{S}')}W^{(2)}$$

is continuous; since it is defined on a compact, it is also uniformly continuous. That is, for any $\varepsilon' > 0$ there exist $\delta' > 0$ such that if $\|(\mathcal{S}, \mathcal{S}', \mathcal{S}'') - (\widetilde{\mathcal{S}}, \widetilde{\mathcal{S}}', \widetilde{\mathcal{S}}'')\| < \delta'$, then

$$|\mathbb{E}_{\pi_{\mathcal{S}''}}^{(\mathcal{S},\mathcal{S}')}W^{(2)} - \mathbb{E}_{\pi_{\widetilde{\mathcal{S}}''}}^{(\widetilde{\mathcal{S}},\widetilde{\mathcal{S}}')}W^{(2)}| < \varepsilon'.$$

Choose $N_0 = \lceil 1/\delta' \rceil$. We then obtain from the above that

$$|\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)} - \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)}| < \varepsilon'. \tag{17}$$

The relations (15)–(17) imply that

$$|\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(1)} - \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)}| \leqslant \varepsilon' + \frac{M_0}{N}.$$

On the other hand, since we assumed that $\mathbf{s}$ is a Nash equilibrium, it holds that

$$\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\mathcal{S}_2,\mathcal{S}')}W^{(1)} = \mathfrak{C}^{(1)}(\mathbf{s}') \geqslant \mathfrak{C}^{(1)}(\mathbf{s}) = \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_1,\mathcal{S})}W^{(1)}, \tag{18}$$

which implies that

$$\mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_1,\mathcal{S})}W^{(1)} - \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_2,\mathcal{S})}W^{(2)} \leqslant \varepsilon' + \frac{M_0}{N}.$$

This concludes the proof of Theorem 3.3.

Proof of Theorem 3.5. To begin, we observe that the proof of the second part is immediate, since, as already noted before, for an external observer, the situation where there are N nodes with strategies $(\mathcal{S}_1, ..., \mathcal{S}_N)$ is indistinguishable from the situation with one node with averaged strategy.

Now, we need to prove that, for any fixed $\varepsilon' > 0$ it holds that

$$\mathfrak{C}^{(1)}(\mathcal{S}, ..., \mathcal{S}) \leqslant \mathfrak{C}^{(1)}(\widetilde{\mathcal{S}}, \mathcal{S}, ..., \mathcal{S}) + \varepsilon' \tag{19}$$

for all large enough N (the claim would then follow by symmetry). Recall that we have

$$\mathfrak{C}^{(1)}(\mathcal{S}, ..., \mathcal{S}) = \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S},\mathcal{S})}W^{(1)}, \tag{20}$$

$$\mathfrak{C}^{(1)}(\mathcal{S}_1, ..., \mathcal{S}_N) = \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_1,\mathcal{S})}W^{(1)}, \tag{21}$$

and

$$\mathfrak{C}^{(1)}(\widetilde{\mathcal{S}}, \mathcal{S}, ..., \mathcal{S}) = \mathbb{E}_{\pi_{\mathcal{S}'}}^{(\widetilde{\mathcal{S}},\mathcal{S}')}W^{(1)}, \tag{22}$$

where

$$\mathcal{S}' = \frac{1}{N}(\widetilde{\mathcal{S}} + (N-1)\mathcal{S}) = \frac{1}{N}\left(\widetilde{\mathcal{S}} + \frac{N-1}{N}\left(\mathcal{S}_1 + \cdots + \mathcal{S}_N\right)\right).$$

Now, the second part of this theorem together with Theorem 3.3 imply[27] that, for any fixed $\varepsilon > 0$

$$|\mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S},\mathcal{S})}W^{(1)} - \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_1,\mathcal{S})}W^{(1)}| < \varepsilon \tag{23}$$

for all large enough N.

Next, let us denote

$$\mathcal{S}'' = \frac{1}{N}\left(\widetilde{\mathcal{S}} + \mathcal{S}_2 + \cdots + \mathcal{S}_N\right).$$

Then, again using the uniform continuity argument (as in the proof of Theorem 3.3), we obtain that, for any $\varepsilon'' > 0$

$$|\mathbb{E}_{\pi_{\mathcal{S}'}}^{(\widetilde{\mathcal{S}},\mathcal{S}')}W^{(1)} - \mathbb{E}_{\pi_{\mathcal{S}''}}^{(\widetilde{\mathcal{S}},\mathcal{S}'')}W^{(1)}| < \varepsilon'' \tag{24}$$

for all large enough N. However,

$$\mathbb{E}_{\pi_{\mathcal{S}''}}^{(\widetilde{\mathcal{S}},\mathcal{S}'')}W^{(1)} = \mathfrak{C}^{(1)}(\widetilde{\mathcal{S}}, \mathcal{S}_2, ..., \mathcal{S}_N) \geqslant \mathfrak{C}^{(1)}(\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_N) = \mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S}_1,\mathcal{S})}W^{(1)},$$

since $(\mathcal{S}_1, ..., \mathcal{S}_N)$ is a Nash equilibrium. Then, (23) and (24) imply that

$$|\mathbb{E}_{\pi_{\mathcal{S}}}^{(\mathcal{S},\mathcal{S})}W^{(1)} - \mathbb{E}_{\pi_{\mathcal{S}'}}^{(\widetilde{\mathcal{S}},\mathcal{S}')}W^{(1)}| < \varepsilon + \varepsilon'',$$

and, recalling (20) and (22), we conclude the proof of Theorem 3.5.

4. Simulations

In this section we investigate Nash equilibria between selfish nodes via simulations. As discussed in Section 1, this is motivated by the following important question: since the choice of an attachment strategy is not enforced, there may indeed be nodes which would prefer to "optimise" their strategies in order to decrease the mean confirmation time of their transactions. So, can this lead to a situation where the corresponding Nash equilibrium is "bad for everybody", effectively leading to the system's malfunctioning?.

Due to Theorem 3.5 we may assume that all selfish nodes use the same attachment strategy. Even then, it is probably unfeasible to calculate that strategy exactly; instead, we resort to simulations, which indeed will show that the equilibrium strategy of the selfish nodes will not be much different from the (suitably chosen) default strategy, at least in the (very natural) situation below. But, before doing that, let us explain the intuition behind this fact. Naively, a reasonable strategy for a selfish node would be the following:

(1) Calculate the exit distribution of the tip-selecting random walk.

[27] Note that Theorem 3.3 implies that, when N is large, the nodes already have "almost" the same cost in the Nash equilibrium $(\mathcal{S}_1, ..., \mathcal{S}_N)$.

(2) Find the two tips where this distribution attains its "best"[28] values.
(3) Approve these two tips.

However, this strategy fails when other selfish nodes are present. To understand this, look at Fig. 4: *many* selfish nodes attach their transactions to the two "best" tips. As a result, the "neighborhood" of these two tips becomes "overcrowded": there is so much competition between the transactions issued by the selfish nodes, that the chances of them being approved soon actually decrease.[29]

To illustrate this fact, several simulations have been done. All the results depicted here were generated using (2) as the transition probabilities, with $q = 1/3$, and a network delay of $h = 1$ second. Also, a transaction will be reattached if the two following criteria are met:

(1) the transaction is older than 20 s
(2) the transaction is not referenced by the tip selected by a random walk with $\alpha = \infty$. [30]

This way, we guarantee not only that the unconfirmed transactions will be eventually confirmed, but also that all transactions that were never reattached are referenced by most of the tips. Note that when the reattachment is allowed in the simulations, if a new transaction references an old, already reattached transaction together with its newly reissued counterpart, there will be a double spending. Even though the odds of that are low (since when a transaction is re-emitted, it will be old enough to be almost never chosen by the random walk algorithm), a specific procedure was included in the simulations in order to not allow double spendings.

The average costs were simulated as defined at Eqs. (5) and (6), so a certain value of M_0 had to be chosen. Since the value of $W(v)/\lambda$ is related to the time of approval of v (whenever the transaction is indeed approved before $t_{M_0}^{(v)}$), we want M_0 to be sufficiently large, in order to capture the effect of most of the approvals. Fig. 5 depicts the typical cumulative distribution of the time of the first approval, for several values of α and λ. Note that roughly 95% of the transactions will be approved before $t = 5$s, and almost its totality will be approved before $t = 10$s. For that reason, in both cases ($\lambda = 25$ and $\lambda = 50$), the mean cost was calculated over the transactions attached during a time interval of approximately 10s ($M_0 = 500$ for $\lambda = 50$ and $M_0 = 250$ for $\lambda = 25$), so almost the totality of approvals will be "seen" by the average cost.

4.1. One dimensional Nash equilibria

In this section, we will study the Nash equilibria $(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ of the tangle problem, considering the following strategy space:

$$\{(1-\theta)\mathcal{S}^0 + \theta\mathcal{S}^1,\ \ 0 \leqslant \theta \leqslant 1\}$$

where the simple strategies $\mathcal{S}^0$ and $\mathcal{S}^1$ are the default tip selection strategy and the "greedy" strategy (defined in the beginning of this section) correspondingly; that is, $\mathcal{S}_i = (1-\theta_i)\mathcal{S}^0 + \theta_i\mathcal{S}^1$ where $\theta_i \in [0, 1]$, $i = 1, \ldots, N$. The goal is to find the Nash equilibria relative to the costs defined in the last section (Eqs. (6) and (5)). The selfish nodes will try to optimise their transaction cost with respect to θ_i.

By Theorem 3.5, each Nash equilibrium in this form will be equivalent to another Nash equilibrium with "averaged" strategies, i.e.:

$$\mathcal{S} = \left(1 - \frac{1}{N}\sum_{k=1}^{N}\theta_i\right)\mathcal{S}^0 + \frac{1}{N}\sum_{k=1}^{N}\theta_i\mathcal{S}^1 = \left(1-\theta\right)\mathcal{S}^0 + \theta\mathcal{S}^1 \text{ for each } i = 1, \ldots, N,$$

Now, suppose that we have a fixed fraction γ of selfish nodes, that choose a strategy among the possible $\mathcal{S}$. The non-selfish nodes will not be able to choose their strategy, so they will be restricted, as expected, to $\mathcal{S}^0$. Note that, since they cannot choose their strategy, they will not "play" the game. Since the costs are linear over $\mathcal{S}$, such mixed strategy game will be equivalent[31] to a game where only a fraction $p = \gamma\theta \leqslant \gamma$ of the nodes chooses $\mathcal{S}^1$ over $\mathcal{S}^0$, and the rest of the nodes chooses $\mathcal{S}^0$ over $\mathcal{S}^1$. Note that this equivalence does not contradict the theorems proved in the last sections, that state:

- all the nodes will have the same average costs when the system is at a Nash equilibrium;
- any Nash equilibrium has an equivalent Nash equilibrium with "averaged" strategies, where all the nodes will have the same strategies.

From now on, we will refer (unless stated otherwise) to this second pure strategy game. Fig. 6(a) represents a typical graph of average costs of transactions issued under $\mathcal{S}^0$ and $\mathcal{S}^1$, as a function of the fraction p, for a low α and two different values of λ. As already demonstrated, when in equilibrium, the selfish nodes should issue transactions with the same average costs. That means that the system should reach equilibrium in one of the following states:

(1) some selfish nodes choose $\mathcal{S}^0$ and the rest choose $\mathcal{S}^1$ $(0 < p < \gamma)$, all of them with the same average costs;
(2) all selfish nodes choose $\mathcal{S}^1$ $(p = \gamma)$;
(3) all selfish nodes choose $\mathcal{S}^0$ $(p = 0)$.

If the two curves on the graphs do not intersect, the equilibrium should be clearly at state (2) or (3), depending on which of the average costs is larger. If the two curves on the graphs intercept each other, we will also have the intersection point as a Nash equilibrium candidate. We call $\bar{\mathbf{s}}$ the vector of strategies on equilibrium and $\bar{p}$ the fraction of nodes that will issue transactions under $\mathcal{S}^1$ when the system is in $\bar{\mathbf{s}}$. We define $p^- = \bar{p} - \frac{\gamma}{N}$ and $p^+ = \bar{p} + \frac{\gamma}{N}$, meaning that p^- and p^+ will be deviations from $\bar{p}$, that result from one node switching strategies, from $\mathcal{S}^0$ to $\mathcal{S}^1$ and from $\mathcal{S}^1$ to $\mathcal{S}^0$, respectively. We also define $\bar{\mathbf{s}}^-$ and $\bar{\mathbf{s}}^+$ as strategy vectors related to p^- and p^+. Note on Fig. 7 that this kind of Nash equilibrium candidate may not be a real equilibrium. In the first example (Fig. 7(a)), when the system is at point $\bar{p}$ and a node switches strategies from $\mathcal{S}^0$ to $\mathcal{S}^1$ (moving from $\bar{p}$ to p^+), the cost actually decreases, so $\bar{p}$ cannot be a Nash equilibrium. On the other hand, the second example (Fig. 7(b)) shows a Nash equilibrium at point $\bar{p}$, since deviations to p^- and p^+ will increase costs.

Now, let us re-examine Fig. 6(a). Here, the Nash equilibrium will occur at the point $\bar{p}$, since we have a situation as on Fig. 7(b). That point is easily found at Fig. 6(b), when $\delta = 0$. Note that the Nash equilibrium for a larger λ will be at a smaller θ_0 than the Nash equilibrium for a smaller λ. This was already expected, since, for a larger λh, the tips will be naturally more "overcrowded", so the effect depicted at Fig. 4 will be amplified. Thus, the Nash equilibrium for the higher λh cases must occur with a smaller proportion of transactions issued with the pure strategy $\mathcal{S}^1$.

Let us now again consider the mixed strategy game. In the case when all the nodes are allowed to choose between the two pure strategies ($\mathcal{S}^0$ and $\mathcal{S}^1$), the Nash equilibrium will be indeed at $\theta_0 = \bar{p}$ (as expected, since in this case $\gamma = 1$). If just a fraction $\gamma = p/\theta > \bar{p}$ of the nodes is selfish, then the Nash equilibrium will occur when $\theta_0 = \bar{p}/\gamma$. Now, if $\gamma \leqslant \bar{p}$, the costs of the nodes will not coincide.[32] In that case, the average cost of transactions under $\mathcal{S}^1$ will always be smaller than the

[28] i.e., the maximum and the second-to-maximum.
[29] The "new" best tips are not among them, as shown on Fig. 4 on the right.
[30] Here, when the random walk must choose among n transactions with the same weight, it will choose randomly, with equal probabilities.
[31] This way, we deal with just one variable (p) instead of two (γ and θ) and none of the parameters of the system is lost.
[32] That is the case for the range of studied parameters.

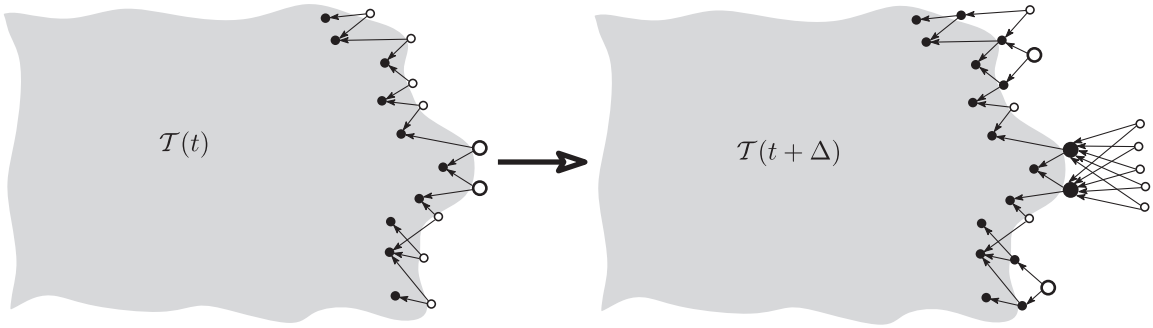

Fig. 4. Why the "greedy" tip selection strategy will not work (the two "best" tips are shown as larger circles).

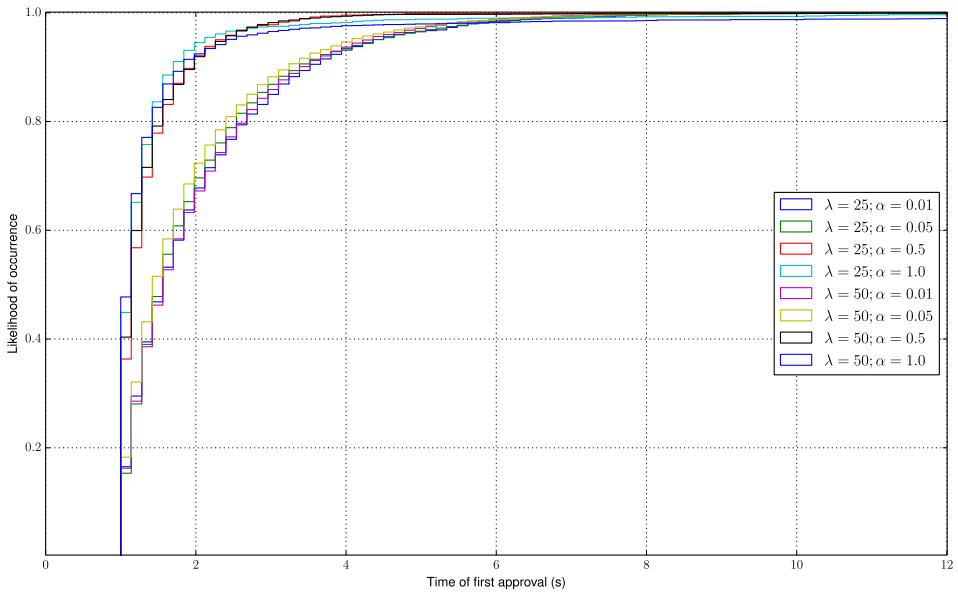

Fig. 5. Cumulative distribution of time of approvals for several values of α and λ.

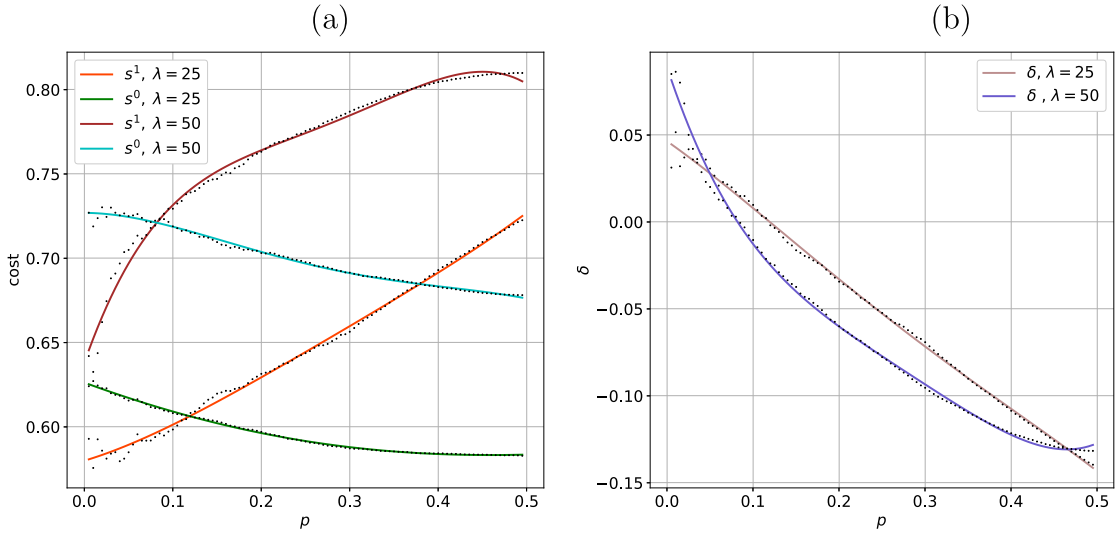

Fig. 6. Dotted lines are the raw data. Solid lines were fitted with least squares polynomials of four-degree. Costs (a) and gain of the strategy $\mathcal{S}^1$ over $\mathcal{S}^0$; (b) for $\alpha = 0.01$.

average cost of transactions under $\mathcal{S}^0$, meaning that the Nash equilibrium will be met at $\theta_0 = 1$. Summing up, the Nash equilibrium θ_0, in these cases, will be met at:

$$\theta_0 = \min\{\bar{p}/\gamma, 1\}.$$

Fig. 8(a) represents a typical graph of average costs of transactions under $\mathcal{S}^0$ and transactions under $\mathcal{S}^1$ as a function of fraction p, for a higher α. In that case, even though the average costs of transactions under $\mathcal{S}^0$ and transactions under $\mathcal{S}^1$ do not coincide for any reasonable p (meaning that, here, the Nash equilibrium will be met at

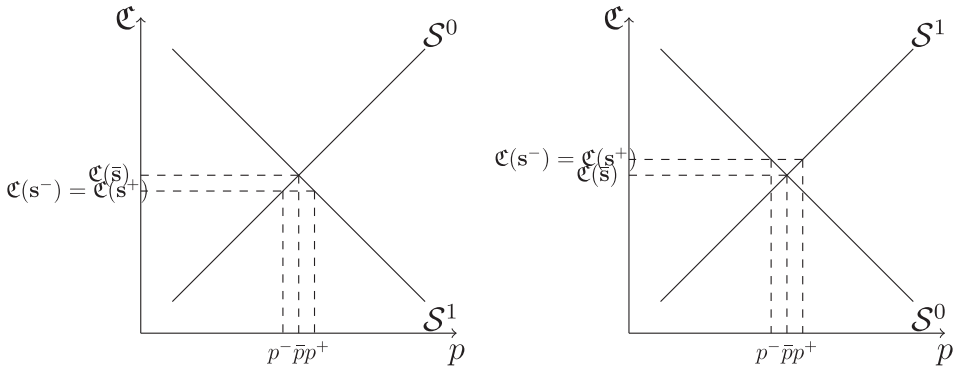

Fig. 7. Different Nash equilibrium points in systems with similar curves.

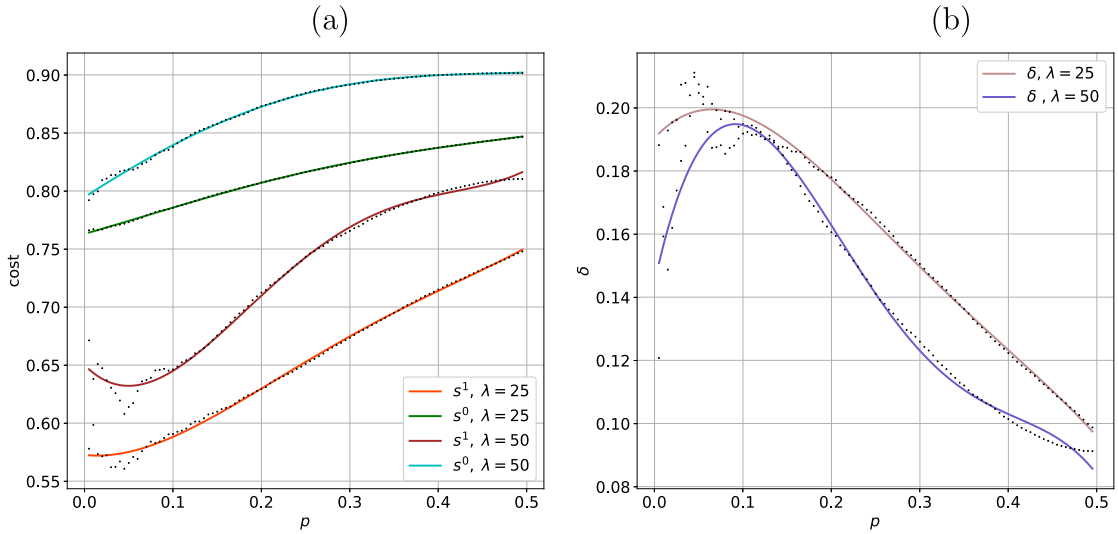

Fig. 8. Costs (a) and gain (b) of the strategy $\mathcal{S}^1$ over $\mathcal{S}^0$; for $\alpha = 0.5$.

$\theta = 1$), the typical difference between the possible pure strategies (that, from now on, we will call absolute gains) will be low, as depicted on Fig. 8(b).

Fig. 9 shows the average cost increase imposed on the nodes following the default strategy by the nodes issuing transactions under $\mathcal{S}^1$. Let $W(p)$ be the non-greedy nodes costs depicted in Fig. 8(a). The cost increase is calculated as $(W(p) - W(0))/W(0)$, so it will be the relative difference of the cost of a non-selfish node in the presence of a fraction p of selfish transactions and the cost of a non-selfish node when there are no selfish transactions at all. This difference is low, meaning that the presence of selfish nodes do not harm the efficiency of the non-selfish nodes. Note that this difference is small for all reasonable values of p, but even for the larger simulated values of p, the difference is still less than 25%. An interesting phenomenon, as shown in the same graph, is that the average cost increase imposed on the non-greedy nodes may actually be negative. For low values of α, just a small fraction of the transactions under $\mathcal{S}^0$ will share the approved tips with the transactions under $\mathcal{S}^1$. This fraction of transactions will approve overcrowded tips, and will have their costs increased. All the other transactions under $\mathcal{S}^0$ will have their sites less crowded, since an increase in $\mathcal{S}^1$ will mean a decrease in competition over these transactions. Finally, on average, the honest nodes will have their costs decreased.

Figs. 10 and 11 are analogous to the first figures, for other values of α and λh; part (a) of each figure represents average costs and part (b) absolute gains.

5. Conclusions and future work

In the first part of this paper, we prove the existence of ("almost symmetric") Nash equilibria for a game in the tangle where a part of players tries to optimise their attachment strategies. In the second part of the paper, we numerically determine, for a simple space strategy and some range of parameters, where these equilibria are located.

Our results show that the studied selfish strategy outperform the non-selfish ones by a reasonable order of magnitude. The data show a 25% (in the most extreme scenario) difference in the nodes gains, which in some situations, may be large enough. Nevertheless, the computational cost of a selfish strategy is intrinsically larger than the computational cost of the non-selfish strategies, since the selfish strategy uses the probability distribution of the tips, which is costly to calculate for a random walk with backtracking. They will also have to monitor the tangle, to know its parameters (like λ, h etc.) and act accordingly. Also, even an extreme scenario, where almost half of the transactions were issued by a selfish node, is not enough to harm the non-selfish ones in a meaningful way.

On the other hand, our results raise further questions. The obtained data exhibit a deep qualitative dependence on the parameter α of the simulation. This parameter is related to the randomness of the random walk: a low α implies a high randomness; a higher α implies a low randomness, meaning that the walk will be almost deterministic. Further simulations will be done in order to study the effect of that

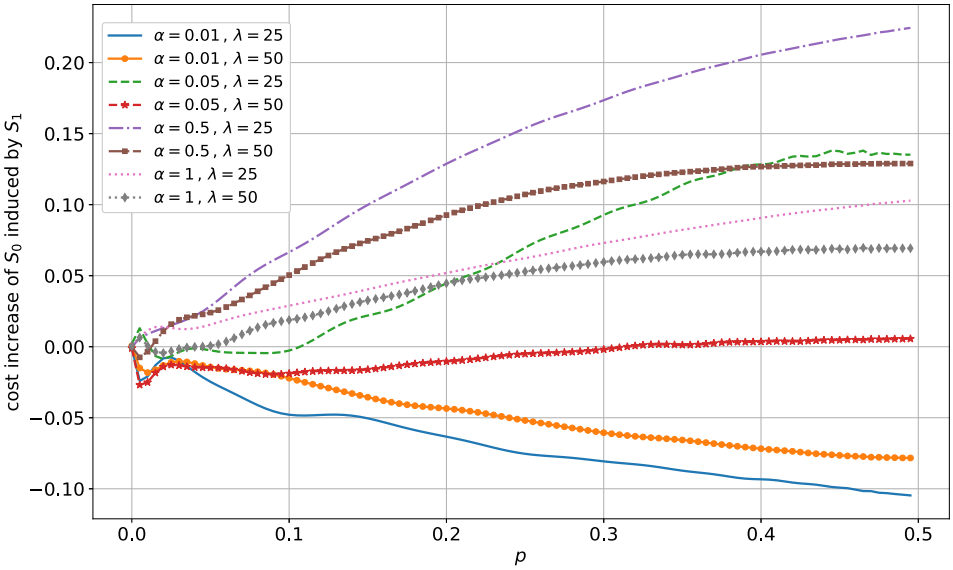

Fig. 9. Relative cost increase of the transactions issued by the strategy $\mathcal{S}^0$ induced by the presence of transactions emitted by the strategy $\mathcal{S}^1$.

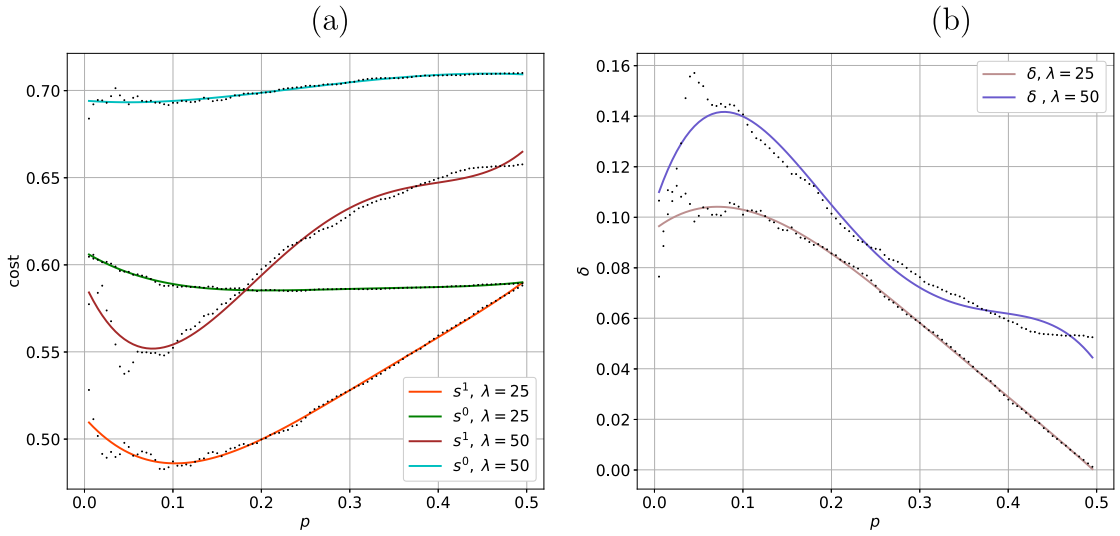

Fig. 10. Costs (a) and gain (b) of the strategy $\mathcal{S}^1$ over $\mathcal{S}^0$; for $\alpha = 0.05$.

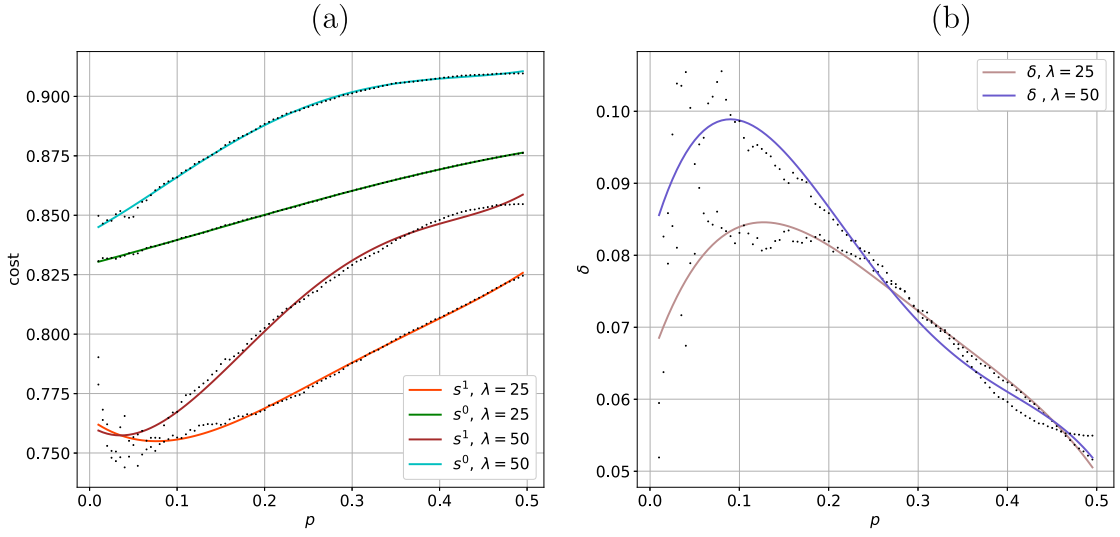

Fig. 11. Costs (a) and gain (b) of the strategy $\mathcal{S}^1$ over $\mathcal{S}^0$; for $\alpha = 1$.

variable in the equilibria. Also, we only studied equilibria for a given cost, relative to the probability of confirmation of the transactions in a certain interval of time. Since this probability depends heavily on the interval of time chosen (because the probability distribution of the confirmations is far from uniform), another time intervals, that will have another practical meaning, must be analysed.

Finally, the equilibrium in the multidimensional strategy space should be studied in a more quantitative and analytic way, since it should depend strongly on α and p; and until now it was studied in just a narrow range of parameters. Further research will also be done in order to optimise the default tip selection strategy in a way that minimises this cost imposed by the selfish strategies. Through implementing research methods and techniques from the cross-reactive fields of measure theory, game theory, and graph theory, progress towards resolving the tangle-related open problems has been well under way and will continue to be under investigation.

As already mentioned, in this paper we consider only "selfish" players, i.e., those who only care about their own costs but still want to use the network in a legitimate way. We do not consider at all the case when there are "malicious" ones, i.e., those who want to disrupt the network even at a cost to themselves. We are going to treat several types of attacks against the network in the subsequent papers. Some preview of this ongoing work is available in Popov (2018).

Acknowledgement

The authors thank Alon Gal, Gur Huberman, Bartosz Kumierz, John Licciardello, Andreas Penzkofer, Samuel Reid, and Clara Shikhelman for valuable comments and suggestions. The authors are also grateful to the anonymous reviewers for carefully reading the first version of this paper and providing valuable comments and suggestions.

References

Baird, L. (2016). The Swirlds hashgraph consensus algorithm: Fair, fast. *Byzantine Fault Tolerance*. http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf.

Churyumov, A. (2016). Byteball: A decentralized system for storage and transfer of value. https://byteball.org/Byteball.pdf.

Cooper, C., & Frieze, A. (2007). The cover time of the preferential attachment graph. *Journal of Combinational Theory B, 97*(2), 269–290.

Cooper, C., Frieze, A., & Pett, S. (2017). The covertime of a biased random walk on Gn, p. 1708.04908.

Cooper, R. B. (1981). *Introduction to queueing theory* (2nd ed.). North Holland.

Doyle, P. G., & Snell, J. L. (1984). *Random walks and electric networks. Carus Mathematical Monographs 22.* Washington: Mathematical Association of America.

Durrett, R. (2012). *Essentials of stochastic processes* (2nd ed.). Springer.

Fey, M. (2012). Symmetric games with only asymmetric equilibria. *Games Economic Behavior, 75*(1), 424–427.

Fink, A. M. (1964). Equilibrium in a stochastic n-person game. *Journal of Science Hiroshima University Series A-I Mathematics, 28*(1), 89–93.

Jerison, D., Levine, L., & Sheffield, S. (2014). Internal DLA and the Gaussian free field. *Duke Mathematics Journal, 163*(2), 267–308.

Kakutani, S. (1941). A generalization of Brouwer's fixed point theorem. *Duke Mathematical Journal, 8*(3), 457–459.

Karlin, A. R., & Peres, Y. (2017). *Game theory, alive.* American Mathematical Society.

Kuśmierz, B., & Gal, A. (2018). Probability of being left behind and probability of becoming permanent tip in the Tangle. https://www.iota.org/research/academic-papers.

Lerner, S. D. (2015). DagCoin: A cryptocurrency without blocks. https://bitslog.wordpress.com/2015/09/11/dagcoin/.

Nash, J. F. (1950). Equilibrium points in n-person games. In: *Proceedings of the National Academy of Sciences, 36*(1), 48–49.

Ok, E. A. (2007). *Real analysis with economics applications.* Princeton University Press.

Popov, S. (2015). The tangle. https://iota.org/IOTA_Whitepaper.pdf.

Popov, S. (2018). Local modifiers in the Tangle. https://www.iota.org/research/academic-papers.

Sompolinsky, Y., Lewenberg, Y., & Zohar, A. (2016). SPECTRE: Serialization of proof-of-work events: confirming transactions via recursive elections. https://eprint.iacr.org/2016/1159.pdf.

Sompolinsky, Y., & Zohar, A. (2013). Secure high-rate transaction processing in Bitcoin. https://eprint.iacr.org/2013/881.pdf.