Equilibria in the Tangle

M.SANDHYA
B.Ed
SRI RAGHAVENDRA COLLEGE OF EDUCATION

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. 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.

Keywords: random walk, Nash equilibrium, directed acyclic graph, cryptocurrencv, tip selection

1 Introduction

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 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

Let us stress also that., we consider only "selfish" players (those who only care about their own costs but still want to use the network in a legitimate way[3]); we do not consider the case when there are "malicious" ones (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.

1.1 Description of the model

In the following we introduce the mathematical model describing the Tangle.

Let card(A) stand for the cardinality of (multi)set A. For an oriented multigraph $T = (V, E)$, where V is the set of vertices and E is the multiset of edges, and $u \in V$, we denote by

$$\deg_{in}(u) = \text{card}\{e = (u_1, u_2) \in E : u_2 = u\},$$

$$\deg_{out}(u) = \text{card}\{e = (u_1, u_2) \in E : u_1 = u\}$$

the "incoming" and "outgoing" degrees of the vertex u (counting the multiple edges). In the following, we refer to rnultigraphs simply as graphs. For $u, v \in V$, we say that u *approves* v, if $(u. v) \in E$. We use the notation $A(u)$ 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, $x_j \in A$ (x_{j-1}) for all $j = 1, \ldots, k$, i.e., there is a directed path from u to v. If $\deg_{in}(w) = 0$ (i.e., there are no edges pointing to w), then we say that $w \in V$ is a *tip.*

Let 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:

- 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 $\deg_{out}(v) = 2$ for all $v \in V \setminus \{\wp\}$, and $\deg_{out}(\wp) = 0$ (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[4] from v to $\wp$ (one can say that the graph is *connected towards* $\wp$).

We now describe the tangle as a continuous-time Markov process on the space g. The state of the tangle at tune $t \geq 0$ is a DAG $T(t)$ ($V_T(t)$, $E_T(t)$, where $V_T(t)$ is the set of vertices and $E_T(t)$ 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 $V_T(0) = \wp$ $E_T(0) = \emptyset$
- The tangle *grows with time*., that is, $V_T(t_1) \subset V_T(t_2)$ and $E_T(t_1) \subset E_T(t_2)$ whenever $0 \leq t_1 < t_2$.
- For a fixed parameter $\lambda > 0$, there is a Poisson process of incoming *transactions;* these transactions then become the vertices of the tangle.
- Each incoming transaction chooses[5] 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 $V_T(t'+) = V_T(t') \cup \{v\}$, and $E_T(t'+) = E_T(t) \cup \{v, v'\}, (v, v'')\}$.

Let us write

$$P(t)(x) = \{y \in T(t) : y \text{ is referenced by } x\},$$

$$F(t)\,(x) = \{z \in T(t) : z \text{ references } x\}$$

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 $P(t)(x) = P(t_0)\,(x)$ for all $t \geq t_0$. We also define the *cumulative weight* $\mathcal{H}_x^{(t)}$ of the vertex x at time t by

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

that is, the cumulative weight of x is one (its "own weight') plus the number of vertices that reference it. Observe that, for any $t > 0$, if y approves x then $\mathcal{H}_x^{(t)} - \mathcal{H}_x^{(t)} \geq 1$, 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[6]. When it is necessary to emphasize that the vertices of $G \in \mathcal{G}$ contair some data, we consider the *marked DAG* $^{(\mathfrak{d})}$ to be $(G,\mathfrak{d} = (V, E, \mathfrak{d})$, where $\mathfrak{d}$ is a funcion $V \longrightarrow$ D. We define g to be the set of all marked DAGs (G,0), where G E $\mathcal{G}$.

1.2 Attachment strategies

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 for the whole system if the new transactions always prefer to select tips as attachment. places, since this way more transactions would he "confirmed'[7]. In any case, it is quite clear that the appropriate choice of the attachment strategy is essential for the correct functioning (whatever this could mean) of the system.

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 st rategy 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 *recom-mended tip selection algorithm,* since, due to reasons described above, the recom-mended 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 $L(t)$ = card $(\mathcal{L}(t))$. To model the network propagation delays, we introduce a parameter $h > 0$, and

assume that at, time t only $T(t\text{-}h)$ 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.

[6]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

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

The initial state of the random walk is the genesis $\wp$[8], 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 $f : \mathbb{R}_+ \to \mathbb{R}_+$ 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 $f(\mathcal{H}_r - \mathcal{H}_y)$, that. is,

$$P_{xy}^{(f)} = \begin{cases} \frac{q}{2}, & \text{if } y \in A(x), \\ \frac{(1)-q)\, f(H_x^{t-h} - H_y^{(t-h)}}{\sum_{z;x\in A(z)} f(H_x^{(t-h)} - H_z^{(t-h)})}, & \text{if } x \in A(y), \\ 0, & \text{Otherwise} \end{cases} \quad (2)$$

(for $x = \wp$ we define the transition probabilities as above, but with $q = 0$). In what follows, we will mostly assume that $f(s) = \exp(-\alpha s)$ for some $a \geq 0$. We use the notation $P^{(\alpha)}$ for the transition probabilities in this case. Intuitively, the smaller is the value of α. the more *random* the walk is[9]. It is worth observing that the case $q = 0$ and $\alpha \to \infty$ corresponds to the GHOST protocol of [21] (more precisely, to the obvious generalization of the GHOST protocol for 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 you 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. $(T(t), t \geq 0)$ 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.

We say that a transaction is *confirmed with confidence* γ_0 (where - γo is some pre-defined number, close to 1), if, with probability at least $-\gamma_0$, the large- α random walk [10] ends in a tip which references that transaction. It. may happen that a transaction

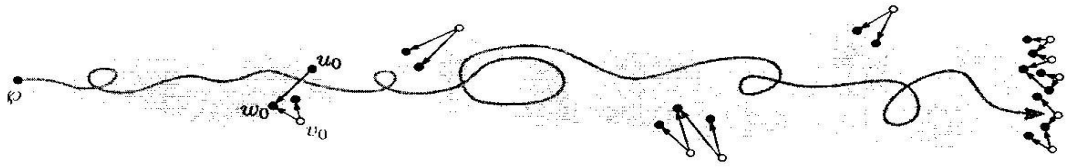

Figure 1: The walk on the tangle and tip selection. Tips are circles and transactions which were approved at least once arc disks.

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 only 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 1.1. *It holds that*

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

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 v_0 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, v_0 is "closest. (in some suitable sense) to the genesis. Let us now think of the following question: what is the probability that v_0 will still be a tip at time $t + 1$?

Look at Figure 1: during the time interval [t, t + 1], $O(1)$ new particles will arrive, and the corresponding walks will travel from the genesis $\wp$ 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 v_0. Assume that w_0 was already confirmed (i.e., 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 t.o the tip v_0 with probability $f(O(t))$.

The probability that $v_0 \in \mathcal{L}(t + 1)$ (i.e., that v_0 still is tip at time $t + 1$) is $f(O(t))$, and the Borel-Cantelli lemma[ll] gives that the probability that v_0 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[12] 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.

It would be better to choose the function f in such a way that, almost surely, every tip eventually gets confirmed,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 reattached[13] 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 rot 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 (roughly) does not depend on the moment when it appeared in the system.

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 be-longing to the queuing theory, known as the Little's formula (sometimes also referred to as the Little's theorem or the Little's identity):

2 Selfish nodes and Nash equilibria

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

$\mathcal{M}(A)$ ={μ : A $\longrightarrow$ $\mathbb{R}$ such that μ (a) $\geq$0 for all $a \in A$ and $\sum_{a\in A}\mu(a) = 1$}
Let

$$\mathfrak{W} = \bigcup_{G=(V,E)\in\mathcal{G}} \mathcal{M}(V\ x\ V)$$

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, an *attachment strategy* $\mathcal{S}$ is a map

$$\mathcal{S} : \mathcal{G}^{[\mathfrak{d}]} \longrightarrow \mathfrak{W}$$

with the property $\mathcal{S}$ (V, E, $\mathfrak{d}$) $\in \mathcal{M}(V\ x\ V)$ 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 $k > 0$ be a fixed number. We now assume that, for a (very) large N, there are kN nodes that follow the default tip selection algorithm, and N "selfish" nodes that try to minimize their "cost", whatever this could mean[16]. Assume that all nodes issue transactions with the same rate $\frac{\lambda}{(k+1)N}$, independently. The overall rate of "honest" transactions in the system is then equal to $\frac{\lambda k}{k+1}$, and the overall rate of transactions issued by selfish nodes equals $\frac{\lambda}{(k+1)}$. Let S_1,...,S_N 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 some 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* $v \in T(t)$ for all $t \geq t_v$. Fix some (typically large) $M_0 \in \mathbb{N}$. Let $t_1^{(v)}$,... $t_{M0}^{(v)}$ be the moments when the subsequent (after v) transactions were attached to the tangle. For $k = 1$. M_0 let $R_k^{(v)}$ be the event that the *default* tip-selecting walk[17] on T ($t_k^{(v)}$) stops in a tip that *does not* reference v. We then define

$$W(v) = 1_{R_1^{(v)}} + ... + 1_{R_{M_0}^{(v)}} \qquad (4)$$

to be the number of times that the M_0 "subsequent" tip selection random walks do not. reference v (in the above, 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.

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

$$\mathfrak{C}^{(k)}(S_1,....,S_N) = M_0^{-1} \lim_{n\to\infty} \frac{W(v_j^{(k)})+\cdots+W(v_n^{(k)})}{n}, \qquad (5)$$

to be the *mean* cost of the *k*th node given that (S_1,....,S_N) are the attachment strategies of the selfish nodes.

3 Simulations

we investigate Nash equlibria between selfish nodes via simulations. 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 equlibrium is"bad for everybody" , effectively leading to the system's malfunctioning (again, we do not specify the exact meaning of that)?

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 equlibrium strategy of the selfish nodes will not be much different from the (suitably chosen) default strategy. But, before doing that, let us explain the intution behind this fact. Naively, a natural strategy for a selfish node would be the following.

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

(2) Find the two tips where this distribution attains its "best" [21] values.

(3) Approve these two tips.

This strategy fails when other selfish nodes are present. To understand this, look at Figure 2: *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[22].

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. Also, a transaction will be reattached if the two following criteria are met:

(1) the transaction is older than 20 seconds[23];
(2) the transaction is not referenced by the tip selected by a random walk with $\alpha = \infty$[24]

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.

It depicts the typical cumulative distribution of the time of the first approval, for a low α and $\lambda h = 50$. Note that roughly 95% of the transactions will be approved before t = 5s, and almost the totality of transactions will be approved before t = 10s. That behaviour will be similar for all studied parameters. The average cost defined in equations (5) and (4) will have a certain meaning, depending on the choice of M_0. This average cost will be related to the average time of approval of a transaction (indeed, the average time will be approximately W/λ). So, in both cases (λ = 25 and λ = 50), the mean cost was calculated over the transactions attached at a interval of time of approximately 10s (M_0 = 500 for λ = 50 and M_0 = 250 for λ = 25), what makes this cost something reasonable to optimise.

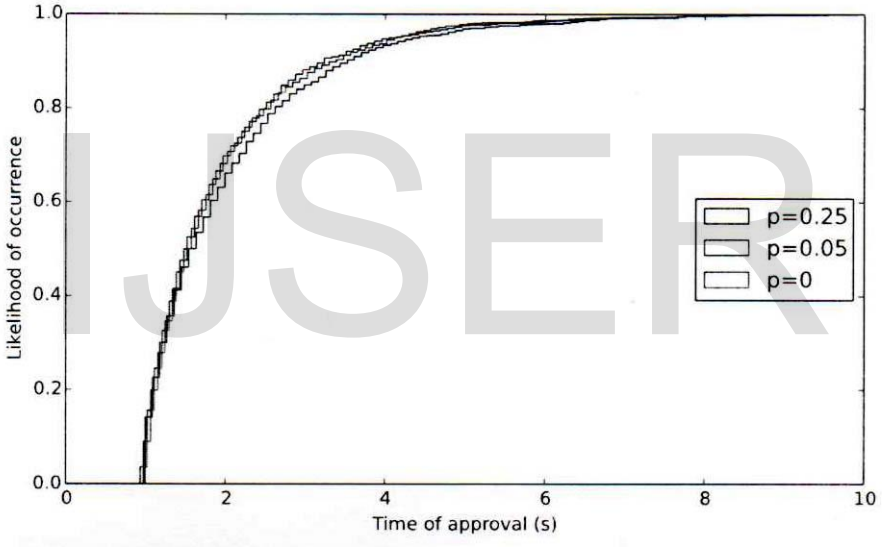

Figure 3: Cumulative distribution of time of approvals for some values of p (that will be defined later)

3.1 One dimensional Nash equilibria

we will study the Nash equilibria of the tangle problem, considering the following strategy subspace:

$$S_i = S = (1 - \theta)S^0 + \theta\, S^1 \qquad \text{for each } i = 1,...., N\,,$$

where S^0 is the default tip selection strategy, S^1 is the selfish strategy defined in the beginning of this section and $\theta \in [0,1]$. The goal is to find the Nash equilibria relative to the costs defined in the last section (equations (5) and (4)). The selfish nodes will try to optimise their transaction cost with respect. to θ.

Now, suppose that we have a fixed fraction γ of selfish nodes, that chooses a strategy among the possible *S*. The non-selfish nodes will not be able to choose their strategy, so they will be restricted, as expected. to S^0. Note that, Since they can not choose their strategy, they will not "play" the game. Since the costs are linear over *S*, such mixed strategy game will be equivalent to a second game where only a fraction $p = \gamma\theta \leq \gamma$ of the nodes chooses S^1 over S^0, and the rest of the nodes chooses S^0 over S^1.

Figure 4 (a) represents a typical graph of average costs of transactions issued under S^o and S^I, as a function of the fraction *p*, for a low a α and two different values of λ. As already demonstrated, when in equilibrium, the selfish nodes should issue transactions with the same average cost. That. means that the system should reach equilibrium in one of the following states:

(1) some selfish nodes choose S^0 and the rest choose $S^{1;}$

(2) all selfish nodes choose S^1;

(3) all selfish nodes choose S^0.

(a) (b)

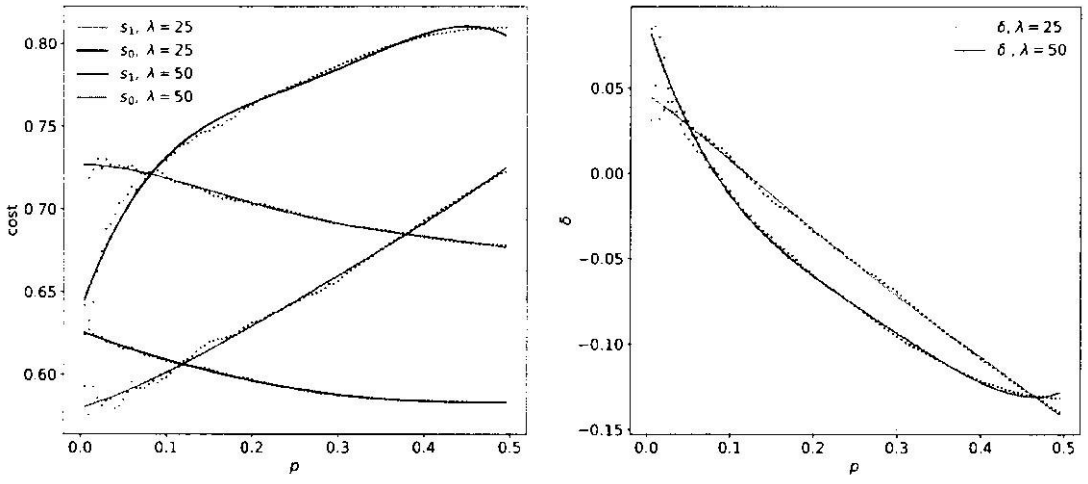

Figure 4: Costs (a) and gain of the strategy S^1 over 8^0; (b) for α = 0.01.

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 calls s the vector of strategies on equilibrium and p the fraction of nodes that will issue transactions under S_1 when the system is in 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 S^0 to S^1 and from S^1 to S^0, respectively. We also define $\bar{s}_1$ and $\bar{s}_2$ as strategy vectors related to $p^-$and p^+. Note on Figure 3.1 that this kind of Nash equilibrium candidate may not he a real equilibrium, when the system is at point $\bar{p}$ and a

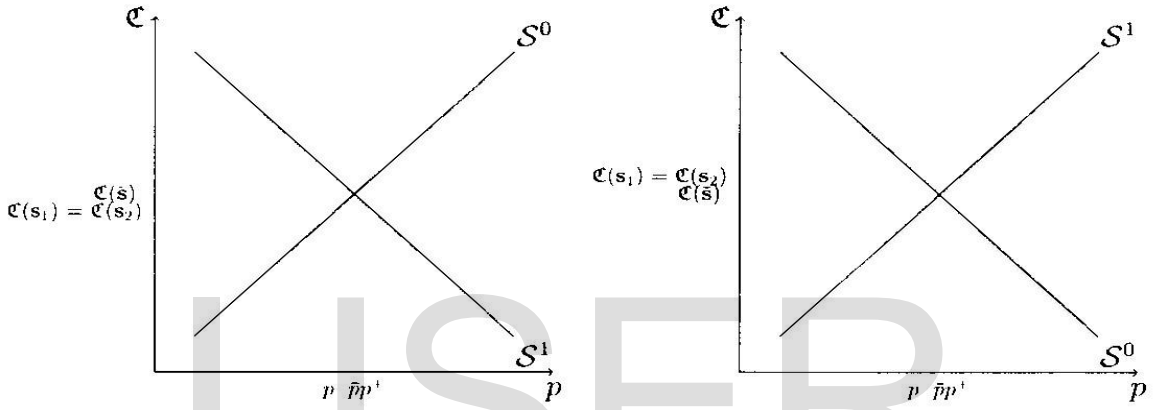

Figure 5: Different Nash equilibrium points in systems with similar curves

node switches strategies, from S^0 to S^1, the cost actually decreases so $\bar{p}$ cannot be a Nash equilibrium. On the other hand, the second example shows a Nash equilibrium at point $\bar{p}$, since deviations to p^- and p^+ will increase costs.

let us re-examine Figure 4(a). Here, the Nash equilibrium will occur atthe point $\bar{p}$, since we have a situation as on Figure 3.1(b). That point is easily found at Figure 4(b), when $\delta = 0$. Note that the Nash equilibrium for a larger λ will be at a smaller $\emptyset_0$ 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 Figure 2 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 S^1.

Reconsider now the mixed strategy game. In the case when all the nodes are allowed to choose between the two pure strategies (S^0 and S^1), the Nash equilibrium will be indeed at $\emptyset_0 = \bar{p}$ (as expected, since in this case $\gamma = 1$). If just a fraction $\gamma = p/\emptyset > \bar{p}$ of the nodes is selfish, then the Nash equilibrium will occur when $\emptyset_0 = \bar{p}/\gamma$. Now, if $\gamma \leq \bar{p}$, the costs of the nodes will not coincide[25]. In that case, the average cost of transactions under S^1 will always be smaller than the average cost of transactions under S^0, meaning that the Nash equilibrium will be met at $\emptyset_0 = 1$. Summing up, the Nash equilibrium $\emptyset_0$, in these cases, will he met at:

$$\emptyset_0 = \min\{\bar{p}/\gamma, 1\}.$$

(a) (b)

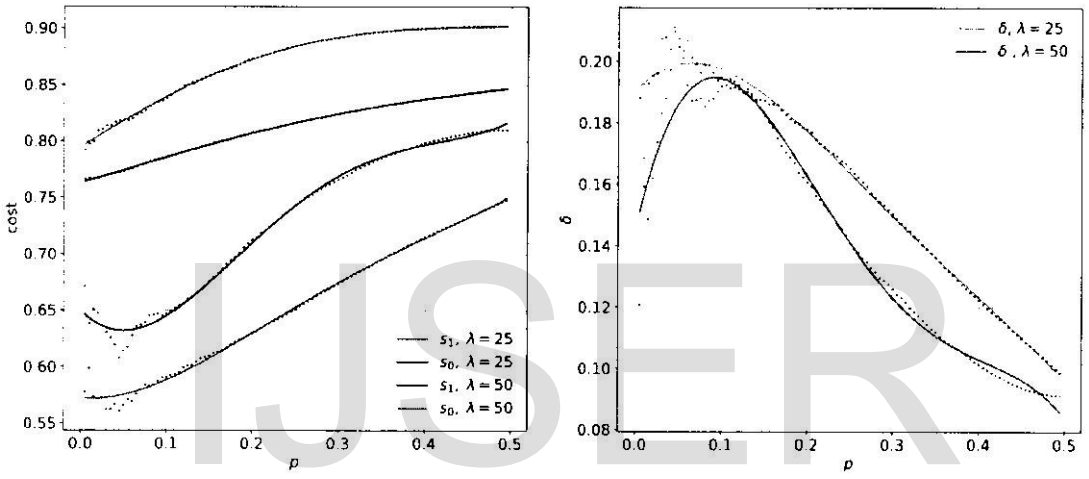

Figure 6: Costs (a.) and gain (b) of the strategy S^1 over S^0; for α = 0.5.

Figure 6(a) represents a typical graph of average costs of transactions under S^0 and transactions under S^1 as a function of fraction *p*, for a higher α. In that case, even though the average costs of transactions under S^0 and transactions under S^1 do not coincide for any reasonable *p* (meaning that, here, the Nash equilibrium will be met at $\emptyset$ = 1), the typical difference between the possible pure strategies (that, from now on, we will call absolute gains) will be low, as depicted on Figure 6(b).

Figure 7 shows the average cost increase imposed on the nodes following the default strategy by the nodes issuing transactions under S^1. Let $W(p)$ be the non-greedy nodes costs depicted in Figure 6(a). The cost increase is calculated as $(W(p) - W(0)/W(0)$, so it will be the perceptual difference of the cost of a non-selfish node in the presence of a percentage *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 available *p*, the difference is 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 S° will share the approved tips

with the transactions under S^1. This fraction of transactions will approve overcrowded tips, and will have their costs increased. All the other transactions under S° will have their sites less crowded, since an increase

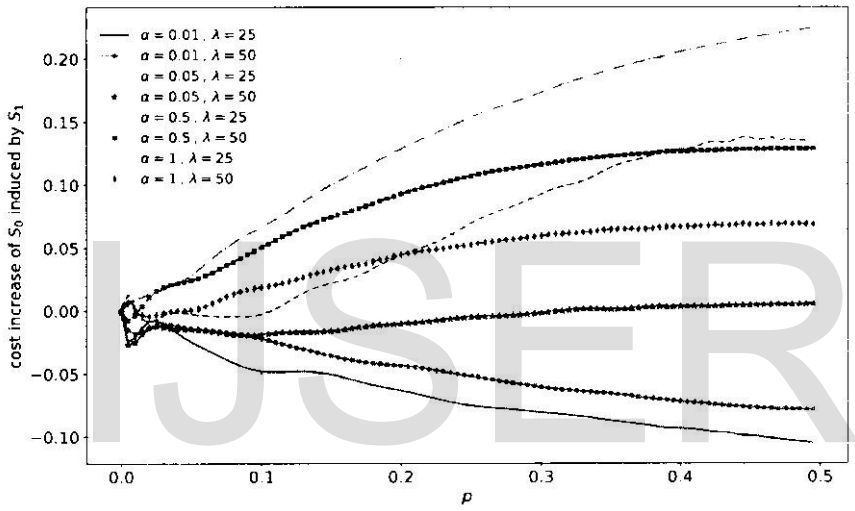

Figure 7: Cost increase of the transactions issued by the strategy S^0 induced by the presence of transactions emitted by the strategy S^1. in percentage.

in S^1 will mean a decrease in competition over these transactions. Finally, on average, the honest nodes will have their costs decreased.

Figures 8 and 9 are analogous to the first figures, for other values of α and λ*h*; part (a) of each figure represents average costs and part (b) absolute gains.

3.2 Multidimentional Nash equilibria

In the same way as the last section, a game with a. mixed strategy space:

$$S = \sum_{i=0}^{N} \theta_i S^i,$$

where $S = \sum_{i=0}^{N} \theta_i S^i$, = 1 and $\{S^i\}_{i=0,\ldots,N}$ a set of strategies is equivalent to a game where the nodes must choose among the simple strategies $\{S^i\}$. That means all the possible strategies must have the same cost in order to the game reach some equilibrium. In this case, in order to simplify the data, the studied object was the probability of a given tip to be chosen by the selfish nodes and the non-selfish nodes. These studied tips were ordered by the random walk exit probability, from the most probable to the least probable. Figure 10 represents the typical probability profile for the selfish and non-selfish nodes. The typical gain of the selfish nodes and the increase of the

(a) (b)

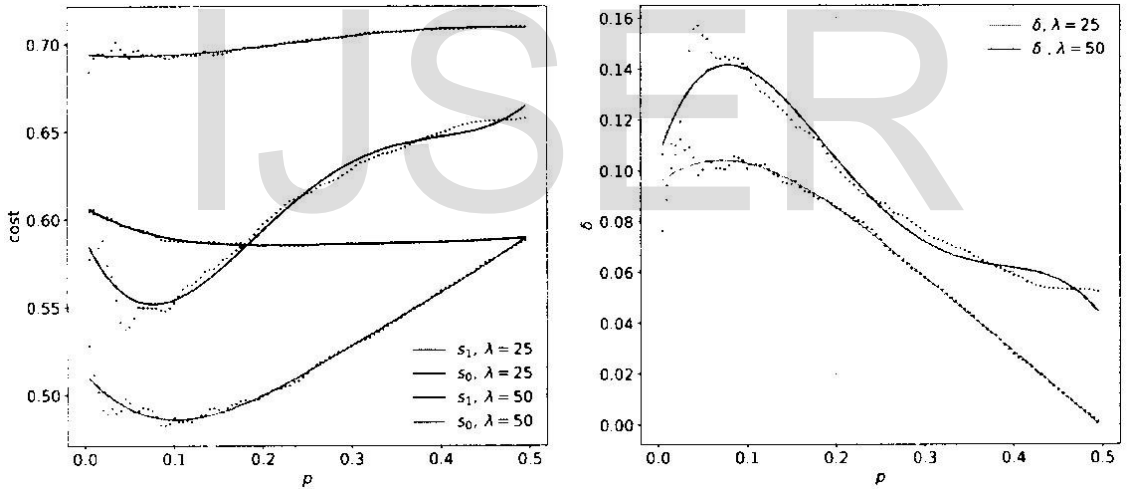

Figure 8: Costs (a) and gain (b) of the strategy S^1 over S^0; for $\alpha = 0.05$.

(b) (b)

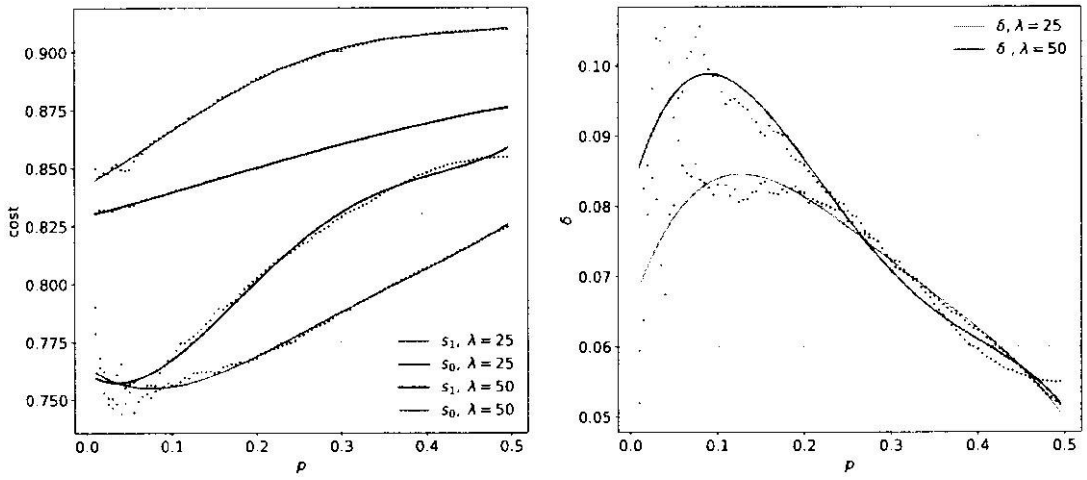

Figure 9: Costs (a) and gain (b) of the strategy S^1 over S^0; for α =1.

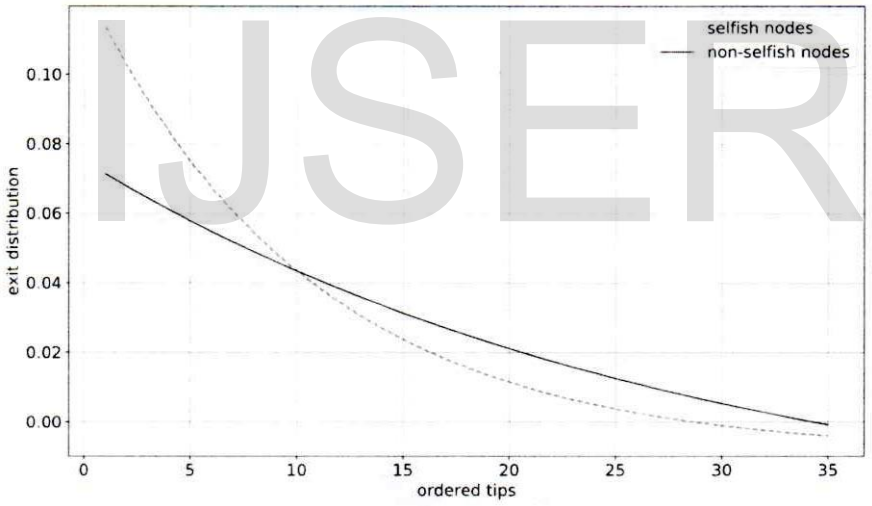

Figure 10: Exit probabilities in the equilibrium for the selfish and the non-selfish nodes.

non-selfish nodes average cost due to the presence of selfish nodes are both small; in our simulations they were always less than 10% (3.615% for the maximum gain and 7.0 % for the maximum of the cost increase of the non-selfish nodes induced by the selfish ones, comparing all parameters of λ and α).