Game theory analysis of incentive distribution for prompt generation of the proof tree in zk-SNARK based sidechains

1st Yuri Bespalov
Bogolyubov Institute for Theoretical Physics
Kiev, Ukraine
yu.n.bespalov@gmail.com

2nd Lyudmila Kovalchuk
Input Output
Kyiv, Ukraine
lyudmila.kovalchuk@iohk.io

3rd Hanna Nelasa
Zaporizhzhia Polytechnic National University
Zaporizhzhia, Ukraine
annanelasa@gmail.com

4th Roman Oliynykov
Input Output
Kharkiv, Ukraine
roman.oliynykov@iohk.io

5th Alberto Garoffolo
Horizen
Milan, Italy
alberto@horizen.global

***Abstract*—In sidechains with Latus consensus, a block forger generates a block using SNARK-proofs, created by decentralized provers and organized in a perfect binary tree (proof tree). One of the most important questions is to assign incentives for these proofs. In this paper, the game theory instruments are used to investigate incentive distribution in proof trees for SNARK-based sidechains to provide stable and efficient block generation. Two different models are considered: when only one sidechain exists, and when there exist a lot of sidechains among which provers may switch any time, trying to get a higher incentive. Utilization of Stirling numbers with non-integer arguments turns out to be very efficient for the second model.**

***Index Terms*—Blockchain, sidechain, Nash equilibrium, Merkle tree, Stirling numbers of the second kind with non-integer arguments**

I. Introduction

This paper works with a scalability solution in the blockchain technology - sidechains [1], [2], [3], [4]. A sidechain is a parallel blockchain bound to the main one, and it provides an additional functionality that for some reasons cannot be available in the original blockchain that is called a mainchain. Here we investigate the Latus Protocol [5] that is a PoS consensus based on the Ouroboros Praos [6] with an additional feature of binding to a PoW mainchain. This binding is needed to provide such necessary blockchain property as persistence [7]. A sidechain sends some information to the mainchain that contains a series of recurrent zk-SNARK-proofs [8], [9]. Such information also allows to establish decentralized and verifiable cross-chain transfers. Latus utilizes a recursive composition of SNARKs to construct a succinct proof of the sidechain state progression for the period of a withdrawal epoch. Then, a SNARK for a withdrawal certificate is constructed so that it proves correct sidechain state transition for the whole epoch, and validates backward transfers. That allows the mainchain to verify the sidechain efficiently without having to rely on an intermediary - such as certifiers [4] - and still be oblivious to the sidechain construction and interactions within it. Block generation in Latus is going on as follows. An entity that creates a block, a block forger, shares a list of transactions that he intends to include into the block. Then the task of zk-SNARK proofs generation is randomly assigned to interested parties called provers. They perform these tasks in parallel, and then submit generated proofs for the sidechain, getting some reward for each accepted proof. Provers construct SNARK-proofs not only for these transactions, but also for each node of the corresponding Merkle tree. We will call such a tree "a proof tree". Each prover sets prices for his proofs, according to the price policy of the current epoch that was set at the end of the previous one. If there are several proofs for a certain node, the block forger chooses the cheapest one. The general purpose of sidechains investigation is to ensure their stable operation (high throughput) via optimal incentive system for all participants.

This paper is a natural continuation of our previous papers [10], [11], where we obtained the following results:

- found estimates for the number of steps to build a complete set of SNARK proofs in the proof tree for blockchains, for two different types of proof construction models: those in which all the proofs to be built are independent (they can be considered as leaves on the proof tree) and those in which the proofs are located at all nodes of the proof tree, and hence form a partially ordered set [10];
- obtained necessary and sufficient conditions for existence of the Nash equilibrium, for various models, price policies, and parameters, both in pure and mixed game theoretic strategies that allows simulation and sometimes prediction of the provers' behavior, proof prices and block forgers' rewards, and help in adequate setting of the price policy, to provide stable operation of the sidechain [11].

In this paper, we first analyze the relation between incentive distributions on the set of proofs and the corresponding Nash equilibrium in the case where all provers are located on the same sidechain and cannot switch to other ones. We consider the game when we can set various possible prices for different proofs, and investigate the question how distribution of incentives for proofs influence their choice by provers. We formulate the relevant definition of a one-step symmetric strategic game and also obtain results on the Nash equilibrium in this game. We show that the Nash equilibrium in mixed strategies occurs when the probabilities in the corresponding distribution are proportional to the incentives for the proofs with which provers are working. It means that provers choose different proofs with probabilities that are approximately proportional to the corresponding incentives. Using this result, we can set a desirable distribution on the set of proofs by setting corresponding incentives to these proofs.

Then, we consider the case when provers can switch arbitrarily among different sidechains choosing more profitable conditions at every moment. Formulas (7), (8) for probabilities to built a level of the tree and the whole tree use Stirling numbers of the second kind whose arguments are usually integer [12]. It makes us to use rounding, which, in turn, gives a "ragged" function. Stirling numbers of complex arguments were proposed in [13]. Our use of this generalization turns out to be a very natural interpolation, a suitable approximation for this model and also convenient for calculation. The results obtained in this case are half-empirical. Through a lot of experiments we show that there is no sense to change proof prices for different levels of the proof tree, and the best solution with respect to the incentive distribution is to set them approximately equal despite of the level number.

II. Preliminaries

Let us recall some basic definitions from the game theory. More details can be found in one of the textbooks, in particular [14].

Definition 1. A *strategic form game* consists of

- the set of *players* $P = \{1, 2, \dots, m\}$;

and for each player i, of

- a non-empty set of *pure strategies* S_i;
- a *utility (payment) function* $u_i : \prod_{i\in P} S_i \to \mathbb{R}$.

A *strategy profile* is a combination of strategies of each player, i.e. an element of the Cartesian product $\prod_{i\in P} S_i$.

Definition 2. A game is called *symmetric*, if all strategy sets S_i are the same, and for each permutation π of strategies

$$u_{\pi(i)}(s_1, \dots, s_m) = u_i(s_{\pi(1)}, \dots, s_{\pi(m)}).$$

In this case, u_i is a symmetric function of all its arguments except for the ith.

Note that the replacement of the left action of symmetric group by the right action leads to a stronger concept of a *fully symmetric game* [15].

Definition 3. If the sets of strategies are equipped by a topology, one can consider the corresponding Borel σ-algebra.

A *mixed (randomized) strategy* μ_i is a Borel probability measure on the set of strategies S_i.

The utility for mixed strategy μ_j on jth place is the expectation calculated via a Lebesgue integral:

$$u_i(\dots, \mu_j, \dots) = \int_{S_j} u_i(\dots, s_j, \dots) d\mu(s_j).$$

Definition 4. A *pure strategy Nash equilibrium* is a strategy profile $(s_i)_{i\in P} \in \prod_{i\in P} S_i$, where for each $i \in P$,

$$\begin{aligned} &u_i(s_1 \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_m) \\ &\quad \geq u_i(s_1 \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_m) \qquad \text{for all } s'_i \in S_i. \end{aligned}$$

A *mixed strategy Nash equilibrium* is a mixed strategy profile $(\mu_i)_{i\in P}$, where for each $i \in P$,

$$\begin{aligned} &u_i(\mu_1 \dots, \mu_{i-1}, \mu_i, \mu_{i+1}, \dots, \mu_m) \\ &\quad \geq u_i(\mu_1 \dots, \mu_{i-1}, \mu'_i, \mu_{i+1}, \dots, \mu_m) \end{aligned}$$

for all mixed strategies μ'_i on S_i.

In this section we consider a symmetric game, and look for a symmetric Nash equilibrium given by the same probability measure μ^* repeated m times. In this case, we can formulate an equivalent criterion of a Nash equilibrium, making comparisons only with pure strategies.

Lemma 1. *For any symmetric game, a symmetric Nash equilibrium is given by a probability measure μ^*, iff the utility*

$$u_1(s_1, \mu^* \dots, \mu^*) = \int_{S^{m-1}} u_1(s_1, \dots, s_m) \prod_{j=2}^{m} d\mu^*(s_j) \quad (1)$$

satisfies the condition

$$\operatorname{supp} \mu^* \subseteq \arg\max_{s_1 \in S} u_1(s_1, \mu^* \dots, \mu^*), \quad (2)$$

where $\operatorname{supp} \mu^*$ *is the support of the game* μ^*.

In this case, the game price is

$$u^* = \max_{s_1 \in S} u_1(s_1, \mu^* \dots, \mu^*).$$

III. Incentive game

In this section, we consider the game when we can set different possible prices for different proofs, and investigate the question how the distribution of incentives for proofs influences their choice. We formulate the corresponding definition of a one-step symmetric strategic game, and also obtain results on the Nash equilibrium in this game. We show that the Nash equilibrium occurs in the case when provers choose different proofs with probabilities that are approximately proportional to the corresponding incentives. It means that we can set a desirable distribution on the set of proofs by setting corresponding incentives to these proofs.

A. Fixed incentives

Here we describe a symmetric Nash equilibrium for a simple model, when each proof has its own unique fixed price (incentive). Such model may be useful when we need to enforce provers to create some proofs before others.

Definition 5. Symmetric strategic game.

- The set of players ($\equiv$ provers) is $\{1, 2, \ldots, m\}$.
- The set of pure strategies for each player is the set of proofs $\{1, 2, \ldots, n\}$. The incentive q_j is assigned for each proof j. The utility of the game is

$$u_i(s_1, \ldots, s_m) = \frac{q_{s_i}}{\#\{i' | s_{i'} = s_i\}}$$

Lemma 2. *The utility* (1) *for the above game is*

$$u_1(s_1, \mu^*, \ldots, \mu^*) = \frac{q_{s_1}}{m\mu^*(s_1)} \left(1 - (1 - \mu^*(s_1))^m\right). \quad (3)$$

Proof. Using (1),

$$\begin{aligned}
&u_1(s_1, \mu^*, \ldots, \mu^*) \\
&= \sum_{s_2, s_3, \ldots, s_m \in \{1, \ldots, n\}} \frac{q_{s_1}}{\#\{i | s_i = s_1\}} \prod_{j=2}^{m} \mu^*(s_j) \\
&= \sum_{k=0}^{m-1} \frac{q_{s_1}}{k+1} \binom{m-1}{k} \mu^*(s_1)^k (1 - \mu^*(s_1))^{m-k-1} \\
&= \frac{q_{s_1}}{m\mu^*(s_1)} \sum_{k=0}^{m-1} \binom{m}{k+1} \mu^*(s_1)^{k+1} (1 - \mu^*(s_1))^{m-k-1} \\
&= \frac{q_{s_1}}{m\mu^*(s_1)} \left(1 - (1 - \mu^*(s_1))^m\right).
\end{aligned}$$

□

The following proposition sets connections between distribution of incentives on proofs and distribution of probabilities to choose the corresponding proof.

Proposition 1. *To obtain a symmetric Nash equilibrium given by a mixed strategy μ^* with $\mu^*(i) = p_i$, one should select incentives according to the proportion*

$$\begin{aligned}
q_i &\sim \frac{p_i}{1 - (1 - p_i)^m} \\
&= \frac{1}{1 + (1 - p_i) + \ldots + (1 - p_i)^{m-1}}.
\end{aligned}$$

Proof. According to (2), the utility (3) should be independent of s_1. □

Example 1. To obtain the Nash equilibrium on a uniform distribution, one should select equal values for all q_i.

Example 2. If $m \gg 1$, i.e. such that $(1 - p_i)^m \ll 1$ for all i, to obtain a Nash equilibrium one should select all q_i proportional to p_i.

Example 3. If $m = 2$, then to obtain a Nash equilibrium one should select $q_i \sim 1/(2 - p_i)$.

B. Incentives from intervals

Here we generalize our model from the previous subsection. In this subsection, we assume that some incentive interval is assigned to each proof. Each prover may choose the proof, and then set the price from the corresponding interval, so this strategy consists of two elements - a proof number and its price.

Definition 6. Symmetric strategic game.

- The set of players ($\equiv$ provers) is $\{1, 2, \ldots, m\}$.
- There are given the set of proofs $\{1, 2, \ldots, n\}$ and an interval of incentives $[q_{j,\min}, q_{j,\max}]$ is assigned to each proof j.
  Let a pure strategy for each player be an element s of the disjoint union of all these intervals $\coprod_{1 \le j \le n} [q_{j,\min}, q_{j,\max}]$, i.e. a pair $(s^\dagger, s^\ddagger)$ with $1 \le s^\dagger \le n$ and $s^\ddagger \in [q_{s^\dagger,\min}, q_{s^\dagger,\max}]$.
  The utility is

$$\begin{aligned}
&u_i(s_1, \ldots, s_m) \\
&= \begin{cases} \dfrac{s^\ddagger_i}{\#\underset{i' \in \{i'' | s^\dagger_{i''} = s^\dagger_i\}}{\arg\min} (s^\ddagger_{i'})}, & \text{if } i \in \underset{i' \in \{i'' | s^\dagger_{i''} = s^\dagger_i\}}{\arg\min} (s^\ddagger_{i'}) \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

A mixed strategy is a probability measure on $\coprod_{1 \le j \le n} [q_{j,\min}, q_{j,\max}]$, or equivalently a discrete probability distribution $(p_j)_{1 \le j \le n}$ on proofs together with the probability measure μ_j on $[q_{j,\min}, q_{j,\max}]$ absolutely continuous with respect to the Lebesgue measure for $1 \le j \le n$.

Lemma 3. *For a mixed strategy $\mu^* = (p_j, \mu_j)_{1 \le j \le n}$*

$$\begin{aligned}
&u_1(s_1, \mu^*, \ldots, \mu^*) \\
&= \frac{s^\ddagger_1 \left((1 - p_{s^\dagger_1} + p_{s^\dagger_1} \mu_1([s^\ddagger_1, q_{s^\dagger_1,\max}]))^m - (1 - p_{s^\dagger_1})^m \right)}{m p_{s^\dagger_1} \mu_1([s^\ddagger_1, q_{s^\dagger_1,\max}])} \\
&= \frac{s^\ddagger_1}{m} \sum_{k=0}^{m-1} (1 - p_{s^\dagger_1} + p_{s^\dagger_1} \mu_1([s^\ddagger_1, q_{s^\dagger_1,\max}]))^k (1 - p_{s^\dagger_1})^{m-k-1}.
\end{aligned}$$

Proof. Using (1),

$$\begin{aligned}
&u_1(s_1, \mu^*, \ldots, \mu^*) \\
&= \sum_{s^\dagger_2, s^\dagger_3, \ldots, s^\dagger_m \in \{1, \ldots, n\}} p_{s^\dagger_2} \cdots p_{s^\dagger_m} \\
&\qquad \times \int u_1(s_1, \ldots, s_m) \, d\mu_2(s^\ddagger_2) \cdots d\mu_m(s^\ddagger_m).
\end{aligned}$$

$$
\begin{aligned}
&u_1(s_1,\mu^*,\dots,\mu^*)\\
&\quad=\sum_{k=0}^{m-1}\binom{m-1}{k}\frac{s_1^\ddagger}{k+1}p_{s_1^\dagger}^k(1-p_{s_1^\dagger})^{m-k-1}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}])^k\\
&\quad=\frac{s_1^\ddagger}{mp_{s_1^\dagger}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}])}\\
&\qquad\times\sum_{k=0}^{m-1}\binom{m}{k+1}p_{s_1^\dagger}^{k+1}(1-p_{s_1^\dagger})^{m-k-1}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}])^{k+1}\\
&\quad=\frac{s_1^\ddagger}{mp_{s_1^\dagger}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}])}\\
&\qquad\times\left((1-p_{s_1^\dagger}+p_{s_1^\dagger}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}]))^m-(1-p_{s_1^\dagger})^m\right).
\end{aligned}
$$

□

Proposition 2. *To obtain a symmetric Nash equilibrium given by a mixed strategy $\mu^*=((p_j)_{1\le j\le n},(\mu_j)_{1\le j\le n})$ one should select incentives according to the proportion*

$$q_{i,\max}\sim(1-p_i)^{1-m}. \tag{4}$$

Proof. According to (2), the utility (3) should be independent of s_1.

$$
\begin{aligned}
&\frac{s_1^\ddagger}{m}\sum_{k=0}^{m-1}(1-p_{s_1^\dagger}+p_{s_1^\dagger}\mu_1([s_1^\ddagger,q_{s_1^\dagger,\max}]))^k(1-p_{s_1^\dagger})^{m-k-1}\\
&\qquad\qquad=q_{,\max}(1-p)^{m-1}.
\end{aligned}
$$

□

The results obtained in this section for two different models may be interpreted as follows:

- according to Proposition 1, in the case of fixed values of incentives and of a sufficiently large number of provers, we should assign incentives proportional to a desired probability distribution, to guarantee stable functioning of the sidechain;
- according to Proposition 2, in the case when provers may choose incentives from some intervals themselves, to achieve stability in the sidechain we should assign right ends of the intervals, as in the proportion (4).

IV. Incentives for tree levels

Suppose that some mainchain has many different sidechains, and the provers at each step can choose which sidechain to work on. Indeed, it is the most advantageous to build proofs at the lowest level of the tree, since in this case the probability that the created proof will be accepted is the largest. Therefore, one possible strategy for provers could be to switch to the sidechain where the bottom levels of the tree is currently being built. Based on this, the question arises: how to distribute incentives for each level of the tree in order to attract the optimal number of provers to work with this level? In the previous section, for the case of one sidechain, a result was obtained showing that incentives must be larger to attract more provers, and vice versa. Following this logic, incentives would need to decrease as the level number increases in order to attract fewer provers for fewer proofs. In what follows, we formulate a theoretical background and then test this assumption experimentally.

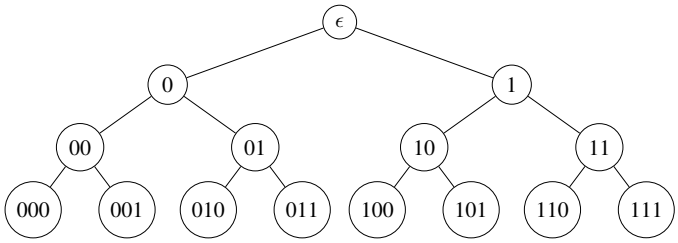

Fig. 1. The perfect binary tree M_4 with nodes labeled by binary strings

A. Probabilities for proof-tree construction

Let us introduce the following notations. We denote by v_p the average incentive received by the prover for the completed proof (taking into account also those proofs that were rejected due to duplicated generation by several provers). Next, let the blockforger want to include the $2^{\ell-1}$ transaction for some nonnegative integer ℓ. Therefore, in order to successfully create a block, provers must build a proof tree consisting of ℓ levels.

Let us denote by M_ℓ the corresponding proof tree.

Definition 7. A rooted binary tree is called *perfect*, if all its interior nodes have two children and all leaves have the same depth (i.e. distance to the root).

The perfect binary tree M_ℓ with ℓ levels has $n=2^{\ell-1}$ leaves and, more generally,

$$n_i=2^{\ell-i}$$

nodes on i-th level $(1\le i\le\ell)$ starting from the bottom. These nodes may be subsequently indexed by $0\le j<2^{\ell-i}$, and the binary string representation has length $\ell-i$. See Figure 1.

We suppose that the blockforger's fee for a block is proportional to the number of transactions. So, we can subtract this sum and consider the part of the transaction fee v_t used to pay to provers. Let us denote $\nu=v_t/v_p$.

We also assume that on i-th level we will pay v_i for a proof, and m_i provers work at this level. The equilibrium condition that each prover expects to obtain v_p for a proof takes the form

$$m_iv_p\approx n_iv_i.$$

Let us denote $\nu_i=v_i/v_p$ and set exactly

$$m_i=[2^{\ell-i}\nu_i], \tag{5}$$

where $x\mapsto[x]$ is the usual rounding function.

Note that v_i (resp. ν_i) for all i satisfy the conservation law

$$\sum_{1\le i\le\ell}n_iv_i=nv_t\quad\text{ or }\quad\sum_{1\le i\le\ell}2^{1-i}\nu_i=\nu. \tag{6}$$

According to [10, Sec 3], the probability that the i-th level will be proved in one step equals to the share of

surjections $\{1,2,\dots,n_i\} \twoheadrightarrow \{1,2,\dots,m_i\}$ among all maps $\{1,2,\dots,n_i\} \to \{1,2,\dots,m_i\}$

$$p_i = \frac{n_i! S(m_i,n_i)}{n_i^{m_i}}. \tag{7}$$

Here $S(m_i,n_i)$ are Stirling numbers of the second kind [12]. Note that $n_\ell = 1$, so for $m_\ell \geq 1$ we have $p_\ell = S(m_\ell,1) = 1$.

The probability that the whole tree will be built in ℓ steps is

$$p = \prod_{i=1}^{\ell-1} p_i = \prod_{i=1}^{\ell-1} \frac{n_i! S(m_i,n_i)}{n_i^{m_i}}. \tag{8}$$

Now we are ready to formulate computational problems:

Problem 1. Let ℓ and $\nu = v_t/v_p$ be fixed. Find the maximum of the probability (8) as a function of ν_i satisfying relation (6).

The above problem may take a significant time to check all suitable values of ν_i. But our numerical calculations show that it is sufficient to consider only cases when these values form a geometric progression with a common ratio z. From relation (6) we get

$$\nu_i = \nu \frac{1-z/2}{1-(z/2)^\ell} z^{i-1}, \qquad 1 \leq i \leq \ell. \tag{9}$$

To have $\nu_\ell \geq 1$, it is natural to consider $\nu \geq 1/(1-z/2)$.

Alternatively, if we suppose that $\nu_\ell = 1$ (the minimal possible number of provers to built a root-proof in one step) and other ν_i form a geometric progression, we get

$$\nu_i = (\nu - 2^{1-\ell}) \frac{1-z/2}{1-(z/2)^{\ell-1}} z^{i-1}, \qquad 1 \leq i < \ell. \tag{10}$$

Problem 2. Let ℓ and $\nu = v_t/v_p$ be fixed. Find the maximum of the probability

$$p = \prod_{i=1}^{\ell-1} \frac{2^{\ell-i}! \cdot S([2^{\ell-i}\nu_i], 2^{\ell-i})}{2^{(\ell-i)[2^{\ell-i}\nu_i]}}. \tag{11}$$

as a function of z, where ν_i are given by (9) or (10).

Interpolation using Stirling numbers for non-integer arguments: Flajolet and Prodinger in [13], applying the Cauchy's coefficient formula to the generating function, obtained generalizations of the Stirling numbers of the second kind for the complex arguments. In the case where only the first argument x is not an integer (with $\Re x > 0$), the usual binomial formula remains valid:

$$S(x,k) = \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} (-1)^{k-j} j^x. \tag{12}$$

We can apply this formula to avoid rounding in (5). The modified formula for probability will be

$$p = 2^{-\sum_{i=1}^{\ell-1}(\ell-i)2^{\ell-i}\nu_i} \cdot \prod_{i=1}^{\ell-1} \sum_{j=1}^{2^{\ell-i}} \binom{2^{\ell-i}}{j} (-1)^j j^{2^{\ell-i}\nu_i}.$$

In the case when ν_i are given by a geometric progression (9),

$$p = 2^{-2^{\ell-1}\nu \frac{(\ell-1)-\ell(z/2)+(z/2)^\ell}{(1-z/2)(1-(z/2)^\ell)}} \cdot \prod_{i=1}^{\ell-1} \sum_{j=1}^{2^{\ell-i}} \binom{2^{\ell-i}}{j} (-1)^j j^{2^{\ell-i}\nu_i}. \tag{13}$$

B. Explicit calculations

Here we present results of calculations related to Problem 2 using Wolfram Mathematica.

Formulas (11), (9) are implemented in the listing

```
PrTr[l_,nu_,z_] := Module[{n, m},
  n = Table[BitShiftLeft[1,l-i],{i,1,l-1}];
  m = Table[ Round[n[[i]] * nu *
   z^(i-1)*(1-z/2)/(1-(z/2)^l)],{i,1,l-1}];
  Product[Factorial[n[[i]]] *
   StirlingS2[m[[i]], n[[i]]] /
   BitShiftLeft[1,(l-i) * m[[i]]],
   {i,1,l-1}]];
```

Application of the rounding function switches further calculations to infinite-precision arithmetics.

Formula (13) is implemented in the listing

```
PrTrC[l_,nu_,z_] := Module[{n},
  n = Table[BitShiftLeft[1,l-i],{i,1,l-1}];
  Product[Sum[Binomial[n[[i]],j] *
   (-1)^j * j^
   (n[[i]] * nu * z^(i-1) *
   (1 - z/2) / (1 - (z/2)^l)),
   {j,n[[i]]}], {i,1,l-1}] /
  2^ (2^(l-1) * nu *
   ((l-1) - l*z/2 + (z/2)^l) /
    ((1 - z/2)(1 - (z/2)^l)))];
```

It assumes arbitrary–precision arithmetics.

Please note that using the formula (10) instead of (9) results in a very similar picture.

Our preferences for application is constructing of a proof tree with $\ell = 9$ levels. We consider another value $\ell = 4$ to compare results.

Note that for fixed ℓ and ν, the first realization of the function $p(z)$ is a piecewise continuous locally constant (because of the rounding function inside). The second realization of $p(z)$ looks like a suitable smooth approximation of it. So, in what follows we prefer to use this realization. The function $p(z)$ reaches the global maximum in the interval inside $[0,1]$ (closer to 1). See Figure 2 and Figure 3 as examples.

Dependencies of $p_{z=1.25}, p_{z=1}, p_{z=.85}, p_{z=.65}$ and $\max_z p$ on ν for $\ell = 4$ levels are shown in Figure 4. Dependencies of $p_{z=1.2}, p_{z=1}, p_{z=.95}, p_{z=.9}$ and $\max_z p$ on ν for $\ell = 9$ levels are shown in Figure 5. All functions are monotone increasing.

It is convenient to consider the relative forms, i.e. differences with the incentive-free case:

$$\Delta p|_{z_1} := p|_{z=z_1} - p|_{z=1}, \qquad \Delta \max_z p := \max_z p - p|_{z=1}.$$

Dependencies of the differences $\Delta p|_{.692}, \Delta p|_{.72}, \Delta p|_{.99}$ and $\Delta \max_z p$ on ν for $\ell = 4$ are shown in Figure 6. Dependencies

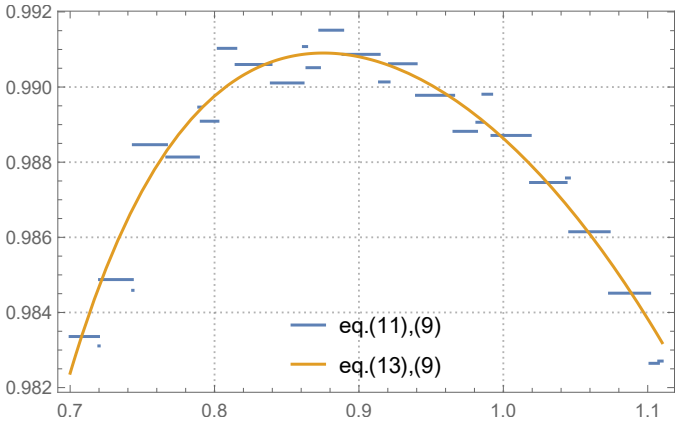

Fig. 2. Dependencies of p on z for $\ell = 4$ and $\nu = 12$

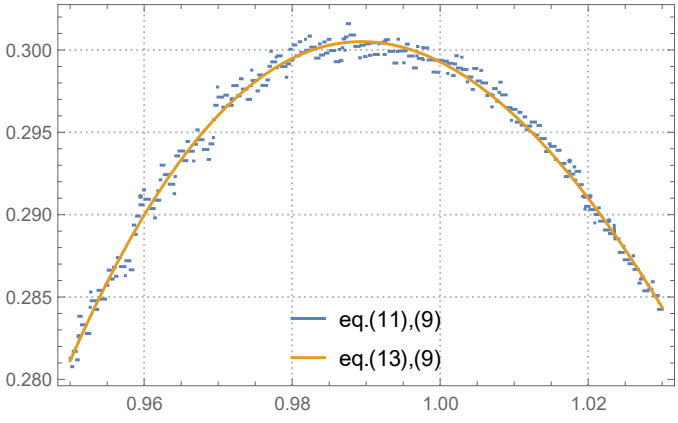

Fig. 3. Dependencies of p on z for $\ell = 9$ and $\nu = 12$

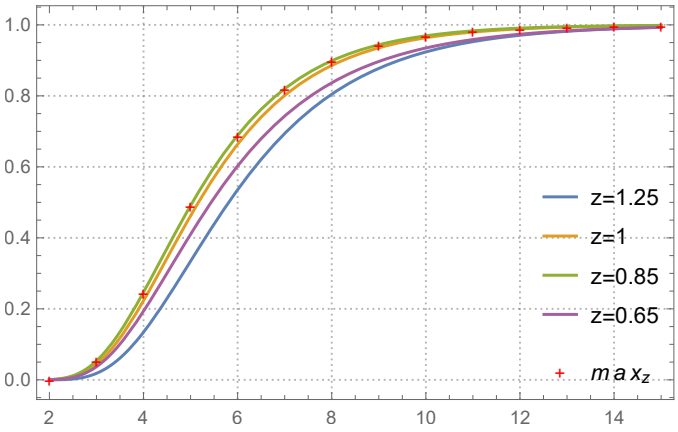

Fig. 4. Dependencies of p for different values of z and $\max_z p$ on ν for $\ell = 4$

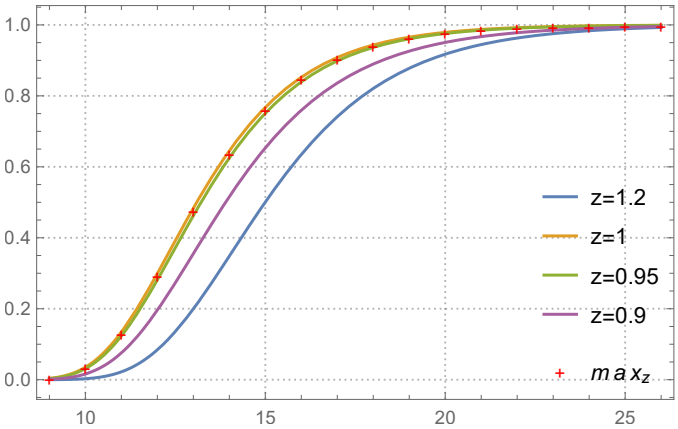

Fig. 5. Dependencies of p for different values of z and $\max_z p$ on ν for $\ell = 9$

of the differences $\Delta p|_{.9784}$, $\Delta p|_{.9794}$, $\Delta p|_{.99}$ and $\Delta \max_z p$ on ν for $\ell = 9$ are shown in Figure 7. These functions come in three forms: positive, negative, and both. It can be seen in both figures that the maximum points fall on the same curve with good accuracy.

One can see that $\Delta \max_z p < .029$ for $\ell = 4$ and $\Delta \max_z p < .0014$ for $\ell = 9$. Moreover, for $\ell = 9$ to reach a probability $p > .9$, we have to assume $\nu \gtrapprox 17$, and in this case $\Delta \max_z p < .0005$ whence we can conclude that at least in the second case changes in incentives for different levels within the range of reaching the maximum practically do not affect the probability of building a tree.

Dependencies of $\operatorname{argmax}_z p$ on ν for $\ell = 4, 5, 6, 9$ are shown in Figure 8. This function is monotone increasing in both arguments ν and ℓ. For large values of ℓ, it is nearly a constant in ν, e.g. for $\ell = 9$, $\operatorname{argmax}_z p \approx .99$.

V. Conclusion

The results of this paper give us a tool to, in a sense, control the behavior of provers to provide stable block generation with sufficiently high throughput in sidechains. We showed that the optimal incentive policy depends on the model we consider - only one sidechain or a lot of ones with provers' ability to switch between them.

For the model from Section III, we give some tools how to assign incentives for proofs in two cases: when this incentives may take only fixed values, and in the case when provers may choose them from some intervals.

For the model from Section IV, we show that here for our conditions there is no sense to change incentive prices for different levels of a proof tree.

The next question is how to simulate provers in more complex, and at the same time, the most practically significant case of optimal incentivizing for generating multiple numbers of proof trees.

Acknowledgment

This work was funded by Input Output (iohk.io), Horizen (horizen.io) and was supported in part by the National Research Foundation of Ukraine under Grant 2020.01/0351.

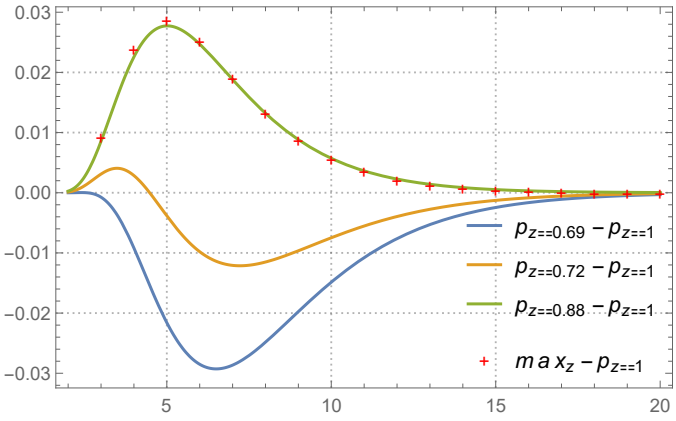

Fig. 6. Dependencies of $\Delta p|_{.692}$, $\Delta p|_{.72}$, $\Delta p|_{.99}$ and $\Delta \max_z p$ on ν for $\ell = 4$

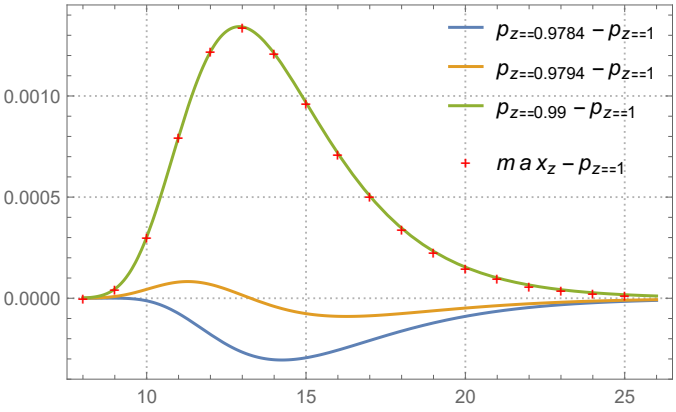

Fig. 7. Dependencies of $\Delta p|_{.9784}$, $\Delta p|_{.9794}$, $\Delta p|_{.99}$, $\Delta \max_z p$ on ν for $\ell = 9$

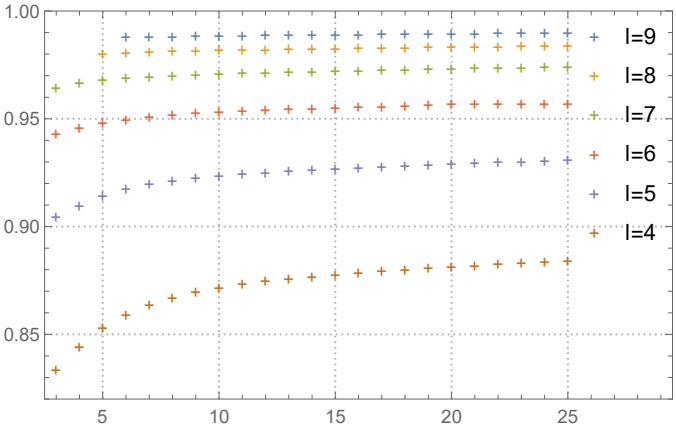

Fig. 8. Dependencies of $\mathrm{argmax}_z\, p$ on ν for $\ell = 4, 5, 6, 7, 8, 9$

References

[1] "Rootstock: smart contracts on bitcoin network," 2018, https://www.rsk.co.

[2] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," 2014, https://blockstream.com/sidechains.pdf.

[3] A. Garoffolo and R. Viglione, "Sidechains: Decoupled consensus between chains," 2018, arXiv:1812.05441.

[4] A. Kiayias and D. Zindros, "Proof-of-work sidechains," 2018, https://ia.cr/2018/1048.

[5] A. Garoffolo, D. Kaidalov, and R. Oliynykov, "Zendoo: a zk-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains," 2020, arXiv:2002.01847.

[6] B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2018, pp. 66–98.

[7] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology - EUROCRYPT 2015, Part II*, ser. - Lecture Notes in Computer Science, vol. 9057. Springer, Berlin, Heidelberg, 2015, pp. 281–310, https://doi.org/10.1007/978-3-662-46803-6_10.

[8] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von Neumann architecture," 2013, https://ia.cr/2013/879.

[9] S. Bowe and A. Gabizon, "Making Groth's zk-SNARK simulation extractable in the random oracle model," 2018, https://ia.cr/2018/187.

[10] Y. Bespalov, A. Garoffolo, L. Kovalchuk, H. Nelasa, and R. Oliynykov, "Probability models of distributed proof generation for zk-SNARK-based blockchains," *Mathematica*, vol. 9, no. 23, p. 3016, 2021, https://www.mdpi.com/2227-7390/9/23/3016 https://doi.org/10.3390/math9233016.

[11] ——, "Game-theoretic view on decentralized proof generation in zk-SNARK based sidechains," in *Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021)*, ser. - CEUR Workshop Proceedings, 2021, vol. 2923, pp. 47–59, http://ceur-ws.org/Vol-2923/.

[12] K. N. Boyadzhiev, "Close encounters with the Stirling numbers of the second kind," The Mathematics Magazine, 85, No. 4, (October 2012), 252-266, 2018, arXiv:1806.09468.

[13] P. Flajolet and H. Prodinger, "On stirling numbers for complex arguments and hankel contours," *SIAM J. Discrete Math.*, vol. 12, no. 2, pp. 155–159, 1999.

[14] R. B. Myerson, *Game Theory: Analysis of Conflict*. Harvard University Press, 1997.

[15] N. Ham, "Notions of symmetry for finite strategic-form games," 2013, arXiv:1311.4766.