Tullock Contests in the Wild: Applications in Blockchains

PRANAV GARIMIDI
a16z crypto
and
MICHAEL NEUDER
Ethereum Foundation
and
TIM ROUGHGARDEN
Columbia University, a16z crypto

This letter shows how Tullock contests—a class of all-pay auctions with proportional allocation rules—can be used to model and reason about several blockchain settings. We review the fundamentals of Tullock contests and their connections to potential games. We discuss why certain properties of Tullock contests, such as sybil-proofness and compatibility with "decentralization," have made them common in blockchain applications. We illustrate how Tullock contests naturally arise in proof-of-work and proof-of-stake blockchain protocols, and are an attractive design for emerging marketplaces for blockspace and succinct proofs.

Categories and Subject Descriptors: K.4.4 [**Computers and Society**]: Electronic Commerce; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Design, Economics, Documentation

Additional Key Words and Phrases: Tullock contests, Blockchain, Consensus mechanisms, Proof-of-Work, Proof-of-Stake, Prover markets

1. INTRODUCTION

Tullock contests have long been a valuable model for studying economic scenarios. Tullock explored the concept in [Tullock 1975; 1980] to examine the outcome of political elections. Since then, the model has found applications ranging from analyzing R&D investments in patent races [Baye and Hoppe 2003] to understanding how much sports teams spend on players [Dietl et al. 2008]. The key feature of a Tullock contest is that competitors must invest costly effort before the winner is chosen probabilistically, with winning probabilities proportional to investments. Crucially, these contests have an all-pay nature—the losing parties pay for their investments without any gain—and so participants must hedge against losing when choosing how much to invest. As a result, multiple parties invest at equilibrium and the most efficient party will not always win the context. In many blockchain-related allocation problems, there is: (i) an explicit goal of multiple active participants; and/or (ii) the possibility of sybil attacks (with one participant masquerading as many). Tullock contests are an attractive design in such settings.

Authors' addresses:
pgarimidi@a16z.com, michael.neuder@ethereum.org, tim.roughgarden@gmail.com

In this article, we first recap the basics of Tullock contests and their properties, and then proceed to a few key examples of their applications in blockchain-related settings. We highlight applications in which market designers prefer Tullock contests over more efficient alternatives due to their non-winner-take-all equilibria.

2. TULLOCK CONTESTS

A Tullock contest is an all-pay auction in which every participant (or "player") makes a costly investment, but only one wins a prize. Formally:

—There are n players.

—Each player $i \in \{1, 2, ..., n\}$ chooses an investment level $b_i \geq 0$ and pays b_i.

—The probability of winning the prize is proportional to one's investment. Thus, for investments $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, player i's probability of winning (a.k.a. their *allocation*) is

$$x_i(\mathbf{b}) = \frac{b_i}{\sum_{j=1}^{n} b_j}. \tag{1}$$

In our analysis of Tullock contests, we will always assume that players have quasi-linear payoffs $U_i(x_i(\mathbf{b})) - b_i$, where U_i is an increasing, differentiable, and (weakly) concave utility function that satisfies $U_i(0) = 0$.[1]

One can interpret the outcome of a Tullock contest as allocating one unit of a divisible good to the participants at a common per-unit price (namely, the price $\sum_{i=1}^{n} b_i$). In particular, a Tullock contest is *sybil-proof*, meaning that no participant can benefit from participating under multiple identities (one cannot do better than splitting one's single-identity equilibrium bid arbitrarily over one's multiple identities).

We will see how multiple different settings can be interpreted as instantiations of this model. In this article, we consider only the complete information setting in which agents' utility functions are common knowledge, and focus on pure Nash equilibria. This is the setting studied in the relevant blockchain literature. Extending the analysis to the incomplete information setting and Bayesian-Nash equilibria, perhaps by building on the techniques in [Syrgkanis and Tardos 2013] and [Caragiannis and Voudouris 2016], is an interesting direction for future work.

2.1 Potential Games and Equilibrium Characterization

A key observation that enables the equilibrium analysis of Tullock contests is that they are *potential games*. In a potential game, the equilibrium outcomes correspond to the global maximizers of a suitably defined potential function. [Johari and Tsitsiklis 2004] give such a potential function characterizing the equilibria of Tullock contests. This function is defined on allocation vectors (rather than bid vectors), and characterizes the allocation vectors that are induced (via (1)) by equilibrium bid vectors.

THEOREM 2.1 EQUILIBRIUM CHARACTERIZATION [JOHARI AND TSITSIKLIS 2004]. *Allocations* $\mathbf{x}$ *that correspond to a pure Nash equilibrium (PNE) of a Tullock contest*

[1]We often consider the special case of linear utility functions, in which, for each i, $U_i(x_i(\mathbf{b})) = v_i \cdot x_i(\mathbf{b})$ for some value $v_i > 0$.

are exactly the solutions to

$$\max \sum_{i=1}^{n} \widetilde{U}_i(x_i)$$

subject to $\sum_{i=1}^{n} x_i = 1$ *and* $x_i \geq 0$ *for all* $i \in \{1, 2, ..., n\}$*, where*

$$\widetilde{U}_i(x_i) = (1 - x_i)U_i(x_i) + \int_0^{x_i} U_i(y)dy.$$

Proof Sketch. Each agent's best response function can be rewritten as a function of the allocation vector $\mathbf{x}$ and the total sum of bids B. Calculations then show that the first-order conditions of the optimization problem match the first-order conditions of the best-response problem faced by players. By our assumptions on players' utility functions (such as concavity), these conditions also characterize the global solutions to the optimization problem and players' best responses.

Because each utility function U_i is weakly concave, each modified utility function $\widetilde{U}_i$ is strictly concave. It follows that the potential function above has a unique maximum. While this only implies a unique equilibrium allocation, rewriting the agents' best-response bids as a function of their allocations further shows that equilibrium bids are unique. This observation gives us the following corollary.

Corollary 2.2. *In a Tullock contest, there exists a unique PNE.*

For blockchain-related applications, it will be useful to extend the basic model of Tullock costs to include player-specific cost multipliers. Consider a variation of the Tullock contest model in which player i pays $c_i \cdot b_i$ to bid b_i—in effect, some players can generate a given amount of investment more efficiently than others. This setting is equivalent to the classic Tullock contest setup in which each player i has the utility function $U_i(x)/c_i$.

Lemma 2.3. *Let* $\mathcal{T}(\mathbf{U}, \mathbf{c})$ *be a Tullock contest in which players have utility functions* $\mathbf{U}$ *and bidding costs* $\mathbf{c}$*. Then* $\widetilde{\mathbf{b}}$ *is an equilibrium bid vector for* $\mathcal{T}(\mathbf{U}, \mathbf{c})$ *if and only if* $\widetilde{\mathbf{b}}$ *is an equilibrium bid vector for* $\mathcal{T}(\mathbf{U}/\mathbf{c}, \mathbf{e})$ *where* $\mathbf{e}$ *denotes the all-ones vector.*

We omit the straightforward proof.

2.2 Equilibrium analysis

Tullock contests have two defining characteristics: (i) the all-pay nature in which players pay their bid even when they lose, and (ii) the proportional allocation. Combined, these characteristics imply that agents have strictly diminishing returns on investment (even with linear utility functions). For this reason, equilibria of Tullock contests are generally oligopolistic outcomes. For example, when agents have linear utility functions, the agents with larger utility functions receive higher allocations at equilibrium, but the allocation is split over two or more agents. It follows that Tullock contests do not generally implement fully efficient equilibria. [Johari and Tsitsiklis 2004] quantify equilibrium inefficiency in Tullock contests and show that the worst-case "price of anarchy" is precisely 3/4—the sum of agent utilities at equilibrium in a Tullock contest is always at least 75% of the maximum possible, and this bound is tight in the worst case.

Theorem 2.4 Efficiency of Tullock contests [Johari and Tsitsiklis 2004]. *In a Tullock contest, let* $\mathbf{d}^*$ *be the welfare-maximizing allocation and* $\mathbf{d}$ *the allocation corresponding to the unique PNE. Then:*

$$\sum_i U_i(d_i) \geq \frac{3}{4} \sum_i U_i(d_i^*).$$

Furthermore, this bound is tight: for every $\epsilon > 0$*, there exist* n *and linear utility functions* $U_1, U_2, \ldots, U_n$ *such that*

$$\sum_i U_i(d_i) \leq \left(\frac{3}{4} + \epsilon\right) \sum_i U_i(d_i^*).$$

For an alternative proof to the one in [Johari and Tsitsiklis 2004], see Section 3 of [Roughgarden 2006].

Thus, while Tullock contests generally result in inefficient equilibria, the efficiency loss is bounded. This efficiency loss may be acceptable (or unavoidable) if other considerations, such as encouraging participation to avoid centralization, are paramount.

The exact realization of the "payments" in a Tullock contest is application-dependent. For example, in Section 3.1, we discuss proof-of-work blockchain protocols, in which the payments correspond to investments in hardware and electricity. In Section 3.3, by contrast, we discuss blockspace auctions in which payments represent direct monetary transfers to an auctioneer. The efficiency objective function in Theorem 2.4 captures the utilities of players, independent of payments.

3. BLOCKCHAIN APPLICATIONS

We now turn our attention to the relevance of Tullock contests for blockchain protocols. The following simple model of a blockchain protocol suffices for this article: an ever-growing sequence of transactions, with a "leader" periodically chosen to append a new block of transactions. The leader is typically drawn from the set of physical machines running the protocol (generally called "miners" in a proof-of-work protocol or "validators" in a proof-of-stake protocol). Two common goals for "decentralized" blockchain protocols are: (i) permissionlessness, meaning that anyone should be able to participate in the protocol as a miner or validator; and (ii) no one or small group of participants should have undue control over the blockchain's transaction sequence. In part with these goals in mind, the Bitcoin protocol (among others) chooses leaders using a "proof-of-work" mechanism, repeatedly choosing a leader with probability proportional to miners' hashrates. The Ethereum protocol (among others) uses a proof-of-stake mechanism to choose leaders, with each leader chosen from the validator set with probability proportional to the amount of cryptocurrency that they have staked (i.e., locked in the blockchain protocol). We next show that these mechanisms are equivalent to Tullock contests, allowing us to use Theorem 2.1 to characterize the relative influence of different parties in these protocols. We then examine how these same ideas have been used to inspire mechanisms for new blockchain applications to achieve similar goals of having a decentralized set of participants.

3.1 Proof-of-Work Protocols

The Bitcoin protocol is "permissionless" in the sense that any party can become a "miner." Miners compete to produce new blocks of transactions by repeatedly hashing candidate strings until they find an input with a sufficiently small output value. The first miner to publish such an input is rewarded with newly created Bitcoin (currently 3.125 BTC, worth over 300,000 USD at this time of writing) and also appends a new block of recent transactions to the running transaction sequence. The threshold for "sufficiently small" is adjusted, as a function of the amount of participating hashrate, so that a new block is produced every ten minutes on average. Miners can invest in hardware and electricity to increase their hashrate and become more competitive as block producers. However, while anyone is free to make investments, only those parties that can operate the most efficiently (e.g., with access to cheap electricity) find it profitable to stay active in the network; otherwise, the cost of running the hardware exceeds the rewards they earn. Quantifying the "decentralization" of a proof-of-work protocol can then be formalized as the following question: what is the distribution of miners' hashrates at equilibrium, as a function of miners' relative costs of operation? [Arnosti and Weinberg 2022] answers this question using the framework of Tullock contests.

Model:

—There is a block reward r. (E.g., 3.125 BTC.)

—Agent i's utility is $U_i(x_i) = rx_i$. (I.e., agents are risk-neutral.)

—Agent i's cost of operating q_i units of hardware is $c_i q_i$. (E.g., reflecting hardware depreciation and electricity costs.)

—The allocation is given by $x_i = q_i / \sum_j q_j$. (The definition of proof-of-work leader selection.)

As shown in Lemma 2.3, this model is equivalent to standard Tullock contests after a simple transformation. This model implicitly assumes that hardware is homogeneous, but agents have different acquisition and/or operating costs. Equivalently, agents could have access to differing quality hardware at the same costs. The block rewards are split pro-rata according to agents' hardware, as is standard in Tullock contests. [Arnosti and Weinberg 2022] characterize the equilibrium in this setting as follows: With respect to fixed agent costs $c_1, c_2, \ldots, c_n$, define the function

$$X(c) = \sum_i \max(1 - c_i/c, 0)$$

and let c^* be the solution to $X(c^*) = 1$. Then,

Theorem 3.1 Proof-of-Work Equilibrium [Arnosti and Weinberg 2022]. *At the unique PNE of the proof-of-work Tullock contest, miners make investments* $q_i = \frac{1}{c_i}\max(1 - c_i/c^*, 0)$*, resulting in allocations* $x_i(\mathbf{q}) = \max(1 - c_i/c^*, 0)$.

Thus, for a given a cost vector, there is a threshold cost c^* such that agents with costs above the threshold do not participate at equilibrium. The following corollary provides one interpretation of this equilibrium.

Corollary 3.2 [Arnosti and Weinberg 2022]. *If miner i participates at equilibrium ($q_i > 0$), then for all j, $x_i(\mathbf{q}) \geq 1 - \frac{c_i}{c_j}$.*

This corollary demonstrates that relatively small differences in the cost of mining can result in highly concentrated allocations at equilibrium—the "natural oligopoly" referred to in the paper's title. For example, with n large and $c_i = i/(i+1)$ for each i, one can calculate $c_7 < c^* \approx .88 < c_8$ [Arnosti and Weinberg 2022]. Because $c_1 = 1/2$ and $c_7 = 7/8$, we have $x_1 \geq 3/7$, representing a substantial amount of power for a single miner.

3.2 Proof-of-Stake Protocols

In a proof-of-stake protocol (including the Ethereum protocol and many others), validators lock up capital (a.k.a. stake), and each leader is chosen with probability proportional to stake. Similarly to the Bitcoin protocol, leaders are responsible for producing blocks that append recent transactions to the running transaction sequence and are also rewarded with newly minted cryptocurrency. In a proof-of-stake protocol, the costly investment is the opportunity cost of locking up stake (as opposed to, for example, investing in U.S. treasury bills). Thus, the analysis in Section 3.1 carries over with this new interpretation, with validators choosing how much stake to invest rather than how much hardware to operate.[2,3]

An additional complication in blockchain protocols with a mature and Turing-complete smart contract layer, including the Ethereum and Solana protocols, is that validators with different levels of sophistication can earn vastly different rewards from block production. On top of standard block rewards and transaction fees, block producers can earn substantial revenue from "maximal extractable value (MEV)." Roughly, MEV refers to rents extracted by a block producer on account of their temporary monopoly power over transaction sequencing (e.g., deciding the order in which trades are executed on a financial exchange) [Daian et al. 2020]. Because some validators know about more pending transactions than others (e.g., due to business agreements with power users) and some validators are better at assembling high-MEV blocks than others (e.g., due to more computational power or better algorithms for exploring the space of possible blocks), some validators can earn much more revenue from a given block production opportunity than others.

To capture the validator heterogeneity introduced by MEV, we consider the following model:

Model:

—There is a base reward r.

—Agent i's utility for being chosen as a block producer is $U_i(x_i) = \mu_i \cdot rx_i$, with μ_i representing the agent's acumen at extracting MEV from the current block production opportunity.

—Agent i chooses an amount π_i of stake and incurs a per-unit cost of c.

[2]This analysis does not consider any returns validators earn from the stake, itself, appreciating.
[3]In practice, the majority of stake controlled by validators has been delegated to them by other parties (and so validators primarily pay operational rather than capital costs). For simplicity, in this article we'll ignore the possibility of delegated stake.

—Allocations are given by $x_i = \pi_i / \sum_j \pi_j$. (The definition of proof-of-stake leader selection.)

For example, a validator i with $\mu_i = 1$ would collect only the base reward for block production, while a validator with $\mu_i = 2$ would collect double the base reward (presumably on account of better MEV extraction). This model is mathematically equivalent to that in the previous section (see also Lemma 2.3), but the change in notation is helpful to indicate different interpretations of this model. [Bahrani et al. 2024] analyze pure Nash equilibria in this setting as a function of the relative sophistication of different validators at block production (i.e., of the μ_i's). They call a validator set (γ, k)-*competitive* if $\mu_{k+1} \geq \gamma \cdot \mu_1$; in words, at least k validators have a reward multiple that is at least a γ fraction of the largest multiple. In the context of block production, this means there are at least k parties capable of producing a block with value at least γ times that produced by the most sophisticated validator. (Larger values of k and γ correspond to "more competitive" validator sets.) [Bahrani et al. 2024] use this parameterization to upper bound the maximum equilibrium allocation of any individual validator.

Theorem 3.3 Proof-of-Stake Equilibrium [Bahrani et al. 2024]. *For every (γ, k)-competitive block producer set with $\gamma \in [0, 1]$ and $k \geq 1$, the unique PNE allocations $\mathbf{x}$ satisfy $x_i \leq 1 - \frac{\gamma k}{k+\gamma}$ for all i.*

For example, with 10 validators at least 90% as sophisticated as the most sophisticated validator, no individual validator will control more than 17.5% of the stake at equilibrium.

If one or a small number of validators are substantially more sophisticated than the rest, how can one avoid centralization (i.e., stake concentration at equilibrium)? Modern block production for the Ethereum protocol is based on *proposer-builder separation (PBS)*, a system in which validators can outsource block production to a specialized set of third parties called *block builders*. The goal of PBS is to preserve decentralization (with many validators participating at equilibrium) by confining centralization to the set of block builders.[4]

[Bahrani et al. 2024] extend their analysis to incorporate PBS, as follows. Under PBS, for each block production opportunity (called a "slot"), the corresponding leader runs a first-price auction in which block builders compete to construct the most valuable block and submit bids for their block to be chosen by the leader. (Thus, the item being sold in the auction is the current block production opportunity; the seller is the validator that was chosen as the current leader; and the bidders are the block builders.) This auction smooths out the differences in sophistication between validators, as every validator's value for being chosen as leader is now typically just the (validator-independent) revenue that they can collect as an auctioneer. Theorem 5.1 of [Bahrani et al. 2024] formally captures the effect of PBS in the above model by showing that, with PBS and at least l competitive builders, the ratio in the expected rewards obtained by any two validators for a given block production opportunity is $1 + O(1/\log l)$. That is, for large l, a block production

[4]Because builders do not participate directly in the blockchain protocol and its decisions, builder centralization is generally viewed as less concerning than validator centralization.

opportunity is almost equally valuable to all validators.[5]

This result implies that, with PBS and in the notation of our model of investments by proof-of-stake validators, the μ_i's of any two validators differ by a factor of $1 + O(1/\log l)$. Plugging this into the equilibrium analysis of Theorem 3.3 shows that, under our idealized version of PBS with n validators and l builders,

$$x_i = \frac{1}{n} + O\left(\frac{1}{\log l}\right)$$

for each validator i. That is, in this model, PBS does indeed guarantee decentralization in the validator set, despite heterogeneous validator sophistication.[6]

3.3 A Market for Block Production Rights

We now switch from analyzing the equilibria of currently implemented protocols to exploring how Tullock contests have been proposed for use in future mechanisms. We start by showing how Tullock contests can address some of the drawbacks of PBS. As discussed above, a key part to PBS helping reduce centralization in the validator set is the existence of a competitive block builder set. These builders specialize in constructing valuable blocks through many means such as unique trading strategies, business relationships, sophisticated block-building algorithms, and better networking infrastructure. The main downside to PBS is the set of builders may become centralized. In practice, it may be that only a small number of entities can consistently win block-building auctions and can reinvest those profits into gaining even more market share. At the time of writing, 96% of Ethereum blocks are built by just three different entities.[7] While block validation in Ethereum remains decentralized, builders have tremendous power in deciding which transactions are included in the running transaction sequence.

There have been discussions of alternative market structures to alleviate builders' market power and encourage participation by a larger set of builders. One widely-discussed idea is "execution tickets" [Drake and Neuder 2023], in which block production rights are allocated by lottery rather than a first-price auction. The idea is that a blockchain protocol would set a ticket price, with builders purchasing as many tickets as they wish. For every slot, the protocol would select one of the tickets uniformly at random, and the ticket owner would be granted exclusive block production rights for that slot. Payments are made up-front, and are not refunded even if the purchaser is never selected as a block producer. For the sake of this analysis, we assume that block production rights cannot be resold once the lottery winners are revealed.[8] [Neuder et al. 2024] describe how the non-resale setting is

[5]This result allows validators to build their own blocks as before (i.e., to ignore all the blocks submitted by builders) but assumes that the builders, as specialized parties, are at least as proficient at block-building as the validator. More precisely, each builder draws their value for a block production opportunity from a distribution that satisfies the monotone hazard rate condition and also first-order stochastically dominates the distribution of the validator's value for the block production opportunity.

[6]In practice, the stake distribution is also influenced by other factors, such as the different yields offered by validators to those who delegate stake to them.

[7]See `https://www.relayscan.io/` for real-time data on the Ethereum block-builder distribution.

[8]For a model that considers how this analysis changes when resale is permitted, see [Pai and

mathematically equivalent to a Tullock context, with agents' values for winning the lottery their values for block production rights and the ticket price set to \$1.

Model:

—Agent i has value v_i for block production rights, and (risk-neutral) utility function $U_i(x_i) = v_i x_i$.
—Agent i purchases b_i tickets at a cost of b_i.
—Allocations are given by $x_i = b_i / \sum_j b_j$.

In the context of execution tickets, agents' payments are direct transfers rather than implied capital or operating costs.

Theorem 2.1 can be invoked again here, replacing the μ_i's with v_i's. Define the function

$$F(x) = \sum_{i=1}^{n} \max\left(1 - \frac{x}{v_i}, 0\right)$$

and let v^* satisfy $F(v^*) = 1$. Then at the (unique) equilibrium bid vector $\mathbf{b}$, the corresponding allocations $\mathbf{x}$ satisfy

$$x_i = \max\left(1 - \frac{v^*}{v_i}, 0\right)$$

for every i.

Because execution tickets are effectively an implementation of a Tullock contest, multiple participants invest at equilibrium and have a non-zero probability of winning the block production rights for a slot. This contrasts with the "winner-take-all" nature of first-price auctions (as used in today's PBS), in which the participant with the highest value wins with certainty at equilibrium. More generally, the execution tickets design is applicable to any domain in which block production rights must be allocated, including "layer-two" protocols and shared sequencers.[9]

3.4 Proof Marketplaces

For our last example, we turn to the emerging application of marketplaces for proofs, and specifically for SNARKs (i.e., succinct noninteractive arguments of knowledge). The point of a SNARK is to enable anyone to quickly verify that a computation was carried out correctly (without redoing the computation). SNARKs are useful for a number blockchain-related applications. For example, one increasingly common architecture for a blockchain protocol is for transaction processing (and corresponding SNARK generation) to be carried out by a small number specialized "provers" (somewhat analogous to the role of builders in PBS), with the (decentralized) set of validators responsible only for SNARK verification. SNARKs are computationally intensive to produce, and proof marketplaces are designed to coordinate the clearing of the market for SNARK generation.

Proof marketplaces are two-sided markets, with agents who have demand for proofs on one side and provers on the other side. For simplicity, we consider here

Resnick 2024].

[9]For example, Espresso Systems has proposed an execution tickets-style design for a shared sequencer [Bünz et al. 2024].

a setting in which a single party demands a proof with multiple provers competing to supply it. One approach to this procurement problem would be to run a reverse first- or second-price auction. This approach runs the risk of a winner-take-all outcome, with only the most efficient prover ever producing proofs. Inspired by the successes of Tullock contests in blockchain-related applications discussed above, [Roy et al. 2024] propose a similar mechanism for proof marketplaces to address these centralization concerns. Below, we give a variation of their mechanism:

Model:

—The auctioneer (representing the buyer) posts a reward r for computing a proof ϕ.
—Each prover has a cost c_i for computing ϕ.
—Each prover submits a nonrefundable bid b_i, paid up front.[10]
—Prover i^* is randomly selected with probability $x_i(\mathbf{b}) = b_i / \sum_j b_j$.
—Upon computing and submitting ϕ to the auctioneer (which the auctioneer verifies to be a correct proof), the prover i^* is paid r.

Under this mechanism, agent i's utility is $U_i(x_i) = (r - c_i)x_i$ and thus their profit given a bid vector of $\mathbf{b}$ is

$$(r - c_i)x_i(\mathbf{b}) - b_i.$$

Thus, we get the classic Tullock contest setup in which each agent has a value of $r - c_i$ for winning the lottery and we can again use Theorem 2.1 to calculate the equilibrium. The auctioneer can expect multiple provers to compete provided r is larger than the two smallest c_i's. More generally, if provers have distinct costs, for each $k \in \{1, 2, \ldots, n\}$, there is a corresponding range of rewards for which exactly k provers will participate and receive non-zero allocations. Thus, if prover costs can be treated as common knowledge, a buyer can use the equilibrium characterization of Theorem 2.1 to choose a reward that incentivizes a target level of participation.

This mechanism for proof marketplaces relies on the fact that the auctioneer—perhaps implemented as a smart contract on a blockchain—can easily and programmatically verify the correctness of submitted proofs. The mechanism is applicable more generally to procurement problems in which satisfactory service provision can be easily verified and the auctioneer can credibly commit to paying out the reward upon successful procurement.

4. OPEN QUESTIONS AND FUTURE DIRECTIONS

—**Optimal fairness:** What does it mean for a mechanism to be optimally "fair" or "decentralized"? Is there a framework that micro-founds the optimal allocation of service providers subject to "sufficient decentralization"?
—**Optimality/uniqueness of Tullock contests:** Under a suitable metric of fairness or decentralization, are Tullock contests the best (or unique) mechanism that is fair/decentralized and also sybil-proof?

[10] In a permissionless context in which anyone can participate as a buyer or prover, these payments are burned rather than passed on to the buyer/auctioneer. (Otherwise, an agent might participate as both a buyer and a prover; by submitting a large fake bid, it could collect the reward r along with the payments made by the other provers.)

—**Guaranteed non-negative utility and sybil-proofness:** One drawback of Tullock contests is that the only way for a participant to guarantee itself non-negative utility (no matter what the other players do) is to bid 0. In particular, a participant that chooses its equilibrium bid may suffer negative utility if other participants do not, for whatever reason, choose their equilibrium bids. Is this property unavoidable for mechanisms that are sybil-proof and not winner-take-all?

REFERENCES

Arnosti, N. and Weinberg, S. M. 2022. Bitcoin: A natural oligopoly. *Manag. Sci. 68,* 7, 4755–4771.

Bahrani, M., Garimidi, P., and Roughgarden, T. 2024. Centralization in block-building and proposer-builder separation. In *Financial Cryptography and Data Security*. Springer, 331–349.

Baye, M. R. and Hoppe, H. C. 2003. The strategic equivalence of rent-seeking, innovation, and patent-race games. *Games and economic behavior 44,* 2, 217–226.

Bünz, B., Fisch, B., and Davidson, E. 2024. The espresso market design. `https://hackmd.io/@EspressoSystems/market-design`. Accessed: 2025-07-02.

Caragiannis, I. and Voudouris, A. A. 2016. Welfare guarantees for proportional allocations. *Theory Comput. Syst. 59,* 4, 581–599.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. 2020. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE symposium on security and privacy (SP)*. IEEE, 910–927.

Dietl, H. M., Franck, E., and Lang, M. 2008. Overinvestment in team sports leagues: A contest theory model. *Scottish Journal of Political Economy 55,* 3, 353–368.

Drake, J. and Neuder, M. 2023. Execution tickets. `https://ethresear.ch/t/execution-tickets/17944`. Ethereum Research Blog. Accessed: 2025-07-02.

Johari, R. and Tsitsiklis, J. N. 2004. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research 29,* 3, 407–435.

Neuder, M., Garimidi, P., and Roughgarden, T. 2024. On block-space distribution mechanisms. `https://ethresear.ch/t/on-block-space-distribution-mechanisms/19764`. Accessed: 2025-04-21.

Pai, M. M. and Resnick, M. 2024. Centralization in attester-proposer separation. *CoRR abs/2408.03116*.

Roughgarden, T. 2006. Potential functions and the inefficiency of equilibria. In *Proceedings of the International Congress of Mathematicians (ICM)*. Vol. 3. 1071–1094.

Roy, U., Guibas, J., Pai, M., Kulkarni, K., and Robinson, D. 2024. Succinct network: Prove the world's software. `https://docs.succinct.xyz/whitepapers/succinct-network`. Accessed: 2025-03-05.

Syrgkanis, V. and Tardos, É. 2013. Composable and efficient mechanisms. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds. ACM, 211–220.

Tullock, G. 1975. On the efficient organization of trials. *Kyklos 28,* 4, 745–762.

Tullock, G. 1980. Efficient rent seeking. In *Toward a Theory of the Rent-Seeking Society*, J. M. Buchanan, R. D. Tollison, and G. Tullock, Eds. Texas A & M University Press, College Station, TX, 97–112.