Research Article

The Complexity of Tullock Contests

Yu He[1,2], Fan Yao[3,2], Yang Yu[3], Xiaoyun Qiu[4], Minming Li[1], Haifeng Xu[2]

1. City University of Hong Kong, Hong Kong; 2. University of Chicago, United States; 3. University of Virginia, United States; 4. Northwestern University, United States

This paper studies the algorithmic complexity for computing the pure Nash Equilibrium (PNE) in Tullock Contests. The (possibly heterogeneous) elasticity parameter r_i determines whether a contestant i's cost function is convex, concave or neither. Our core finding is that the domains of r_i governs the complexity for solving Tullock contents.
- When no contestant's r_i lies between $(1, 2]$, we can design an efficient algorithm to compute the pure NE;
- When many r_i values fall within $(1, 2]$, we prove that determining NE existence can not be solved in polynomial time, assuming the Exponential Time Hypothesis (ETH);
- When many r_i values fall within $(1, 2]$, we design a Fully Polynomial-Time Approximation Scheme (FPTAS) to find an ϵ-PNE when an exact PNE exists.
All our algorithms are efficiently implemented for solving large-scale instances, and computational experiments demonstrate their effectiveness even in complex scenarios.

1. Introduction

The Tullock contest, introduced by Gordon Tullock[1], is a fundamental model in economic theory, widely used to analyze competitive environments where participants expend resources to increase their probability of winning a prize. This model has significant implications for decentralized systems, such as blockchain, where participants (e.g., miners in proof-of-work systems) compete for rewards by investing computational resources. The competitive nature of blockchain mining and validation closely aligns with the Tullock contest framework, as both involve strategic decision-making about resource expenditure to maximize the probability of success[2][3][4].

In Bitcoin, for example, miners compete to solve cryptographic puzzles, with the first to do so earning a block reward. The Tullock contest's winner-takes-all nature mirrors this dynamic, where each participant's probability of success is proportional to the resources they allocate[5][6]. This competition introduces economic inefficiencies and challenges related to resource allocation and centralization, which can be rigorously analyzed through the Tullock contest model[7][8]. As network difficulty increases, the marginal gains from additional computational power decrease, leading to centralization of mining power and higher barriers for smaller participants[9].

A key feature of the Tullock contest is the elasticity parameter r_i, which determines the relationship between a participant's resource expenditure and their success probability. Distinct ranges of r_i: $r_i < 1$, $r_i \in [1, 2]$, and $r_i > 2$, correspond to diminishing, moderate, and increasing returns to effort, respectively. These regimes provide a versatile framework for understanding strategic interactions across various domains, including blockchain, economic competition, and platform dynamics.

The elasticity parameter captures diverse dynamics: $r_i < 1$ fosters inclusivity by encouraging smaller investments to yield disproportionately higher returns, a feature relevant for collaborative systems like public goods provision[10] or decentralized logistics[11]. In contrast, $r_i > 2$ leads to "winner-takes-all" scenarios, exemplified by resource concentration in decentralized finance or vehicular edge computing[12][13]. Finally, $r_i \in [1, 2]$ represents balanced dynamics, supporting efficient governance and equitable resource allocation, as studied in blockchain platforms and renewable energy trading[14][15].

1.1. Motivation

The application of the Tullock contest model to blockchain systems is motivated by the need to understand how decentralized competition impacts network security, efficiency, and centralization. In proof-of-work blockchains, participants independently decide how much computational power to expend, creating a competitive environment where resource allocation is critical for securing rewards. Analyzing these decisions through the Tullock contest framework provides insights into miners' strategic behavior and the overall functioning of decentralized networks[16][9].

One key motivation for applying the Tullock contest to blockchain is the analysis of *Nash equilibria* in competitive environments. By modeling miners' efforts in terms of resource expenditure and reward probability, the Tullock framework helps to understand how rational agents optimize their strategies in decentralized systems[17]. Additionally, the framework sheds light on potential inefficiencies and externalities, such as energy consumption, that arise in competitive mining processes[18].

Furthermore, blockchain systems inherently involve decentralized decision-making, where participants (miners or validators) operate autonomously. The Tullock contest model allows for the examination of these independent decisions and how they collectively influence network security and reward distribution. This approach can be extended to study the implications of technological and informational asymmetries, where different participants may have varying access to resources or information, affecting their competitive standing[9].

In decentralized systems such as Bitcoin, mining can be viewed as a rent-seeking contest, where participants compete for a fixed reward. The Tullock contest model facilitates understanding of the trade-offs between resource expenditure and reward probability, providing a theoretical framework for studying centralization tendencies in mining and other blockchain operations[5][9]. For instance, research on Bitcoin mining suggests that as network difficulty increases, resource allocation becomes more inefficient, leading to higher barriers for smaller participants[7].

The elasticity parameter further enables the analysis of key trade-offs in decentralized systems. For example, while $r_i < 1$ encourages inclusivity, $r_i > 2$ results in centralization tendencies, and $r_i \in [1, 2]$ strikes a balance between these extremes. Such insights are vital for understanding how strategic interactions shape resource allocation and reward distribution in blockchain and other competitive systems.

1.2. Related Work

The study of Tullock contests in economic theory has been extensively developed, particularly in the context of *asymmetric contests*, where participants differ in their resources, information, or technologies. These models are highly relevant to decentralized systems like blockchain, where miners or validators differ in computational power, risk preferences, or access to market data.

Klunover[19] explores the role of *asymmetric technologies* in Tullock contests, showing that asymmetries can lead to increased efficiency under certain conditions. This is particularly applicable to blockchain mining, where technological asymmetries among participants (e.g., hardware differences) are common and significantly impact mining outcomes.

Cornes and Hartley[17] study the effects of *asymmetric information* in contests, demonstrating how differences in resources and information levels influence strategic behavior and outcomes. In

doi.org/10.32388/WPJBOT

blockchain systems, such information asymmetries affect miners' competitive strategies, especially with regard to network conditions and protocol changes.

Cornes and Hartley[20] also investigate the impact of *risk aversion* in asymmetric contests, showing that differences in participants' risk preferences alter contest outcomes. This research is relevant for understanding how varying risk tolerances influence miners' participation in blockchain systems, particularly for smaller mining operations with less computational power.

Further, Fu, Lu, and Zhang[21] extend the Tullock contest framework to examine optimal contest designs under conditions of asymmetric entry and information disclosure. Their findings offer valuable implications for the design of blockchain governance models, where participants' power and influence often vary based on their computational resources or staked assets. This research is particularly relevant for the design of decentralized finance (DeFi) protocols and other blockchain-based governance systems.

1.3. Our Contributions

This paper makes several key contributions to the understanding of Nash Equilibria in Tullock contests, particularly in the context of algorithmic game theory and its application to blockchain systems. These contributions address both theoretical and practical aspects of Tullock contests under varying conditions of elasticity and effort.

1. **Complexity of Computing Nash Equilibria in Tullock Contests**: We provide a detailed analysis of the computational complexity involved in determining pure Nash Equilibria (PNE) in Tullock contests. Specifically, we identify the role of the elasticity parameter r_i in governing the feasibility of computing a PNE. Our results show that when no contestant's r_i lies in the interval $(1, 2]$, the computation of PNE is efficient and can be achieved via polynomial-time algorithms. However, when multiple r_i values fall within this interval, determining the existence of a PNE becomes NP-complete. This provides novel insights into the boundary conditions for tractable and intractable instances of Tullock contests.
2. **Development of Approximation Algorithms:** For cases where some contestants' r_i values lie within $(1, 2]$, we design a Fully Polynomial-Time Approximation Scheme (FPTAS) that efficiently computes an ϵ-approximate Nash Equilibrium when an exact PNE may not be achievable due to computational complexity. This approximation algorithm expands the range of tractable cases and demonstrates its effectiveness even in scenarios that are NP-complete.

3. **Practical Algorithmic Implementations:** We implement the proposed algorithms and evaluate their performance in large-scale computational experiments. Our results show that the algorithms can solve complex Tullock contest instances efficiently, including both small- and large-scale cases. These implementations contribute to practical applications in fields like blockchain, where Tullock-like competitive structures are prevalent, such as in mining or decentralized governance systems.

These contributions collectively enhance the understanding of Tullock contests in both theoretical and applied contexts, particularly in scenarios where computational complexity poses challenges. Our findings have implications for the design of efficient algorithms in game-theoretic models and their application to decentralized networks, highlighting the intersection of algorithmic game theory and blockchain economics.

2. Preliminaries: Tullock Contests and Their Useful Properties

This paper develops algorithmic results for *Tulock contests*, an impactful and textbook-style contest model of[1]. A Tullock contest consists of n contestants ($n \geq 2$), competing for a reward amount R (> 0). The action of each contestant $i \in [n]$ is to pick a non-negative *effort level* $x_i \in \mathbb{R}_{\geq 0}$. Each contestant i is characterized by a *production function* f_i, fully specified by two positive parameters $(a_i, r_i) \in \mathbb{R}^2_{>0}$. Formally, at cost x_i, contestant i's production is $f_i(x_i) = a_i x_i^{r_i}$, in which $a_i > 0$ describes i's *production efficiency*, and r_i (> 0) captures the *elasticity* of effort[1]. Notably, many previous works restrict r_i to be the same for each contestant, whereas our model here allows heterogeneous elasticities. As we shall show, this generality would not fundamentally change the complexity of the contest.

All contestants are assumed to simultaneously move. The Tullock contests models the probability p_i for contestant i to win as being proportional to i's production, given by: $p_i = \frac{f_i(x_i)}{\sum_{j=1}^n f_j(x_j)}$. Hence the payoff for each i is naturally formulated as follows, where the effort x_i is interpreted as effort cost:

$$u_i(x_i; x_{-i}) = \frac{a_i (x_i)^{r_i}}{\sum_{j=1}^n a_j (x_j)^{r_j}} \cdot R - x_i. \tag{1}$$

A Tullock contest is thus fully specified by $2n+1$ positive numbers, $\{(a_i, r_i)\}_{i=1}^n \cup \{R\}$. These numbers hence are the input to the computational problem of our interest, which looks to output an equilibrium of the game (as defined below), or asserts no equilibrium exists. To avoid uninteresting

doi.org/10.32388/WPJBOT

corner cases, we assume $R > 1$ throughout the paper; that is, the contest has a non-trivial amount of reward.[1]

Definition 1 (Pure-Strategy Nash Equilibrium). *An action profile* $x^* = \{x_1^*, x_2^*, \cdots, x_n^*\}$ *is a pure-strategy Nash Equilibrium if it satisfies*

$$u_i(x_i^*, \mathbf{x}_{-1}^*) \geq u_i(x_i, \mathbf{x}_{-1}^*) \quad \forall x_i \in \mathbb{R}_{\geq 0}, \forall i \in [n].$$

Regimes of the Elasticity

An important conceptual insight of our complexity-theoretic study is that the elasticity parameter r_i – which can be verified to equal $\frac{\Delta f_i / f_i}{\Delta x_i / x_i}$ – turns out to govern the complexity of Tullock contests' equilibria. The following three regimes of elasticity parameter are intrinsic.

Definition 2 (Small/Medium/Large Elasticity). *We say* r_i *is (1) small elasticity if* $r_i \in (0, 1]$; *(2) medium elasticity if* $r_i \in (1, 2]$; *(3) large elasticity if* $r_i \in (2, \infty)$.

Recall that $f_i(x_i) = a_i(x_i)^{r_i}$. Hence the production function is concave under small elasticity, but convex under medium and large elasticity. let $\mathcal{I}$ denote the set of all contestants, $\mathcal{I}^1$ denote the subset of contestants with $r_i \leq 1$, and $\mathcal{I}^2$ denote the subset of contestants with $r_i > 1$. The number of contestants in these subsets is denoted as $n_1 = |\mathcal{I}^1|$ and $n_2 = |\mathcal{I}^2|$, respectively, such that $n_1 + n_2 = n$, where n is the total number of contestants. These distinctions will aid in analyzing the properties of the game and the computational complexity associated with different regimes of r_i.

Remarks on Tullock Contests

A few remarks are worthwhile to mention. First, a natural question one might have is why the specific winning probability and utility format in Equation (1) is of particular interest. This question turns out to have a quite satisfactory answer. Classic works[22][23] provided axiomatic justifications and show that under natural axioms, contestants' utilities are *uniquely* characterized by the form of Eq. (1). These axioms are: (1) contestant i's winning probability strictly increases in x_i and decreases in x_j for any $j \neq i$; (2) the choice between two alternatives is independent of an unchosen third alternative (also widely known as Luce's Choice Axiom[24] or independence from irrelevant alternatives); (3) if every contestant simultaneously scale up their effort level by any factor λ, each contestant's winning probability would not change. Second, Tullock contest is a fundamental model in political economy to study rent-seeking. For instance,[25] identify conditions under which a variety of rent-seeking

 doi.org/10.32388/WPJBOT

contests, innovation tournaments, and patent-race games are strategically equivalent to the Tullock contest. Third, two notable special cases of the contest are the *lottery contest* (i.e., $r_i = 1, \forall i$) and *winner-take-all* contest ($r_i = \infty, \forall i$).

2.1. Equilibrium Properties

A (Useful) Equivalent Contest Re-formulation

For analytical convenience, it turns out to be useful to re-formulate the contest by switching from strategy variables x_i to $y_i = f_i(x_i) = a_i(x_i)^{r_i}$, which represents i's production amount from effort level x_i. Naturally, $A = \sum_{j=1}^n y_j$ denote the aggregate production. Since $x_i = (y_i/a_i)^{1/r_i}$, the re-formulated contestant utility is thus expressed as follows:

$$\pi_i(y_i; y_{-i}) = \frac{y_i}{A} \cdot R - (y_i/a_i)^{1/r_i}.$$

Note that the two utility functions u_i, π_i are different due to their different formats with different inputs, despite describing the same contestant's utility. For convenience, we denote $g_i(y_i) = (y_i/a_i)^{1/r_i}$ which is the inverse of production function f_i.

It is not difficult to see that this re-formulation is equivalent to the original game since x_i and y_i are in one-to-one correspondence; hence the equilibrium in Definition 1 can be adapted to equilibrium conditions for y^* and we will analyze them interchangeably. The contest re-formulation above leads to the following two natural concepts that are crucial to later analysis.

Definition 3 (Aggregated Action and Action Share). *For any contestant i exerting effort x_i, define i's action $y_i = f_i(x_i) = a_i(x_i)^{r_i}$ as i's production amount. Then the total production $A = \sum_{j=1}^n y_j$ is called the* aggregated action, *whereas $\sigma_i = \frac{y_i}{A}$ is called contestant i's* action share*.*

The action share σ_i equals the probability that contestant i wins the contest, whereas $\sigma_i A$ is i'th production amount which is also referred to as i'th *action*.

Before delving into the formal definitions, it is helpful to think of the action share σ_i as both the probability that contestant i wins the contest and the proportion of the total action A attributed to i. Contestants strategically adjust their actions based on their production capabilities and the observed behavior of others. To formalize this, we introduce the concepts of action share functions under best-response, which describe contestants' optimal behaviors under various conditions.

Definition 4 (Action Share function under Best-Response [17]). *The action share functions $k_1(A;a_i)$ and $k_2(A;a_i)$ represent the optimal response of contestant i to a given aggregated action A:*

$$k_1(A;a_i) \quad \text{satisfies} \quad b_1(A, k_1(A;a_i); a_i) = 0,$$

$$k_2(A;a_i) \geq \frac{r_i - 1}{r_i} \quad \text{satisfies} \quad b_2(A, k_2(A;a_i); a_i) = 0,$$

where best-response $b(A, \sigma_i; a_i)$ defined as:

$$b(A,\sigma_i;a_i) = \begin{cases} (1-\sigma_i)R - g_i'(\sigma_i A)A, & \text{if~} r_i \leq 1, \\ a_i R^{r_i}(1-\sigma_i)^{r_i} - A, & \text{if~} r_i > 1, \end{cases}$$

where $g_i(\cdot)$ is the cost function, defined as the inverse of $f_i(\cdot)$.

For convenience, we denote the best-response for $r_i \leq 1$ as $b_1(A,\sigma_i;a_i)$, and for $r_i > 1$ as $b_2(A,\sigma_i;a_i)$.

Remarks

For a contestant $i \in \mathcal{I}^1$, the action share function $k_1(A;a_i)$ has a unique solution for any given aggregate action A. This uniqueness ensures that $k_1(A;a_i)$ is well-defined and, as we will analyze later, exhibits monotonic behavior with respect to A, reflecting a consistent relationship between the aggregate action and the contestant's optimal response.

For a contestant $i \in \mathcal{I}^2$, the equation governing the action share, $a_i R^{r_i} r_i^{r_i} (1-\sigma_i)^{r_i} \sigma_i^{r_i - 1} = A$, indicates that σ_i generally has at least one solution for a given A. By analyzing the utility function, we observe that when $\sigma_i = \frac{r_i - 1}{r_i}$, the corresponding aggregate action A is given by:

$$A = a_i R^{r_i} \left(\frac{r_i - 1}{r_i} \right)^{r_i - 1}.$$

At this specific point, the utility of contestant i equals zero. Hence, any valid value of $k_2(A;a_i)$ must be a solution to the equation $b_2(A,\sigma_i;a_i) = 0$, where the utility of contestant i is strictly positive. As established in [17], the equation $b_2(A,\sigma_i;a_i) = 0$ admits exactly one solution that satisfies this positivity condition. Furthermore, this solution always lies within the interval $[\frac{r_i - 1}{r_i}, 1]$, ensuring the action share is well-bounded.

It is critical to emphasize that the action share functions $k_1(A;a_i)$ and $k_2(A;a_i)$ describe contestant i's optimal response to a given aggregate action A. However, these functions do not necessarily correspond to contestant i's realized action share in equilibrium. The actual action share σ_i in equilibrium is determined by the aggregate actions of all contestants, which inherently depend on the equilibrium conditions. Notably, A represents the total contributions from all players, which may or may not include contestant i's own participation. Consequently, the relationship between k_1, k_2, and the equilibrium value of σ_i is governed by a broader consistency condition across all contestants' strategies.

Fact 1. *The optimal action share σ_i depends on r_i and the aggregate action A, as follows:*

$$\sigma_i = \begin{cases} \begin{cases} 0, & \text{if~} f_i'(0) < \infty \text{~and~} A \geq R f_i'(0), \\ k_1(A; a_i), & \text{otherwise}, \end{cases} & \text{if~} r_i \leq 1, \\ \begin{cases} k_2(A; a_i), & \text{if~} A \in [0, \underline{A}_i], \\ k_2(A; a_i) \text{~or~} 0, & \text{if~} A \in [\underline{A}_i, \overline{A}_i], \\ 0, & \text{if~} A > \overline{A}_i, \end{cases} & \text{if~} r_i > 1. \end{cases}$$

where

$$\underline{A}_i \equiv a_i R^{r_i} \frac{(r_i - 1)^{r_i - 1}}{r_i^{r_i}}, \quad \overline{A}_i \equiv r_i \cdot \underline{A}_i = a_i R^{r_i} \left(\frac{r_i - 1}{r_i} \right)^{r_i - 1}$$

Key Insights:

For $r_i \leq 1$, the optimal response action share σ_i is uniquely determined for any aggregate action A and exhibits explicit monotonicity properties. This case will be rigorously analyzed in Proposition 1, where we demonstrate the monotonicity properties of $k_1(A; a_i)$ and its implications for equilibrium computation.

When $r_i > 1$, the parameters $\underline{A}_i$ and $\overline{A}_i$ play critical roles in determining contestant i's participation thresholds. The upper boundary, $\overline{A}_i$, marks the point beyond which contestant i ceases to participate. Specifically, when $A > \overline{A}_i$, the utility function for contestant i becomes negative, rendering participation unprofitable. As a result, the best response in this range is $\sigma_i = 0$, reflecting the fact that any effort exerted would lead to a net loss.

The lower threshold, $\underline{A}_i$, defines the point at which contestant i begins to consider abstention. If the aggregate contribution of all other contestants equals $\underline{A}_i$, contestant i's utility is exactly zero, irrespective of the effort they exert. Consequently, their best response is also $\sigma_i = 0$. For $A < \underline{A}_i$, however, participation becomes advantageous, as the utility derived from contestant i's effort exceeds the associated cost. In this range, their optimal action share is given by $k_2(A; a_i)$, which satisfies their best-response condition.

The interval $A \in [\underline{A}_i, \overline{A}_i]$ presents a nuanced landscape where contestant i may either participate or abstain, depending on the aggregate contributions.

doi.org/10.32388/WPJBOT

1. **Participation by Contestant i:** In this scenario, A includes i's contribution. The remaining aggregate action, $A - \sigma_i A$, reflects the actions of other contestants. Contestant i adjusts σ_i to balance their contribution optimally with the behavior of others.
2. **Non-Participation by Contestant i:** Here, A is solely the result of other contestants' actions. Contestant i's best response is to remain inactive, leading to $\sigma_i = 0$.

These dynamics, governed by the thresholds $\underline{A}_i$ and $\overline{A}_i$, highlight the interplay between individual decisions and aggregate outcomes. This behavior underscores the complexity of equilibrium analysis in contests with $r_i > 1$, where the potential for multiple equilibria emerges within the interval $[\underline{A}_i, \overline{A}_i]$.

Proposition 1 from [17] characterizes the monotonicity of the optimal response action share, which will be instrumental for the equilibrium analysis that follows.

Proposition 1 ([17]). *Given the best-response $b(A, \sigma_i; a_i)$ mentioned above, for any agent i:*

1. *If $i \in \mathcal{I}^1$:*
   a. *If $r_i < 1$, σ_i decreases as A increases, with $\sigma_i \to 0$ as $A \to \infty$.*
   b. *If $r_i = 1$, σ_i decreases for $0 \leq A < Rf'(0)$ and becomes $\sigma_i = 0$ for $A \geq Rf'(0)$.*
2. *If $i \in \mathcal{I}^2$, within the interval $A \in [0, \overline{A}_i]$: σ_i decreases monotonically as A increases.*

Figure 1 provides a clear visualization of how the action share σ_i evolves with the aggregate action A under different elasticity regimes ($r_i \leq 1, r_i = 1, r_i > 1$). The figure highlights the distinct patterns of behavior for contestants with varying elasticity parameters, demonstrating the critical role of r_i in shaping strategic interactions.

doi.org/10.32388/WPJBOT

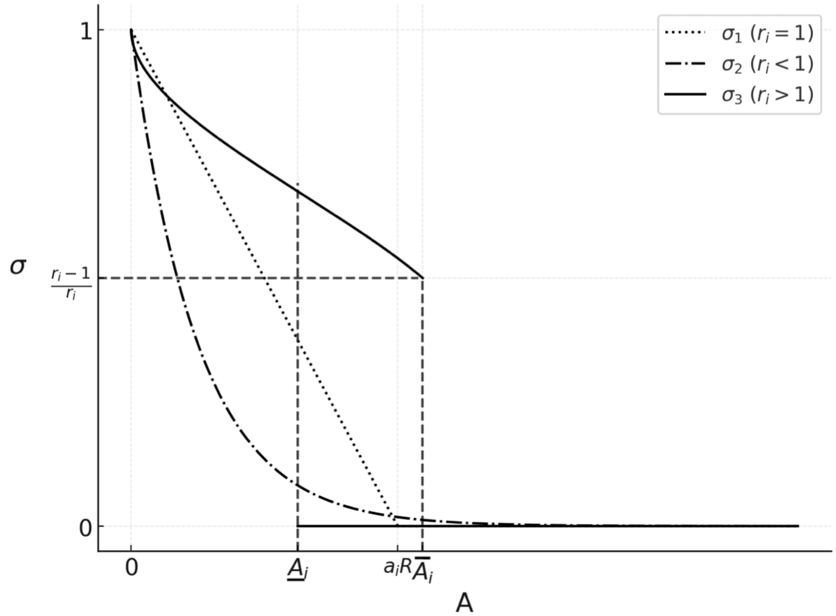

Figure 1. Action share as a function of aggregate action

Active Players.

A contestant i is said to be *active* if $\sigma_i > 0$ in equilibrium. We now present a general characterization of Nash equilibrium in terms of the aggregated action A, the set of active contestants $\mathcal{I}^A$, and the corresponding action share profile $\boldsymbol{\sigma}^* = \{\sigma_i^*\}_{i \in \mathcal{I}^A}$. This comprehensive framework integrates the activity conditions for contestants with $r_i \leq 1$ and $r_i > 1$, ensuring consistency with individual rationality and best response dynamics. The following proposition formalizes these conditions:

Proposition 2 ([17]). *The triplet $\{A^*, \mathcal{I}^A, \boldsymbol{\sigma}^*\}$ constitutes a Nash equilibrium if and only if the following conditions are satisfied:*

1. *Given A^*, it is individually rational for each contestant in $\mathcal{I}^A$ to be active and for each contestant not in $\mathcal{I}^A$ to be inactive. Formally:*
   - *For contestants with $r_i = 1$, $a_i R_{i \notin \mathcal{I}^A} \leq A^* < a_i R_{i \in \mathcal{I}^A}$.*
   - *For contestants with $r_i > 1$, $\max_{i \notin \mathcal{I}^A} \underline{A}_i \leq A^* \leq \min_{i \in \mathcal{I}^A} \overline{A}_i$.*
2. *Given A^* and $\mathcal{I}^A$, the action shares $\boldsymbol{\sigma}^*$ are consistent with the best response of each active contestant. Specifically:*

   $b_1(A^*, \sigma_i^*; a_i) = 0, \quad \forall i \in \{\mathcal{I}^A \cap \mathcal{I}^1\},$

 doi.org/10.32388/WPJBOT

$b_2(A^*, \sigma_i^*; a_i) = 0, \quad \forall i \in \{\mathcal{I}^A \cap \mathcal{I}^2\},$

where b_1 and b_2 are the functions corresponding to $r_i \leq 1$ and $r_i > 1$, respectively, as previously defined.

3. *The action shares of all active contestants sum to* 1*, i.e.,* $\sum_{i \in \mathcal{I}^A} \sigma_i^* = 1$.

In accordance with the methodology outlined in [17], the identification of pure Nash Equilibria in Tullock contests necessitates a systematic two-step approach. The initial step involves determining the aggregate action A, which represents the total effort exerted by all contestants and serves as a fundamental parameter governing the contest dynamics. Once A is established, the subsequent step requires the precise identification of the active players $\mathcal{I}^A$, those contestants whose equilibrium strategies involve non-zero contributions to the aggregate action. A critical condition to ensure the validity of this equilibrium is that the cumulative action shares of these active players satisfy the equilibrium constraint: $\sum_{i \in \mathcal{I}^A} \sigma_i = 1$. This structured approach not only provides a theoretical basis for equilibrium analysis but also underscores the intricate interplay between individual strategies and collective outcomes in such contests.

3. Efficiently Solvable Regimes

The computational complexity of finding a Pure-Strategy Nash Equilibrium (PNE) in Tullock contests varies based on the characteristics of contestants' production functions, specifically the elasticity parameter r_i. In this section, we identify specific conditions under which the PNE can be computed efficiently. We focus on three distinct regimes: small elasticity ($r_i \leq 1, \forall i$), large elasticity ($r_i \geq 2, \forall i$) and mixed regimes where both types are present.

3.1. Tullock is a Monotone Game under Small Elasticity ($r_i \leq 1, \forall i$)

The regime in which all contestants have concave production functions constitutes a fundamental scenario in the theoretical analysis of Tullock contests. Under this setting, the game can be rigorously classified as a subclass of monotone games [26], specifically as a socially-concave game. This classification is supported by auxiliary results established in prior research [27][28][29]. Leveraging the properties of monotone games, as outlined in [26], the existence of a pure Nash equilibrium (PNE) in this regime is formally guaranteed, ensuring the validity of equilibrium solutions in this context.

From a theoretical perspective, concave production functions ensure that each contestant's utility function is strictly quasi-concave with respect to their effort. This property, as established in Proposition 1, guarantees the monotonicity of contestants' best-response strategies, where action shares σ_i decrease strictly as the aggregate action A increases. As a result, the equilibrium condition:

$$\sum_{i=1}^{n} \sigma_i = 1,$$

is uniquely satisfied, which means the contest has a unqiue Nash Equilibrium in pure strategy as demonstrated in [30]. This monotonicity simplifies the computation of equilibria, serving as the foundation for the algorithmic methods discussed in this section.

Figure 2 illustrates the relationship between the action share σ_i and the aggregate action A for the case where $r_i = 1$. As shown, σ_i decreases monotonically as A increases, consistent with the theoretical result that the best-response strategy exhibits strict monotonicity under concave production functions. This relationship is fundamental to the equilibrium computation, as it ensures the existence and uniqueness of a solution satisfying the equilibrium condition $\sum_{i=1}^{n} \sigma_i = 1$.

doi.org/10.32388/WPJBOT

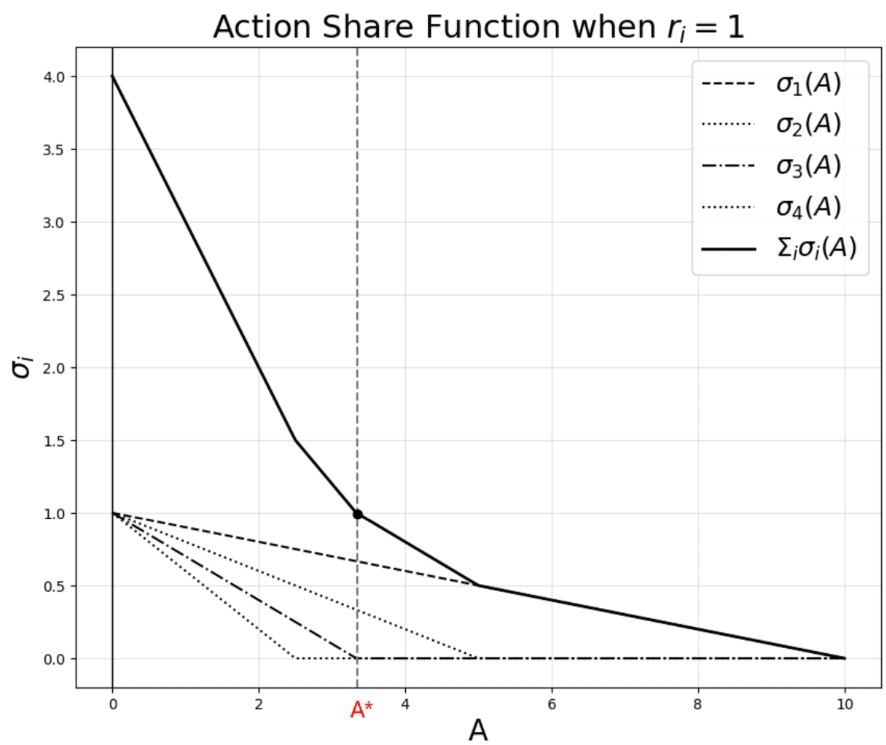

Figure 2. Action share as a function of aggregate action when $r_i = 1$. This highlights the monotonic relationship between A and σ_i, fundamental to the computation of equilibria.

The monotonic behavior of σ_i not only simplifies the analysis but also forms the basis for efficient computational methods, such as the bisection approach, which leverages this property to identify the equilibrium point with polynomial complexity. Figure 2 visually emphasizes how this monotonicity constrains the contestants' strategies, thereby reducing the dimensionality of the problem.

Building on these theoretical insights, we propose two computational methods for efficiently identifying unique pure Nash Equilibria (PNE) under concave production functions.

Bisection Method for Finding Pure Nash Equilibrium

The bisection method exploits the monotonicity of action shares σ_i with respect to A. By iteratively adjusting A within a bounded interval $[A_{\min}, A_{\max}]$, the algorithm identifies the unique equilibrium point satisfying $\sum_{i=1}^{n} \sigma_i = 1$. This method is particularly effective due to its deterministic convergence, achieving polynomial complexity in the number of iterations.

In addition to the bisection method, other polynomial-time algorithms can be employed to find the unique PNE. One such method is the Multi-agent Mirror Descent (MMD) with perfect gradient algorithm, which is particularly useful in games with differentiable utility functions.

Multi-Agent Mirror Descent for Finding PNE

The core idea of the MMD with perfect gradient algorithm, as described in the literature (e.g., [31][28]), is to iteratively adjust each player's strategy based on the exact gradient of their utility function. The algorithm updates the strategies by performing a mirror descent step, which ensures that all players' strategies converge to the unique pure-NE. The MMD with perfect gradient algorithm is robust and efficient, providing convergence guarantees under mild regularity conditions on the utility functions.

This approach is well-suited for large-scale contests with differentiable utility functions. Its iterative nature allows for parallelization, enabling fast convergence even in high-dimensional strategy spaces.

3.2. Tullock Contest under Large Elasticity ($r_i > 2, \forall i$)

In this regime, the elasticity parameters r_i are greater than or equal to 2, leading to convex production functions that exhibit increasing marginal returns on effort. Unlike the concave regime, where the equilibrium is distributed among contestants, the convex regime typically results in a "winner-takes-all" scenario, where only one contestant remains active in the equilibrium.

Proposition 3. *A Tullock contest with $r_i > 2$ for all contestants admits no pure Nash equilibrium.*

To provide an intuitive explanation, consider the condition $r_i > 2$. For any active contestant i, their action share σ_i must exceed $\frac{r_i - 1}{r_i} > \frac{1}{2}$. This implies that at most one contestant can be active in equilibrium, as the total action share across all contestants must sum to 1. However, if only one contestant is active, their action share would necessarily be $\sigma_i = 1$, resulting in an aggregate action $A = 0$. In this scenario, all contestants would find it optimal to become active, as the absence of competition would maximize their utility. This contradiction demonstrates that no configuration satisfies the equilibrium conditions, confirming the absence of a pure Nash equilibrium under $r_i > 2$. Full details are provided in Appendix A.

Corollary 1. *In a Tullock contest, if a pure Nash Equilibrium exists, there can be at most one active contestant with $r_i > 2$.*

This corollary, derived directly from Proposition 3, reinforces the "winner-takes-all" dynamic in the convex regime. The restriction to a single active contestant follows from the requirement that their action share exceeds $\frac{1}{2}$, leaving no room for additional active players with large elasticity.

3.3. Mixed Regimes with no Medium Elasticity: $r_i \in (0,1] \cup (2,\infty)$

In real-world scenarios, contestants often exhibit heterogeneous elasticity parameters, resulting in mixed regimes. These regimes combine contestants with small elasticity ($r_i \leq 1$) and large elasticity ($r_i > 2$). The interplay between these two types of contestants introduces unique challenges in equilibrium computation due to the contrasting strategic behaviors.

Analysis of Mixed Regimes

As outlined in Corollary 1, pure Nash equilibrium in mixed regimes manifest under two specific scenarios:

1. **Homogeneous Concave Regime:** All active contestants satisfy small elasticity ($r_i \leq 1$). This case follows directly from the results established in Section 3.1.
2. **Heterogeneous Regime with a Dominant Player:** Among the active contestants, exactly one has large elasticity ($r_i > 2$), while the remaining contestants (if any) satisfy small elasticity ($r_i \leq 1$).

In the heterogeneous regime, the contestant with large elasticity dominates the equilibrium dynamics, contributing significantly to the aggregate action. This requires the remaining contestants to adjust their strategies to satisfy the equilibrium conditions.

Computational Methodology for Mixed Regimes

To compute equilibria in mixed regimes, we propose a hybrid algorithm that integrates the methods developed for purely concave and super-convex cases. The algorithm consists of two main steps. First, identify potential active players by evaluating each contestant with $r_i > 2$. This step verifies whether a contestant can satisfy the equilibrium conditions as the dominant player, while all others remain inactive. Second, for each identified candidate, compute the equilibrium contributions of the concave contestants ($r_i \leq 1$). This is achieved using the bisection method described in Section 3.1, ensuring that the aggregate action aligns with the equilibrium condition $\sum_{i=1}^{n} \sigma_i = 1$.

Theorem 1. *A polynomial-time algorithm exists to determine the presence of a pure Nash equilibrium in mixed elasticity regimes, provided no elasticity parameter* r_i *lies within the interval* $(1,2]$. *If such an* *equilibrium exists, the algorithm identifies it.*

We prove Theorem 1 by constructing a polynomial-time algorithm (Algorithm 2). It combines the monotonicity properties of concave production functions with the dominance property of large elasticity contestants. The hybrid nature of this approach ensures polynomial complexity. Full details are provided in Appendix B.

The analysis demonstrates that hybrid computational methods can efficiently handle mixed elasticity scenarios. These results apply to contests with heterogeneous resource distributions, such as economic markets with dominant firms or political campaigns with asymmetrically funded candidates. Moreover, the hybrid framework offers a foundation for future research into more complex regimes involving intermediate elasticity parameters ($r_i \in (1, 2]$).

4. The Challenging Case: Medium Elasticity

While small and large elasticity regimes allow for efficient computation of Pure Nash Equilibrium, medium elasticity ($r_i \in (1, 2]$) pose significant computational challenges. These intermediate elasticity values introduce non-linear payoffs, resulting in complex strategic interactions and non-monotonic dynamics that fundamentally complicate the equilibrium analysis.

4.1. The Hardness of Determining PNE

This subsection establishes the computationally intractability of determining whether a given Tullock contest with all $r_i \in (1, 2)$ admits a pure NE or not. Formally, our main result is the following.

Theorem 2. *If there are polynomial many contestants whose $r_i \in (1, 2]$, then determining the existence of a pure Nash equilibrium in a Tullock contest can not be solved in polynomial time, assuming the Exponential Time Hypothesis (ETH).*

Our reduction is from the following variant of the Subset Sum problem.

Definition 5 (Subset Sum with Large Targets (SSLT)). *Given a set of positive numbers $\mathbf{Z} = \{z_1, \ldots, z_n\}$ and a target $\bar{z}$ satisfying $\bar{z} \geq 2 \max_{z \in \mathbf{Z}} z$, determine whether there exists a subset $S \subseteq \mathbf{Z}$ such that $\sum_{z \in S} z = \bar{z}$.*

The proof has two major steps. First, we establish a reduction from Tullock contests to SSLT (Lemma 1). Second, we show that SSLT is intractable, assuming the Exponential Time Hypothesis (ETH).

We now establish a relationship between this problem and the PNE problem in Tullock contests under medium elasticity. Specifically, we construct a Tullock contest such that the existence of a PNE corresponds directly to a solution of the Subset Sum with Bounded Maximum problem.

Lemma 1. *For any instance of SSLT, there exists a corresponding Tullock contest under medium elasticity* $(\forall i, r_i \in (1,2])$. *SSLT has a solution if and only if the corresponding Tullock contest has a Nash equilibrium, with the equilibrium directly corresponding to the solution of the SSLT.*

The proof demonstrates that the PNE problem in Tullock contests with $r_i \in (1,2]$ exactly mirrors the combinatorial structure of the SSLT. Specifically, each contestant's action share corresponds directly to a subset element, and the equilibrium condition aligns with satisfying the target sum in the subset sum framework. This one-to-one correspondence ensures that solving the equilibrium problem is as computationally hard as solving the SSLT. Full details are provided in Appendix D.

Lemma 2. *SSLT can not be solved in polynomial time, assuming the Exponential Time Hypothesis (ETH).*

The Subset Sum with Bounded Maximum problem inherits the hardness of the original Subset Sum Problem due to its combinatorial nature. The bounded condition $\bar{z} \geq 2 \max_{z \in \mathrm{Z}} z$ ensures that any feasible subset must collectively satisfy the target while being constrained by an upper limit on individual elements. This additional constraint refines the solution space but does not reduce the exponential number of subset combinations that must be evaluated. A quasi-polynomial time reduction from the Subset Sum Problem to the bounded variant confirms that the latter retains the complexity of computation. Detailed steps of the reduction and its validation are provided in Appendix C.

This complexity is not confined to cases where all elasticity parameters lie within $(1,2]$. Even in mixed cases, where only polynomial many contestants has medium elasticity, the computational challenge persists. The presence of many contestant with $r_i \in (1,2]$ introduces the same combinatorial hardness. The detailed proof is provided in Appendix E.

Collectively, these findings underscore the intrinsic computational difficulty of Tullock contests involving medium elasticity values. The stark contrast between this regime and the more analytically tractable concave and super-convex cases highlights the unique challenges associated with non-monotonic payoff structures. From a practical standpoint, the hardness of the problem indicates that exact equilibrium computation is infeasible for large-scale contests, necessitating the exploration of approximate or heuristic methods.

doi.org/10.32388/WPJBOT

4.2. *Approximate Algorithms for Finding a PNE*

Given the NP-completeness of finding an exact PNE in this regime, it is clear that exact computation is not feasible for large instances. As a result, alternative solution concepts, such as ϵ-approximate solutions, become relevant. These approximations allow for a bounded deviation from the exact equilibrium, making it possible to achieve computationally efficient solutions while still capturing the strategic essence of the contest.

The concept of an ϵ-approximate solution provides a useful starting point for this approach. An ϵ-approximate solution maintains the core structure of a Nash Equilibrium but allows for a small deviation in terms of the sum of action shares, enabling more flexible computation without compromising strategic fidelity. This notion serves as a foundation for further analysis, where we will investigate its relationship with ϵ-Nash Equilibria and assess the gap between the two.

Definition 6 (ϵ-approximate solution). *The triplet $\{A^*, \mathcal{I}^A, \sigma^*\}$ constitutes a ϵ-approximate solution if and only if the following conditions are satisfied.*

1. Given A^*, it is individually rational for each contestant in $\mathcal{I}^A$ to be active and for each contestant not in $\mathcal{I}^A$ to be inactive. Formally:
    - For contestants with $r_i > 1$, $\max_{i \notin \mathcal{I}^A} \underline{A}_i \leq A^* \leq \min_{i \in \mathcal{I}^A} \bar{A}_i$.
    - For contestants with $r_i = 1$, $Rf_i'(0)_{i \notin \mathcal{I}^A} \leq A^* < Rf_i'(0)_{i \in \mathcal{I}^A}$.
2. Given A^* and $\mathcal{I}^A$, the action shares σ^* are consistent with the best response of each active contestant. Specifically:
    - $b_1(A^*, \sigma_i^*; a_i) = 0, \quad \forall i \in \{\mathcal{I}^A \cap \mathcal{I}^1\}$,
    - $b_2(A^*, \sigma_i^*; a_i) = 0, \quad \forall i \in \{\mathcal{I}^A \cap \mathcal{I}^2\}$,
3. *The action shares of all active players falls into a ϵ neighborhood of 1, i.e., $\sum_{i \in \mathcal{I}^A} \sigma_i^* \in (1-\epsilon, 1+\epsilon)$.*

4.2.1. *FPTAS for ϵ-Nash Equilibrium*

The main idea for circumventing the computational challenge for finding an exact Nash equilibrium in this game is to use an FPTAS algorithm for computing the subset sum problem that verifies whether there exists an action share profile that constitutes an ϵ-approximate solution. With this, we now have an FPTAS to find an ϵ-approximate solution. And we will show that the ϵ-approximate solution we found can be converted into an ϵ-Nash equilibrium. Let

$$\rho \triangleq \max_{a_i, i\in\mathcal{I}, A\in[\underline{A}_j, \overline{A}_j], j\in\mathcal{I}^2} \left\{ \max\left(\left| \frac{\partial k_1(A;a_i)}{\partial A} \right|, \left| \frac{\partial k_2(A;a_i)}{\partial A} \right| \right) \right\},$$

where recall $k_1(A;a_i)$, $k_2(A;a_i)$ is the solution such that $b_1(A, k(A;a_i);a_i) = 0$, $b_2(A, k(A;a_i);a_i) = 0$. Note that when $A \in [\underline{A}_i, \overline{A}_i]$, the solution $k_1(A;t_i)$, $k_2(A;t_i)$ is continuous and twice differentiable in A, and hence

$$\left| \frac{\partial \sigma_i}{\partial A} \right| > 0.$$

ρ exists and is finite (detail calculation see L_1 and L_2 in proposition 9). By treating ρ as a constant, we show that our algorithm only has a polynomial dependence on the number of contestants.

Remarks

Let $\underline{A} = \min_{i\in\mathcal{I}^2} \underline{A}_i$ and $\overline{A} = \max_{i\in\mathcal{I}^2} \overline{A}_i$. Based on the participation behavior of contestants, the space of aggregate actions $\mathbb{R}_{\geq 0}$ is divided into three distinct intervals: $(0, \underline{A}]$, $(\underline{A}, \overline{A}]$, and $(\overline{A}, \infty)$. In the interval $(0, \underline{A}]$, all contestants, including those in $\mathcal{I}^2$, participate with certainty. Since the set of active players is fixed in this interval, a PNE can be efficiently identified using binary search, which also satisfies the conditions for an ϵ-approximate solution. For the interval $(\overline{A}, \infty)$, no contestant in $\mathcal{I}^2$ participates, and the set of active players in $\mathcal{I}^1$ is determinable. Binary search can again be employed to find a PNE in this range, ensuring an ϵ-approximate solution. The interval $(\underline{A}, \overline{A}]$ presents additional challenges, as at least one contestant in $\mathcal{I}^2$ may or may not participate. While the active set of players in $\mathcal{I}^1$ can still be determined, the main complexity lies in verifying the existence of an ϵ-approximate solution within this range.

For the challenging interval $(\underline{A}, \overline{A}]$, we construct a set of candidate nodes, $\mathcal{A}$, to identify ϵ-approximate solutions. This set comprises:

- A_1: Nodes evenly spaced across $(\underline{A}, \overline{A}]$ with a spacing of $\delta > 0$, formally defined as:

$$A_1 = \{A | A = \underline{A} + k\delta, A \leq \overline{A}, k \in \mathbb{N}^+\}.$$

- A_2: Points $\{\underline{A}_i, \overline{A}_i\}_{i\in\mathcal{I}^2}$, where the best-response correspondence is discontinuous.

The combined set is: $\mathcal{A} = \mathrm{sorted}(A_1 \cup A_2)$, and the total number of nodes is bounded by: $|\mathcal{A}| = 2n_2 + \frac{\overline{A} - \underline{A}}{\delta}$.

Lemma 3 ensures that it's sufficient to only verify the existence of ϵ-approximate solution at the selected nodes.

Lemma 3. *For any $\epsilon > 0$, let $\delta < \frac{\epsilon}{\rho n}$. If there exists an ϵ-approximate solution in an interval $[A_1, A_2]$, where A_1 and A_2 are two adjacent nodes in $\mathcal{A}$, then at least one of these nodes, A_1 or A_2, constitutes an ϵ-approximate solution.*

The construction of $\mathcal{A}$ and the result of Lemma 3 provide the theoretical foundation for efficiently searching for ϵ-approximate solutions. By restricting the verification process to the nodes in $\mathcal{A}$, the algorithm significantly reduces the computational overhead, as it eliminates the need to explore the entire range of aggregate actions A.

We now leverage these insights to establish the existence of a polynomial-time algorithm that outputs an ϵ-approximate solution whenever a pure Nash equilibrium exists.

Theorem 3. *There exists a $\mathtt{poly}(\frac{1}{\epsilon}, \rho, n)$ time algorithm that is guaranteed to output an ϵ-solution whenever a pure Nash equilibrium exists.*

Here, we construct the algorithm J based on the interval analysis discussed above. For the intervals $(0, \underline{A}]$ and $[\overline{A}, \infty)$, the computation is straightforward due to the fixed or easily determinable active player sets. The primary computational challenge lies within the interval $[\underline{A}, \overline{A}]$, where the active player set is not fixed. To address this, we focus on the candidate nodes $\mathcal{A}$ constructed within this interval.

At each selected node $A_i \in \mathcal{A}$, the algorithm first computes the action shares of active players in $\mathcal{I}^1$. Subsequently, it evaluates the participation status of the remaining contestants in $\mathcal{I}^2$. For these contestants, verifying an ϵ-approximate solution reduces to solving a subset selection problem. This subset selection problem is efficiently addressed using the *Approximate Subset Sum Algorithm* [32], which operates with a computational complexity of $O(\frac{n}{\epsilon})$.

By combining these steps, the algorithm ensures a total runtime complexity of $O(\frac{n^2}{\epsilon^2})$, where $|\mathcal{A}| = O(\frac{n}{\epsilon})$ and each verification step has a complexity of $O(\frac{n}{\epsilon})$. The detailed proof of this construction is provided in Appendix G.

With the FPTAS framework established, we now examine its relationship with ϵ-Nash equilibrium. The algorithm efficiently computes an approximate solution within polynomial time, as demonstrated. However, to ensure that this solution qualifies as an ϵ-Nash equilibrium, it is essential to establish certain continuity and stability properties of the contestants' strategies and utilities. These properties are formally captured in Lemma 3, which provides the necessary theoretical underpinning for connecting the algorithm's output to the ϵ-NE concept.

Lemma 4. *Let $\underline{A} \triangleq \min_{i\in\mathcal{I}^2} \underline{A}_i$ and $\overline{A} \triangleq \max_{i\in\mathcal{I}^2} \overline{A}_i$. The following continuity properties hold:*

1. *For a contestant i, the action share σ_i is Lipschitz continuous with respect to the aggregate action A. Specifically:*

- *If $i \in \mathcal{I}^2$, σ_i is Lipschitz continuous on the interval $A \in [\underline{A}, \overline{A}_i) \cup (\overline{A}_i, \overline{A}]$.*

2. *If $i \in \mathcal{I}^1$, σ_i is Lipschitz continuous on the interval $A \in [\underline{A}, \overline{A}]$.*

1. *The utility function is Lipschitz continuous with respect to the aggregate action A. Specifically:*

- *If $i \in \mathcal{I}^2$, the utility function is Lipschitz continuous on the interval $A \in [\underline{A}, \overline{A}_i) \cup (\overline{A}_i, \overline{A}]$.*
- *If $i \in \mathcal{I}^1$, the utility function is Lipschitz continuous on the interval $A \in [\underline{A}, \overline{A}]$.*

The Lipschitz continuity properties in Lemma 4 ensure that small variations in the aggregate action A result in bounded changes in action shares σ_i and utility functions. This property is crucial for establishing the stability and approximation guarantees of the proposed algorithm. The full proof of Lemma 4, detailing the derivation of the Lipschitz constants, is provided in Appendix H.

Proposition 4. *If a pure Nash equilibrium exists, there is at least one of the ϵ-approximate solutions obtained through Algorithm 3 constitute an $(L\epsilon)$-Nash equilibrium.*

We begin by considering the aggregate action A under PNE. If A corresponds to any contestant's boundary, our solution will identify it directly, as the algorithm systematically enumerates all boundary points. If A does not coincide with any contestant's boundary, as detailed in Appendix I, there must exist an ϵ-approximate solution sufficiently close to A. Leveraging the Lipschitz continuity properties, we demonstrate that the deviation from a contestant's best-response utility is bounded by $L\epsilon$, ensuring the ϵ-approximate solution is an $L\epsilon$-NE.

The result highlights that the proposed algorithm efficiently computes approximate equilibria with a controllable accuracy parameter ϵ. From a theoretical perspective, this result bridges the gap between exact equilibrium computation, which is often computationally prohibitive. The focus on $(L\epsilon)$-Nash equilibria offers a flexible and computationally feasible framework for addressing complex strategic interactions in Tullock contests, particularly in settings where elasticity parameters $r_i \in (1, 2]$ lead to non-linear payoff dynamics. This approach not only extends the tractability of Tullock contests to challenging regimes but also provides a foundation for future research into scalable solutions for real-world strategic decision-making problems.

5. Implementation of the Algorithm

To support the practical application of our proposed algorithm for computing ϵ-Nash Equilibria (ϵ-NE) in Tullock contests, we have developed a Python module. This implementation integrates all core components of the algorithm, emphasizing computational efficiency and scalability across diverse contest configurations.

The module has been thoroughly tested for performance and correctness, demonstrating its adaptability to both theoretical analysis and real-world simulations. Detailed documentation and usage examples are provided to facilitate reproducibility and further research.

The Python module is publicly available on GitHub: https://github.com/1653133307/Tullock.

6. Discussion

Summary of Findings

This paper explores the computational complexity of computing Pure Nash Equilibria (PNE) in Tullock contests under various elasticity conditions. The primary contributions include:

- **Complexity Classification:** We demonstrated that the computational complexity for determining a PNE hinges critically on the elasticity parameter r_i. When all r_i values are outside the interval $(1, 2]$, efficient polynomial-time algorithms exist. However, when multiple r_i values fall within $(1, 2]$, the problem becomes intractable.
- **Approximation Framework:** To address the hardness in the challenging case, we propose an approximation framework to tackle the challenges in the hard case, demonstrating that the ϵ-approximate solutions produced by our algorithm guarantee at least one valid $(L\epsilon)$-Nash equilibrium, provided a PNE exists. This provides a computationally feasible approach for obtaining approximate solutions while preserving the equilibrium properties.
- **Practical Implementation:** A Python module was implemented to compute the ϵ-Nash equilibrium, making the theoretical findings accessible for practical scenarios. This module can handle diverse Tullock contest configurations, enabling efficient analysis and solution generation.

Implications

The results have both theoretical and practical significance. Theoretically, this work advances the understanding of how elasticity conditions influence the tractability of Tullock contests. The intractable result highlights the intrinsic difficulty of certain cases, emphasizing the need for approximation techniques. Practically, the proposed algorithms and their implementation provide a valuable tool for analyzing strategic interactions in decentralized systems, including blockchain applications.

These findings also shed light on the broader relationship between game theory and computational complexity. The results illustrate how structural properties of the game—such as the nature of the production function—can dramatically affect the feasibility of equilibrium computation. This underscores the importance of tailoring algorithmic solutions to the specific characteristics of the problem at hand.

Open Question

An important open question remains: when a single contestant or a constant number of contestants have elasticity parameters $r_i \in (1, 2]$, is it possible to compute a pure Nash equilibrium in polynomial time? While our results establish the hardness of the general case with many contestants in this range, the scenario with only one or a constant number of such r_i values may represent a tractable special case. Resolving this question could illuminate the precise boundary between computational feasibility and infeasibility in Tullock contests.

Appendix A. Proof of Proposition 3

Consider a contestant i with $r_i > 2$. If contestant i is an active player in the equilibrium, then its action share σ_i must satisfy $\sigma_i \geq \frac{r_i - 1}{r_i}$, as $k(\overline{A}_i; a_i) = \frac{r_i - 1}{r_i}$. Given that $\frac{r_i - 1}{r_i} > \frac{1}{2}$ for $r_i > 2$, it follows that the action share σ_i for any active contestant i would exceed $\frac{1}{2}$.

Since the sum of the action shares for all active contestants must equal 1 in a pure Nash Equilibrium, it is impossible for more than one contestant to have an action share $\sigma_i > \frac{1}{2}$. Because the number of the contestant $n \geq 2$, if a pure Nash Equilibrium exists, it must be the case that exactly one contestant is active, with this contestant holding an action share of 1, while all other contestants have an action share of 0.

doi.org/10.32388/WPJBOT

However, the optimal action share of 1 leads to aggregation action $A = 0$. Since $\underline{A} > 0$, all other contestants also have an active action share of 1 on $A = 0$.

Hence, a Tullock contest with $r_i > 2$ for all contestants admits no pure Nash equilibrium.

Appendix B. Proof of Theorem 1

The proof is structured based on the classification of contestants into two categories: those with small elasticity ($r_i \in (0,1]$) and those with large elasticity ($r_i \in (2,\infty)$). The assumption that no r_i lies within $(1,2]$ simplifies the problem into three distinct subcases, which we address as follows:

Case 1: All active contestants satisfy $r_i \in (0,1]$**.** This case has been fully analyzed in Section 3.1. The monotonicity of action shares ensures the existence and uniqueness of a pure Nash equilibrium, which can be efficiently computed using the bisection method with polynomial complexity.

Case 2: Active contestants are mixed with $r_i \in (0,1]$ **and** $r_i \geq 2$.

In this case, according to Corollary 1, exactly one contestant satisfies $r_i \geq 2$, while others satisfy $r_i \in (0,1]$. The algorithm proceeds as follows: For each contestant j with $r_j \geq 2$, define the active player set to include j and all contestants with $r_i \in (0,1]$. A binary search is then performed over the feasible interval $[\underline{A_j}, \overline{A_j}]$, where the total contributions of the active players, $S(A) = \sum_{i \in \text{active set}} \sigma_i(A)$, are evaluated. If $S(A) = 1$ for some A^*, an equilibrium is identified. Otherwise, the next contestant with $r_j \geq 2$ is considered. If no equilibrium is found after exhausting all candidates, it implies that no pure Nash equilibrium exists.

Case 3: No equilibrium exists. If none of the above cases yield a valid equilibrium, the exclusion of $r_i \in (1,2]$ ensures that no feasible pure Nash equilibrium configuration exists.

The overall complexity of the algorithm is dominated by the enumeration of $r_j \geq 2$ contestants and the binary search within each case, resulting in a total complexity of $O(n^2 \log(A_{\max}))$, which is polynomial.

Appendix C. Proof of Lemma 2

To prove that Subset Sum with Bounded Maximum (SSLT) can not be solved in polynomial time, assuming the ETH, we show that SSLT is intractable via a quasi-polynomial time reduction from the classic Subset Sum problem.

Subset Sum is defined as follows:

Definition 7. *Given a set of integers* $\mathbf{Z} = \{z_1, z_2, \ldots, z_n\}$ *and a target* T*, determine whether there exists a subset* $S \subseteq \mathbf{Z}$ *such that* $\sum_{z \in S} z = T$*.*

We construct a reduction that uses a hypothetical polynomial-time oracle for SSLT to solve Subset Sum.

The reduction works by recursively breaking the Subset Sum instance into smaller problems and invoking the SSLT oracle when certain conditions are met. Specifically, the algorithm operates as follows:

```
Function SubsetSum(Z', T')
if T' < 0 or Z' is empty then
    return False
if T' == 0 then
    return True
max_element ← max(Z')
if T' ≥ 2 · max_element then
    return SSLT(Z', T') {Call SSLT oracle}
Partition Z' into Z_L = {z ∈ Z' | z < T'/2} and Z_H = {z ∈ Z' | z ≥ T'/2}
for z_h in Z_H ∪ {None} do
    if z_h == None then
        if Call SubsetSum(Z_L, T') then
            return True
    else
        new_target ← T' − z_h
        if Call SubsetSum(Z_L, new_target) then
            return True
return False
```

Algorithm 1. Subset Sum Recursive Algorithm

The reduction relies on the following observations. When $T' \geq 2 \max_{z \in \mathbf{Z}'} z$, the condition for SSLT is satisfied, and the oracle can directly solve the problem. When $T' < 2 \max_{z \in \mathbf{Z}'} z$, the set $\mathbf{Z}'$ is split into two subsets: large elements ($z \geq T'/2$) and small elements ($z < T'/2$). Since at most one large element can be included in a valid subset summing to T', we enumerate the possibilities for including or excluding each large element. The recursive calls then focus on the small elements, progressively reducing the target T'.

At each recursive step, T' decreases by at least half, ensuring logarithmic recursion depth. For each level of recursion, at most $|\mathbf{Z}'|$ subsets are considered. Thus, the total complexity of the algorithm, assuming a polynomial-time SSLT oracle, is $O(n^{\log T})$, which is quasi-polynomial.

doi.org/10.32388/WPJBOT

This reduction implies that if SSLT were solvable in polynomial time, Subset Sum could be solved in quasi-polynomial time. Under the ETH, no $2^{o(n)}$-time algorithm exists for NP-complete problems like Subset Sum. Hence, the existence of a quasi-polynomial time algorithm for SSLT would contradict ETH. This establishes that SSLT can not be solved in polynomial time.

Therefore SSLT can not be solved in polynomial time, assuming the Exponential Time Hypothesis.

Appendix D. Proof of Lemma 1

Consider an instance of the Subset Sum with Bounded Maximum problem defined by a set of positive integers $Z = \{z_1, \ldots, z_n\}$ where $\max_{z \in Z} z \leq \frac{1}{2}\bar{z}$, and a target sum $\bar{z}$. Let $z' = \min_{i \in \{1,\ldots,n\}} z_i$. We construct a corresponding Tullock contest with $n+1$ contestants, where the contest parameters are defined as follows:

$$\begin{cases} r_i = \frac{1}{1-\frac{z_i}{\bar{z}}} & \text{for~} i \in \{1,\ldots,n\} \\ r_i = \frac{1}{\frac{z'}{\bar{z}}-\epsilon} & \text{for~} i = n+1 \end{cases}$$

Given that $\max_{z \in Z} z \leq \frac{1}{2}\bar{z}$, it follows that $r_i \in (1,2]$ for all $i \in \{1,\ldots,n+1\}$.

The corresponding production function parameters are defined as:

$$a_i = \begin{cases} \frac{\bar{z}}{R^{r_i} \cdot (\frac{r_i-1}{r_i})^{r_i-1}} & \text{for~} i \in \{1,\ldots,n\} \\ \frac{\bar{z}}{R^{r_i} \cdot \frac{(r_i-1)^{r_i-1}}{}} & \text{for~} i = n+1 \end{cases}$$

We begin by considering the constructed Tullock contest, where the goal is to show that this contest corresponds to an instance of the Subset Sum with Bounded Maximum problem. Specifically, we aim to prove that there is a Nash Equilibrium (NE) in the Tullock contest if and only if the Subset Sum problem has a solution.

First, consider the case where the Subset Sum with Bounded Maximum problem has no solution. In the corresponding Tullock contest, let us examine the instance we have constructed.

$$\begin{cases} \overline{A_i} = R^{r_i} \cdot (\frac{r_i-1}{r_i})^{r_i-1} \cdot a_i = \bar{z} & \text{for~} i \in \{1,\ldots,n\} \\ \underline{A}_{n+1} = R^{r_i} \cdot \frac{(r_i-1)^{r_i-1}}{} \cdot a_i = \bar{z} & \text{for~} i = n+1 \end{cases}$$

Let A^* represent the aggregate action of all contestants.

We establish that if $A^* \neq \bar{z}$, there cannot be a Nash Equilibrium.

If $A^* < \bar{z}$:

doi.org/10.32388/WPJBOT

In this case, the action share of contestant $n+1$, denoted by σ_{n+1}, satisfies $\sigma_{n+1} = k(A^*; a_{n+1})$. Since the function $k(A; a_i)$ is decreasing on the interval $[0, \overline{A}_{n+1}]$, it follows that $k(A^*; a_{n+1}) > k(\overline{A}_{n+1}; a_{n+1}) = \frac{r_{n+1}-1}{r_{n+1}}$. Given that $\frac{r_{n+1}-1}{r_{n+1}} = 1 - \frac{z'}{\bar{z}} + \epsilon > 1 - \frac{z'}{\bar{z}}$, we have $\sigma_{n+1} \in [1 - \frac{z'}{\bar{z}} + \epsilon, 1)$.

For the other contestants $i \in \{1, \dots, n\}$, the action share σ_i is either $k(A^*; a_i)$ or 0. Since $k(A^*; a_i) > k(\overline{A}_i; a_i) = \frac{r_i - 1}{r_i} = \frac{z_i}{\bar{z}} \geq \frac{z'}{\bar{z}}$. The sum of action shares either exceeds one or equals $k(A^*; a_{n+1})$, but never equals one. This implies that no Nash equilibrium can exist when $A^* < \bar{z}$.

If $A^* > \bar{z}$:

Here, $\sigma_i = 0$ for all $i \in \{1, \dots, n\}$, meaning only contestant $n+1$ could potentially be active. Since this scenario implies $\sum_{i \in I^A} \sigma_i < 1$, no equilibrium exists when $A^* > \bar{z}$.

By construction, the aggregate action A^* must equal $\bar{z}$ for an equilibrium to exist. This is because any deviation from $\bar{z}$ would result in either an excess or deficit in the total action shares σ_i, which would violate the condition $\sum_{i \in I^A} \sigma_i = 1$.

Next, we focus on the case where $A^* = \bar{z}$. In this scenario, if there is a Nash Equilibrium, we must show that contestant $n+1$ cannot be an active participant. Given the construction, contestant $n+1$ has a σ_{n+1} value that satisfies $\sigma_{n+1} > 1 - \min_{i \in n} \sigma_i$. If contestant $n+1$ were to participate, their σ would exceed the allowable action share, thereby preventing the existence of an NE. Thus, contestant $n+1$ must be inactive in any equilibrium.

With contestant $n+1$ excluded, the problem reduces to determining whether the remaining n contestants can collectively form an equilibrium with $A = \bar{z}$. Each of the remaining n contestants has the option to set their σ_i value to either $k(A^*; a_i)$ or 0, which directly corresponds to the choice of including or excluding an element in the Subset Sum problem. Therefore, finding an equilibrium among these n contestants is equivalent to solving the Subset Sum problem: determining whether a subset $S \subseteq Z$ exists such that $\sum_{z \in S} z = \bar{z}$.

Consequently, the existence of a Nash Equilibrium in the Tullock contest corresponds exactly to the existence of a solution to the Subset Sum with Bounded Maximum problem. If the Subset Sum problem has a solution, the corresponding Tullock contest will have a Nash Equilibrium, and if no solution exists, no such equilibrium will be found in the contest.

Therefore, we have shown that the Tullock contest instance has a Nash Equilibrium if and only if the corresponding Subset Sum with Bounded Maximum problem has a solution.

Appendix E. Proof of Theorem 2

To prove Theorem 2, we show that determining the existence of a pure Nash equilibrium (PNE) in a Tullock contest with medium elasticity r_i remains hardness even when additional contestants with small or large elasticity r_i are introduced.

Consider a known hard instance of the PNE problem where all $r_i \in (1, 2]$. We construct a new Tullock contest by adding contestants with $r_j \leq 1$ and $r_k > 2$, assigning their production parameters a_j and a_k arbitrarily small values ($a_j, a_k \to 0$). This ensures these new contestants have negligible impact on the contest outcome, effectively excluding them from the equilibrium computation.

In this modified contest, the equilibrium conditions remain governed by the original contestants with $r_i \in (1, 2]$, as the contributions from the newly added contestants are negligible. Consequently, solving for a PNE in the modified contest reduces directly to solving the original hard problem. Thus, the problem of determining the existence of a PNE in this generalized setting is also hardness.

Appendix F. Proof of Lemma 3

Because σ_i and A are one-to-one mappings on the interval $(v_k, v_{k+1}]$ for all $i \in I$, we rewrite $b_2(A, \sigma_i; t_i) = 0$ as $\sigma_i = \sigma(A, t_i)$ for notation convenience.

Let $\{A^*, I_A, \sigma^*\}$ be the ϵ-Nash equilibrium, and let the two nodes next to A^* be A_1 and A_2. Then $A_1 < A^* < A_2$.

By the definition of the ϵ-Nash equilibrium,

$$\sum_{i \in I_A} \sigma(A^*, t_i) \in (1 - \epsilon, 1 + \epsilon).$$

Because $\sigma(\cdot)$ is decreasing in A,

$$\sum_{i \in I_A} \sigma(A_2, t_i) < \sum_{i \in I_A} \sigma(A^*, t_i) < \sum_{i \in I_A} \sigma(A_1, t_i).$$

We show that either A_1 or A_2 constitutes an ϵ-Nash equilibrium by contradiction. If neither A_1 nor A_2 constitutes an ϵ-Nash equilibrium, then

$$\sum_{i \in I_A} \sigma(A_2, t_i) \leq 1 - \epsilon, \quad \sum_{i \in I_A} \sigma(A_1, t_i) \geq 1 + \epsilon,$$

which implies

doi.org/10.32388/WPJBOT

$$\sum_{i \in I_A} \sigma(A_1, t_i) - \sum_{i \in I_A} \sigma(A_2, t_i) \geq 2\epsilon.$$

On the other hand,

$$\sum_{i \in I_A} \sigma(A_1, t_i) - \sum_{i \in I_A} \sigma(A_2, t_i) \leq \sum_{i \in I_A} \delta\rho \leq n\delta\rho < \epsilon.$$

This is a contradiction.

Appendix G. Proof of Theorem 3

We prove Theorem 3 by constructing a polynomial-time algorithm (Algorithm 3) to compute an ϵ-approximate solution. The algorithm systematically partitions the range of aggregate actions A and verifies potential solutions within each interval. Let $\underline{A}$ and $\bar{A}$ be as defined earlier. The algorithm examines the three distinct intervals of A:

Interval $(0, \underline{A}]$: In this interval, all contestants, including those in $\mathcal{I}^2$, participate. Since the active player set is fixed, a PNE can be efficiently identified using binary search, which also guarantees an ϵ-approximate solution with polynomial time.

Interval $(\bar{A}, \infty)$: In this interval, no contestant in $\mathcal{I}^2$ participates. The active player set in $\mathcal{I}^1$ is determinable, and binary search can again be employed to find a PNE, ensuring an ϵ-approximate solution with polynomial time.

Interval $(\underline{A}, \bar{A}]$: To address this, a set of candidate nodes is constructed for verification. This set, denoted as $\mathcal{A}$, consists of two subsets. The first subset, A_1, comprises nodes that are evenly spaced across $(\underline{A}, \bar{A}]$ with a spacing of $\delta > 0$, formally defined as $A_1 = \{A | A = \underline{A} + k\delta, A \leq \bar{A}, k \in \mathbb{N}^+\}$. The second subset, A_2, includes points $\{\underline{A}_i, \bar{A}_i\}_{i \in \mathcal{I}^2}$, where the best response correspondence is discontinuous. Incorporating these points ensures that no potential ϵ-approximate solution is missed due to such discontinuities. The combined set $\mathcal{A}$ is the union of A_1 and A_2, sorted in ascending order: $\mathcal{A} = \text{sorted}(A_1 \cup A_2)$. The total number of nodes in $\mathcal{A}$ is bounded by $|\mathcal{A}| = 2n_2 + \frac{\bar{A} - \underline{A}}{\delta}$.

By Lemma 3, if there exists an ϵ-approximate solution in the interval, then at least one of its neighboring nodes constitutes an ϵ-approximate solution. So that it's enough to only verify the selected nodes to determine the existence of ϵ-approximate solution. This is because if we choose δ cleverly, how much the sum of the action shares can descend between two adjacent nodes won't exceed ϵ.

doi.org/10.32388/WPJBOT

Lastly, we explain the process of verifying an ϵ-approximate solution at each selected node A_i in polynomial time. For illustration, consider a node $A_i \in (\underline{A}_2, \overline{A}_1]$. We begin by computing the action shares of the active players from contestants in $\mathcal{I}^1$, as well as the action shares of active players from $\mathcal{I}^2$. These active players' action shares are denoted collectively as the set S_0.

Next, we analyze the remaining contestants in $\mathcal{I}^2$. For these contestants, we can exclude those who are guaranteed not to be active players based on their properties. For the remaining k contestants in $\mathcal{I}^2$, each can either participate ($\sigma_i > 0$) or not ($\sigma_i = 0$), leading to 2^k possible combinations of action shares. Since this exponential growth is computationally infeasible for large k, a direct enumeration of all combinations is impractical.

To compute an ϵ-approximate solution in polynomial time, we instead consider the scenario where all k remaining players participate. Specifically, let $S = \{\sigma_i, \cdots, \sigma_j\}$ denote the set of their action shares, where each share satisfies $b_2(A, \sigma_i; a_i) = 0, \cdots, b_2(A, \sigma_j; a_j) = 0$. The problem then reduces to determining whether there exists a subset of S such that the total sum of its elements, combined with S_0, falls within the range $(1 - \epsilon, 1 + \epsilon)$.

To solve this subset selection problem, we use the *Approximate Subset Sum Algorithm*, a well-established algorithm in the literature (e.g., [32]), which solves this problem efficiently in time $O(n/\epsilon)$. The key idea of this algorithm is to dynamically construct multiple subsets while systematically discarding intermediate subsets whose current sums are very close to one another. This pruning strategy ensures that the algorithm maintains computational feasibility by keeping the number of subsets manageable.

By applying this algorithm, we efficiently verify whether a given node A_i admits an ϵ-approximate solution, thereby making the computation of approximate solutions feasible within polynomial time. This method leverages the special structure of the Tullock contest and provides a practical approach for handling otherwise computationally intractable scenarios.

Algorithm 4 describes the verification of ϵ-approximate solution. One thing to notice is, at a given aggregated action A, there can be active players, non-active players, and uncertain players. So that the initial subset should include all participating players, and the selection happens among uncertain players.

In summary, if there exists an (exact) Nash equilibrium, our algorithm will output at least one ϵ-approximate solution, and it's enough to only verify a set of selected nodes in the order of $O(n/\epsilon)$.

Moreover, the time for each verification is in $O(n/\epsilon)$, so that our algorithm runs in time $O(n^2/\epsilon^2)$.

Appendix H. Proof of Lemma 4

We prove that the action share σ_i and the utility function u_i are Lipschitz continuous with respect to the aggregate action A, considering contestants in both $\mathcal{I}^2$ and $\mathcal{I}^1$.

Lipschitz continuity of σ_i with respect to A

First, we trying to prove that the action share function $\sigma_i = k_2(A; a_i)$ for players in $\mathcal{I}^2$ is Lipschitz continuous with respect to A when $A \in [\underline{A}, \overline{A}_i) \cup (\overline{A}_i, \overline{A}]$. For $A \in (\overline{A}_i, \overline{A}]$, $\sigma = 0$ is a constant. And for $A \in [\underline{A}, \overline{A}_i)$, the action share σ_i is defined implicitly by the equation: $k_2(A, \sigma_i) = a_i R^{r_i} r_i^{r_i} (1-\sigma_i)^{r_i} \sigma_i^{r_i - 1} - A = 0$.

To show Lipschitz continuity, we need to establish that there exists a constant $L > 0$ such that for any $A_1, A_2 \in [\underline{A}, \overline{A}_i)$, the following inequality holds:

$$|\sigma_{i,A_1} - \sigma_{i,A_2}| \leq L|A_1 - A_2|.$$

The Implicit Function Theorem ensures that σ_i is differentiable with respect to A if $k_2(A, \sigma_i)$ is continuously differentiable and $\frac{\partial k_2}{\partial \sigma_i} \neq 0$. Differentiating $k_2(A, \sigma_i) = 0$ with respect to A, we obtain:

$$\frac{d\sigma_i}{dA} = -\frac{\frac{\partial k_2}{\partial A}}{\frac{\partial k_2}{\partial \sigma_i}}.$$

The partial derivatives are:

$$\frac{\partial k_2}{\partial \sigma_i} = a_i R^{r_i} r_i^{r_i} \left[-r_i (1-\sigma_i)^{r_i - 1} \sigma_i^{r_i - 1} + (r_i - 1)(1-\sigma_i)^{r_i} \sigma_i^{r_i - 2} \right], \quad \frac{\partial k_2}{\partial A} = -1.$$

Thus, the derivative of σ_i with respect to A is:

$$\frac{d\sigma_i}{dA} = \frac{1}{\frac{\partial k_2}{\partial \sigma_i}}.$$

Analysis of $|\frac{\partial k_2}{\partial \sigma_i}|$: The behavior of $\frac{\partial k_2}{\partial \sigma_i}$ depends on $\sigma_i \in \left[\frac{r_i - 1}{r_i}, \overline{\sigma_i}\right]$, where $\overline{\sigma_i}$ is the value of σ_i corresponding to $A = \underline{A}$. Within this interval, $\frac{\partial k_2}{\partial \sigma_i}$ is strictly negative, ensuring $\frac{d\sigma_i}{dA}$ is well-defined and the value of it first decreases and then increases. Therefore, we evaluate $|\frac{\partial k_2}{\partial \sigma_i}|$ at the endpoints of this interval to determine its minimum absolute value.

At $\sigma_i = \frac{r_i - 1}{r_i}$: Substituting $\sigma_i = \frac{r_i - 1}{r_i}$ into $\frac{\partial k_2}{\partial \sigma_i}$, we have:

doi.org/10.32388/WPJBOT

$$\frac{\partial k_2}{\partial \sigma_i} = a_i R^{r_i} r_i^{r_i} \left[-r_i \cdot r_i^{1-r_i} \left(\frac{r_i - 1}{r_i} \right)^{r_i - 1} + (r_i - 1) \cdot r_i^{-r_i} \left(\frac{r_i - 1}{r_i} \right)^{r_i - 2} \right].$$

Simplifying further:

$$\frac{\partial k_2}{\partial \sigma_i} = -a_i R^{r_i} \cdot \frac{(r_i - 1)^{r_i - 1}}{r_i}, \quad \left| \frac{\partial k_2}{\partial \sigma_i} \right| \geq \frac{3}{5} \cdot \left(\frac{2}{3} \right)^{2/3} \cdot a_i R^{r_i}.$$

Let $c_1 = \frac{3}{5} \cdot \left(\frac{2}{3} \right)^{2/3} \cdot a_i R^{r_i}$.

At $\sigma_i = \overline{\sigma_i}$: Substituting $\sigma_i = \overline{\sigma_i}$ into $\frac{\partial k_2}{\partial \sigma_i}$, we get:

$$\frac{\partial k_2}{\partial \sigma_i} = \underline{A} \left(\frac{-r_i}{1 - \overline{\sigma_i}} + \frac{r_i - 1}{\overline{\sigma_i}} \right) = \underline{A} \cdot \frac{r_i + \overline{\sigma_i} - 2r_i \overline{\sigma_i} - 1}{(1 - \overline{\sigma_i})\overline{\sigma_i}}.$$

The numerator equals zero when $\overline{\sigma_i} = \frac{r_i - 1}{2r_i - 1}$. Given $\underline{A} \leq \underline{A}_i$, $\overline{\sigma_i} \geq \sigma_{i, \underline{A}_i}$. When $A = \underline{A}_i$, solving the implicit equation for σ_i yields:

$$\sigma_i = \frac{1 + \sqrt{1 - 4e^{\frac{(r_i - 1)\ln(r_i - 1) - 2r_i \ln r_i}{r_i}}}}{2}.$$

Let $F_1(r_i)$ and $F_2(r_i)$ denote:

$$F_1(r_i) = \frac{1 + \sqrt{1 - 4e^{\frac{(r_i - 1)\ln(r_i - 1) - 2r_i \ln r_i}{r_i}}}}{2}, \quad F_2(r_i) = \frac{r_i - 1}{2r_i - 1}.$$

Both $F_1(r_i)$ and $F_2(r_i)$ are increasing functions. For $r_i > 1$, $F_1(r_i)$ approaches 1, and $F_2(r_i)$ approaches $\frac{1}{2}$ as $r_i \to \infty$. Thus, there exist constants $c_2 \in [0, \frac{1}{2})$ and c_3 such that:

$$\bar{\sigma}_i > \frac{r_i - 1}{2r_i - 1} + c_2, \quad \text{and} \quad \left| \frac{\partial k_2}{\partial \sigma_i} \right| > \underline{A} \cdot c_3.$$

Since $\underline{A} = \min_{i \in \mathcal{I}^2} \underline{A}_i$, and using bounds for $\bar{\sigma}_i$, we can show:

$$\left| \frac{\partial k_2}{\partial \sigma_i} \right| \geq \underline{A} \cdot c_3,$$

where $c_3 > 0$ is a constant dependent on r_i.

Minimum of $\left| \frac{\partial k_2}{\partial \sigma_i} \right|$: The minimum of $\left| \frac{\partial k_2}{\partial \sigma_i} \right|$ over $\sigma_i \in \left[\frac{r_i - 1}{r_i}, \bar{\sigma}_i \right]$ is given by:

$$C = \min(c_1, \underline{A} \cdot c_3).$$

Since $\left| \frac{\partial k_2}{\partial \sigma_i} \right| \geq C > 0$, the Lipschitz constant for σ_i with respect to A is:

$$L_1 = \frac{1}{C}, \quad L_1 = \max \left(\frac{1}{c_1}, \frac{1}{\underline{A} \cdot c_3} \right).$$

 doi.org/10.32388/WPJBOT

For any $A_1, A_2 \in [\underline{A}, \overline{A}_i)$, we have:

$$|\sigma_{i,A_1} - \sigma_{i,A_2}| \leq L_1 \, |A_1 - A_2| \, .$$

Thus, σ_i is Lipschitz continuous with respect to A on $[\underline{A}, \overline{A}_i) \cup (\overline{A_i}, \overline{A}]$ with constant L_1.

Next, we trying to prove that the action share function $\sigma_i = k_1(A; a_i)$ for players in $\mathcal{I}^1$ is Lipschitz continuous with respect to A when $A \in [\underline{A}, \overline{A}]$, σ_i satisfies:

$$k_2(\sigma_i, A) = (1-\sigma_i)R - \frac{A}{r_i a_i}\left(\frac{\sigma_i A}{a_i}\right)^{\frac{1}{r_i}-1} = 0.$$

Similarly, using the implicit function theorem, we compute:

$$\frac{\partial k_2}{\partial \sigma_i} = -R - \frac{A^2}{r_i a_i^2}\beta\sigma_i^{\beta-1} \leq -R, \quad \frac{\partial k_1}{\partial A} = \frac{(1-\sigma_i)R}{A r_i} \leq \frac{(1-\frac{r_i-1}{r_i})R}{\underline{A r}}$$

The magnitude of $\frac{\partial k_1}{\partial \sigma_i}$ remains bounded below by a positive constant due to the bounded range of σ_i. Hence:

$$\left|\frac{d\sigma_i}{dA}\right| \leq \frac{1}{\underline{A r}^2},$$

where $\underline{r} = \min_{i\in\mathcal{I}^1} r_i > 0$. Thus, σ_i is Lipschitz continuous with constant $L_2 = \frac{1}{\underline{A r}^2}$ on $A \in [\underline{A}, \overline{A}]$.

Lipschitz continuity of the utility function with respect to A. The utility function is given by:

$$u_{i,A} = \sigma_{i,A}R - \left(\frac{\sigma_{i,A}A}{a_i}\right)^{1/r_i}.$$

Using the triangle inequality:

$$|u_{i,A_1} - u_{i,A_2}| \leq |\sigma_{i,A_1} - \sigma_{i,A_2}|R + \left|\left(\frac{\sigma_{i,A_1}A_1}{a_i}\right)^{1/r_i} - \left(\frac{\sigma_{i,A_2}A_2}{a_i}\right)^{1/r_i}\right|.$$

For the first term, since $\sigma_{i,A}$ is Lipschitz continuous, there exists a constant $L_\sigma = \max(L_1, L_2)$ such that:

$$|\sigma_{i,A_1} - \sigma_{i,A_2}| \leq L_\sigma|A_1 - A_2| \Rightarrow |\sigma_{i,A_1} - \sigma_{i,A_2}|R \leq L_\sigma R|A_1 - A_2|.$$

For the second term, consider $p(x) = x^{1/r_i}$. Its derivative is:

$$p'(x) = \frac{1}{r_i}x^{1/r_i-1}.$$

doi.org/10.32388/WPJBOT

Over the range $x \in \left[\frac{\bar{\sigma}_i \underline{A}}{a_i}, \frac{\underline{\sigma}_i \bar{A}}{a_i}\right]$, the derivative is bounded. Let $L_p > 0$ denote the bound of $p'(x)$. Using the mean value theorem:

$$\left|\left(\frac{\sigma_{i,A_1} A_1}{a_i}\right)^{1/r_i} - \left(\frac{\sigma_{i,A_2} A_2}{a_i}\right)^{1/r_i}\right| \leq L_p \left|\frac{\sigma_{i,A_1} A_1}{a_i} - \frac{\sigma_{i,A_2} A_2}{a_i}\right|.$$

Substituting:

$$\frac{1}{a_i}|\sigma_{i,A_1} A_1 - \sigma_{i,A_2} A_2| \leq |\sigma_{i,A_1}||A_1 - A_2| + |A_2||\sigma_{i,A_1} - \sigma_{i,A_2}|$$

Since $\sigma_{i,A}$ and A are bounded, let $i\sigma_{i,A} i \leq \bar{\sigma}_i$ and $iAi \leq \bar{A}$.

$$\left|\frac{\sigma_{i,A_1} A_1}{a_i} - \frac{\sigma_{i,A_2} A_2}{a_i}\right| \leq \frac{1}{a_i}(\bar{\sigma}_i + \bar{A} L_\sigma)|A_1 - A_2|,$$

where $\bar{\sigma}_i$ and $\bar{A}$ are bounds on $\sigma_{i,A}$ and A, respectively. Combining these results, we obtain:

$$|u_{i,A_1} - u_{i,A_2}| \leq L_\sigma R|A_1 - A_2| + L_p \cdot \frac{1}{a_i}(\bar{\sigma}_i + \bar{A} L_\sigma)|A_1 - A_2|.$$

Define the Lipschitz constant for $u_{i,A}$ as:

$$L_u = L_\sigma R + L_p \cdot \frac{1}{a_i}(\bar{\sigma}_i + \bar{A} L_\sigma).$$

Thus:

$$|u_{i,A_1} - u_{i,A_2}| \leq L_u|A_1 - A_2|,$$

Therefore, the utility function is Lipschitz continuous on A.

Appendix I. Proof of Proposition 4

We prove Proposition 4 by analyzing the aggregate action A^* corresponding to a PNE and demonstrating that the ϵ-solution identified by Algorithm 3 constitutes an $(L\epsilon)$-Nash equilibrium.

Case 1: $A^* < \underline{A}$ or $A^* > \bar{A}$

In this scenario, Algorithm 3 directly computes the PNE. By definition, an exact PNE trivially satisfies the conditions of an $(L\epsilon)$-Nash equilibrium, thus establishing the proposition in this case.

Case 2: $A^* \in [\underline{A}, \bar{A}]$

When A^* lies within the range $[\underline{A}, \bar{A}]$, we split the proof into two subcases.

doi.org/10.32388/WPJBOT

Subcase 2.1: A^* is on the boundary of some contestant $i \in \mathcal{I}^2$ ($A^* = \underline{A_i}$ or $A^* = \bar{A}_i$). Since Algorithm 3 enumerates all the boundaries of contestant $i \in \mathcal{I}^2$, any A^* that falls exactly on one of these boundaries will be directly found as an output by the algorithm. In this case, the equilibrium is trivially an $L\epsilon$-Nash equilibrium.

Subcase 2.2: A^* is not on the boundary of any contestant $i \in \mathcal{I}^2$ ($A^* \neq \underline{A_i}$, $A^* \neq \bar{A}_i$). In this scenario, we aim to show that there exists an A' close to A^*, where A' serves as an ϵ-approximate solution. By Lemma 3, A' will be covered within the set of candidate nodes examined by Algorithm 3. Furthermore, according to Lemma 4, the action share for each contestant is Lipschitz continuous with respect to A, and we know that the action share function of the active players is monotonic. Therefore, there exists an $A' \in (A^* - \frac{\epsilon}{L_\sigma n}, A^* + \frac{\epsilon}{L_\sigma n})$ such that the active players under A' are the same as those under A^*, and the sum of action shares satisfies $\sum_i \sigma_{i,A'} \in (1-\epsilon, 1+\epsilon)$.

By Lemma 4, the utility function is Lipschitz continuous with respect to A, and thus the deviation in utilities between A^* and A' is bounded by:

$$u_{i,A'} - u_{i,A^*} < L_u |A' - A^*| < L_u \frac{\epsilon}{L_\sigma n} = L\epsilon.$$

In such case, the approximate solution A' found by the algorithm is an $(L\epsilon)$-Nash equilibrium.

Above all, we have shown that if a pure Nash equilibrium exists, at least one of the ϵ-approximate solutions identified through Algorithm 3 constitutes an $(L\epsilon)$-Nash equilibrium.

doi.org/10.32388/WPJBOT

Appendix J. Algorithms

Input: Number of players n, reward R, a vector of efficiency a, a vector of elasticity r.
Output: A pure Nash equilibrium list Y if it exists, or a statement that no PNE exists.

$Y \leftarrow$ nil // initialize an empty list to store outputs
$\mathcal{I}^A \leftarrow \mathcal{I}^1$. // Define the active player set in case 2.
Perform binary seach for $A*$ over $[\max \underline{A_i}, A_{max}]$:
while A^* not found **do**
    Compute $S(A^*) = \sum_{i\in\mathcal{I}^A} \sigma_i$.
    if $S(A^*) = 1$ **then**
        Append the PNE solution to Y.
for each contestant $j \in \mathcal{I}^2$ **do**
    $\mathcal{I}^A \leftarrow \mathcal{I}^1 \cup \{j\}$. // Define the active player set in case 2.
    Perform binary search for A^* over $[\underline{A_j}, \overline{A_j}]$:
    while A^* not found **do**
        Compute $S(A^*) = \sum_{i\in\mathcal{I}^A} \sigma_i$.
        if $S(A^*) = 1$ **then**
            Append the PNE solution to Y.
if There exists solution in Y **then**
    return Y as the equilibrium solution.
return No PNE exists.

Algorithm 2. Identifying PNE in Mix Regime

Input: A vector of production efficiency a, number of players n, reward R, a vector of elasticity r, ϵ magnitude and discretization distance δ.
Output: A list of ϵ-approximate solutions Y.
Compute $\{\underline{A_i}, \bar{A}_i\}_{i\in\mathcal{I}^2}$, and let $\mathcal{A}_1 = \bigcup_{i\in\mathcal{I}^2}\{\underline{A_i}, \bar{A}_i\}$;
$\{A_{(1)}, ..., A(2n_2)\} \leftarrow$ sorted($\mathcal{A}_1$) // sort $\mathcal{A}_1$ in ascending order
$\mathcal{A}_2 \leftarrow \{A_{(1)} + k\delta \,|\, k \in \mathbb{N}^+, A_{(1)} + k\delta \leq A_{(2n_2)}\}$.
$\mathcal{A} \leftarrow \mathcal{A}_1 \cup \mathcal{A}_2$ // the set of nodes to be verified
$Y \leftarrow$ nil // initialize an empty list to store outputs
Binary search for ϵ-approximate solutions in $(0, A_{(1)}], [A_{(2n_2)}, \infty]$ and append to Y if they exist
for each node in A **do**
    Run `APPROX-SUBSET-SUM` to verify if this node constitutes an approximate solution
    if TRUE **then**
        Append the approximate solution to Y
return Y

Algorithm 3. SEARCH-ϵ-NE(a, n, R, r, ϵ)

Input: S_0: the sum of the action shares of players who participate for sure; $\boldsymbol{\sigma}$: a vector of action shares of uncertain players if they participate; ϵ: approximation parameter.
Output: If there are subsets of $\boldsymbol{\sigma}$ that together with S_0 sum up to between $(1-\epsilon, 1+\epsilon)$, then return at least one of them; otherwise, return nothing.
$n \leftarrow |\boldsymbol{\sigma}|$
$X_0 \leftarrow [S_0]$
$Y_0 \leftarrow [S_0]$
for $i \leftarrow 1$ to n **do**
    $X_i \leftarrow$ `MergeList`$(X_{i-1}, X_{i-1} + \sigma_i)$
    $X_i \leftarrow$ `TRIM-FROM-BELOW`$(X_i, \frac{\epsilon}{2n})$
    $Y_i \leftarrow$ `MergeList`$(Y_{i-1}, Y_{i-1} + \sigma_i)$
    $Y_i \leftarrow$ `TRIM-FROM-ABOVE`$(Y_i, \frac{\epsilon}{2n})$
return all elements in X_n and Y_n that are in $(1-\epsilon, 1+\epsilon)$

Algorithm 4. APPROX-SUBSET-SUM(S_0, σ, ϵ)

$n \leftarrow |L|$
$L \leftarrow \text{sorted}(L)$ // sort L in ascending order
$last \leftarrow y_1$ // y_1 is the first element in L
$L' \leftarrow [y_1]$
for $i \leftarrow 2$ to n **do**
    if $y_i > (1+\delta) \cdot last$ **then**
        Append y_i onto the end of L'.
        $last \leftarrow y_i$
return L'

Algorithm 5. TRIM-FROM-BELOW(L, δ)

$n \leftarrow |L|$
$L \leftarrow \text{sorted}(L)$ // sort L in descending order
$last \leftarrow y_1$ // y_1 is the first element in L
$L' \leftarrow [y_1]$
for $i \leftarrow 2$ to n **do**
    if $y_i < \text{last} \cdot (1-\delta)$ **then**
        Append y_i onto the end of L'.
        $last \leftarrow y_i$
return L'

Algorithm 6. TRIM-FROM-ABOVE(L, δ)

 doi.org/10.32388/WPJBOT

Footnotes

[1] This is mainly to avoid uninteresting corner cases when $R^{r_i} \to 0$ for large r_i; other than this case, all our techniques and insights carry through.

References

1. a, b, c *Tullock G. Efficient rent-seeking. Toward a Theory of the Rent-Seeking Society. 1980.*
2. ^ *Jia H, Skaperdas S, Vaidya S (2013). "Contest functions: Theoretical foundations and issues in estimation". International Journal of Industrial Organization. 2013.*
3. ^ *Lim W, Matros A, Turocy TL (2014). "Bounded rationality and group size in Tullock contests". Journal of Economic Behavior & Organization.*
4. ^ *Luo T, Kanhere SS, Tan HP, Wu F (2015). "Crowdsourcing with Tullock contests: A new perspective". IEEE Transactions on Mobile Computing.*
5. a, b *Cong LW, He Z, Li J (2018). "Decentralized mining in centralized pools". The Review of Financial Studies.*
6. ^ *Thum M. "The economic cost of Bitcoin mining". CESifo Forum. 2018.*
7. a, b *Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008.*
8. ^ *Kroll JA, Davey IC, Felten EW. "The economics of Bitcoin mining or, Bitcoin in the presence of adversaries". In: Proceedings of WEIS; 2013.*
9. a, b, c, d *Easley D, O'Hara M, Basu S (2019). "From mining to markets: The evolution of Bitcoin transaction fees". Journal of Financial Economics.*
10. ^ *Herzberg J. Essays on Environmental and Resource Economics: Evaluation of Voluntary and Mandatory Policies [PhD thesis]. Aachen: RWTH Aachen University; 2005.*
11. ^ *Ren L, Li Y, Wang R, Zhang S (2024). "Enabling joint distribution with blockchain technology in last-mile logistics". Computers & Industrial Engineering. 2024.*
12. ^ *Fabi M, Kassoul M, Prat J. Systematization of knowledge: constant function market makers. In: Finance, Digital Assets, and Blockchain Applications. Edward Elgar Publishing; 2024.*
13. ^ *Lang P, Tian D, Duan X, Zhou J (2022). "Cooperative computation offloading in blockchain-based vehicular edge computing networks". IEEE Transactions on Vehicular Networks.*

14. ^Canidio A (2018). "Financial incentives for the development of blockchain-based platforms". MPRA Pa per.
15. ^Boumaiza A. Towards a Blockchain-enabled Transactive Renewable Energy Trading Market. In: IEEE I nternational Conference on Smart Energy Systems; 2024.
16. ^Carlsten M, Kalodner H, Weinberg SM, Narayanan A. (2016). "On the instability of Bitcoin without the block reward". In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS).
17. a, b, c, d, e, f, g, hCornes R, Hartley R (2005). "Asymmetric contests with general technologies". Economic Theory. 26 (4): 923–946.
18. ^Stoll C, Klaaßen L, Gallersdörfer U (2019). "The carbon footprint of bitcoin". Joule. 3 (7): 1647–1661.
19. ^Klunover D. "Asymmetric technologies in contests". European Journal of Political Economy. 2021.
20. ^Cornes R, Hartley R (2012). "Risk aversion in symmetric and asymmetric contests". Economic Theory. 2012.
21. ^Fu Q, Lu J, Zhang J (2016). "Optimal contest design under asymmetric entry and information disclosur e". Canadian Journal of Economics. 2016.
22. ^Skaperdas S. "Contest success functions". Economic theory. 7: 283–290, 1996.
23. ^Clark DJ, Riis C (1998). "Contest success functions: an extension". Economic Theory. 11: 201–204.
24. ^Luce RD. Individual choice behavior. Vol. 4. New York: Wiley; 1959.
25. ^Baye MR, Hoppe HC (2003). "The strategic equivalence of rent-seeking, innovation, and patent-race g ames". Games and economic behavior. 44 (2): 217–226.
26. a, bRosen JB (1965). "Existence and uniqueness of equilibrium points for concave n-person games". Eco nometrica: Journal of the Econometric Society. pages 520–534.
27. ^Yao F, Liao Y, Liu J, Nie S, Wang Q, Xu H, Wang H (2024). "Unveiling User Satisfaction and Creator Pro ductivity Trade-Offs in Recommendation Platforms". arXiv preprint arXiv:2410.23683.
28. a, bYao F, Li C, Nekipelov D, Wang H, Xu H (2024). "Human vs. Generative AI in Content Creation Compe tition: Symbiosis or Conflict?" arXiv preprint arXiv:2402.15467.
29. ^Even-Dar E, Mansour Y, Nadav U (2009). "On the convergence of regret minimization dynamics in co ncave games". In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 52 3--532.

30. [^]*Szidarovszky F, Okuguchi K (1997). "On the Existence and Uniqueness of Pure Nash Equilibrium in Rent-Seeking Games". Games and Economic Behavior.* ***18*** *(1): 135-140. ISSN 0899-8256.*

31. [^]*Bravo M, Leslie D, Mertikopoulos P (2018). "Bandit learning in concave N-person games". Advances in Neural Information Processing Systems.* ***31****.*

32. [a, b]*Martello S, Toth P. Knapsack problems: algorithms and computer implementations. USA: John Wiley & Sons, Inc.; 1990. ISBN 0471924202.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.