# VCG-Based Mechanism in Multi-Rounds Auction

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Master of Science

by

Yanchen Liu

May 2025

# APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Yanchen Liu

This Thesis has been read and approved by the examing committee:

Advisor: Matheus Venturyne Xavier Ferreira

Advisor:

Committee Member: Chenyu Wei

Committee Member: Weikai Lin

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2025

# VCG-based Mechanism in Multi-rounds Auction

YANCHEN LIU, University of Virginia, USA

Auction mechanisms are designed to address two fundamental issues: determining the winner of a specific item and establishing the payment required from the winner. In multi-round auctions where bidders directly report their valuations, as the scenario we study in this paper, the mechanism must define both the allocation and payment rules to achieve a specific objective, such as maximizing social welfare or revenue. In this study, we focus on identifying a welfare-optimal allocation strategy that ensures items are assigned to the bidders with the highest valuations among all participants.

Regarding the payment rule, while the second-price mechanism is optimal in single-round auctions, the dynamic nature of multi-round auctions introduces a new challenge: ensuring incentive compatibility for every bidder, even when they have complete knowledge of other bidders' strategies. In an ideal mechanism, the winner should pay only the second-highest price among the other bidders in a given round, without being influenced by previous or future rounds. However, a strategic bidder may exploit their knowledge of others' strategies, potentially causing honest bidders to experience regret. This regret is defined as the difference in utility when the strategic bidder bids honestly versus when they bid manipulatively, often at the expense of overall social welfare.

This paper proposes a novel payment rule designed to maximize social welfare with less fluctuation while adhering to the constraints of auction design theory, defined in previous related work. The proposed mechanism ensures that, even if a strategic bidder has complete information about other participants, including their valuations and bidding strategies, they are still incentivized to bid truthfully based on their valuation. The results demonstrate that the mechanism mollifies strategic bidding from all participants, even in the presence of full strategic foresight, thereby optimizing social welfare and balancing utility across bidders.

**ACM Reference Format:**

Yanchen Liu. 2025. VCG-based Mechanism in Multi-rounds Auction. 1, 1 (April 2025), 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Auctions have a long history, dating back to the origins of trade. However, the introduction of multi-round auctions for identical items has revealed incompatibilities between existing payment rules and the goals of incentive compatibility and social welfare optimization. In this paper, we propose a payment rule based on the Vickrey-Clarke-Groves (VCG) mechanism [5], combined with a specific allocation strategy, to encourage bidders to submit their true maximum valuations. This approach not only ensures incentive compatibility but also maximizes social welfare. Notable auction formats include the *First-Price Sealed-Bid Auction*, the *Second-Price Sealed-Bid Auction*, and the *All-Pay Auction* [1].

With the advent of online platforms, auctions have evolved beyond physical events, enabling broader participation and greater efficiency. However, this transition has also introduced new challenges, such as increased opportunities for manipulation and fraud. Exploiting inefficiency in auction protocols or leveraging advanced technological tools, some participants can manipulate outcomes to their advantage, undermining both fairness and efficiency.

To mitigate these issues, modern auction mechanism design emphasizes three key properties: **incentive compatibility**, **individual rationality**, and **computational efficiency**. These properties ensure the effectiveness of an auction mechanism by motivating bidders to truthfully report their valuations rather than manipulating their bids to gain an unfair advantage, even when they possess complete information about other participants. In the context of single-item auctions, the *Second-Price Auction* has been shown as the optimal mechanism for achieving incentive compatibility, as it aligns individual incentives with the optimal social welfare.

Building on this foundation, this paper investigates the design and analysis of auction mechanisms that preserve incentive compatibility while addressing the complexities introduced by multi-round auctions and strategic bidding behaviors. By balancing

Author's Contact Information: Yanchen Liu, University of Virginia, Charlottesville, USA, fty3wb@virginia.edu.


Manuscript submitted to ACM

individual utility maximization with social welfare stability, this study proposes a novel payment rule that integrates the Vickrey-Clarke-Groves (VCG) mechanism. The proposed approach enhances the robustness and fairness of auction systems in multi-round direct revelation auction settings.

### 1.1 Brief Overview

In this paper, we consider an auction scenario consisting of multiple rounds and multiple participants. The bidding strategy matrix is predefined and remains fixed throughout the game, implying that participants cannot alter their bids once the auction begins. Given this specific scenario, we propose an auction mechanism that integrates an allocation rule based on the *Kuhn-Munkres algorithm* [10] and a payment rule derived from the *Vickrey–Clarke–Groves (VCG) mechanism.* The designed mechanism aims to achieve two primary goals: maximizing social welfare and incentivizing honest bidding among participants, thereby ensuring incentive compatibility throughout the auction.

We introduce a metric termed *regret* [8], defined as the difference between the best possible utility a strategic bidder could achieve and the utility obtained through honest bidding. To evaluate the performance of our mechanism, we conduct a simulation comparing it with the sequential second-price mechanism, which is optimal for single-round auctions. The results, illustrated through diagrams and datasets included in the appendices, demonstrate that our mechanism effectively eliminates regret, thus encouraging truthful bidding behavior. Furthermore, due to the properties of the *Kuhn-Munkres algorithm*, the mechanism ensures the maximization of social welfare.

### 1.2 Related Literature

Significant prior research has been conducted to optimize auction mechanisms and systems across various scenarios. Some studies emphasize cryptographic aspects [2], while others aim to achieve equilibrium within auction systems [4]. Building upon concepts discussed in the textbook *Algorithmic Game Theory* by Noam Nisan et al [6], our work specifically addresses sealed-bid strategies with the objective of maximizing social welfare. Maximizing social welfare constitutes one foundational approach to mechanism design, alongside revenue maximization.

Additionally, the work on the FedEx Problem by Amos Fiat et al [3]. offers valuable insights into resource allocation efficiency. Inspired by their multi-round scenario, our study similarly investigates sequential allocation rules. Furthermore, chapter 7 of *Algorithmic Game Theory* [6] introduces the VCG mechanism as an effective payment rule that ensures incentive compatibility, thereby encouraging bidders to bid truthfully. And the preliminary knowledge of this mechanism would be illustrated in detail in the following section.

Although our mechanism design focuses on auctions with a predefined and fixed value matrix, the approach is broadly applicable to payment rules for various resources, including blockchain-based scenarios. Notably, the research conducted by Mallesh M. Pai et al [7]. explores blockchain mechanisms for dynamic transaction fees, highlighting a pertinent area for future exploration.

From an algorithmic perspective, our current method incurs a computational complexity of $O(n^3)$ due to exhaustive enumeration. Optimizing the performance through linear programming techniques [9] represents a promising direction for future research to enhance algorithmic efficiency.

#### 1.2.1 *VCG mechanism.*

The Vickrey-Clarke-Groves (VCG) mechanism is designed based on the reported participant profile $v$, which represents the private information disclosed by each participant in the game. By collecting and processing these participant profiles, we define the function $x$ as the allocation rule, also referred to as the alternative determination function. This function takes the reported profiles as input and determines the available alternatives $a$, formally expressed as:

$$a = x(v). \tag{1}$$

To illustrate the definition of alternatives, let $\mathbf{A}$ denote the set of all available alternatives when all agents participate in the mechanism. For any agent $i$, we define $\mathbf{A}^{-i}$ as the *set of alternatives available when agent $i$ is not present.* A specific alternative $a_i \in \mathbf{A}$ represents a particular arrangement or allocation of the resource being managed.

For example, in a time-slot allocation scenario, the set $\mathbf{A}$ consists of all available time slots, and the mechanism selects a specific time slot $a_i$ for allocation. In an auction scenario as we focus on in this paper, we define $a_i$ as the outcome in which the $i$-th bidder wins the auction and receives the item. The set of alternatives in an auction is given by:

$$\mathbf{A} = \{a_1, a_2, \ldots, a_m\}, \quad m \in \mathbb{N}^+, \tag{2}$$

where $a_i$ represents the allocation of the item to bidder $i$. The feasibility of each trade depends on the participating agents, as only those who place bids are eligible to win the item. More concretely, $a_1$ signifies that the item is assigned to bidder 1, $a_2$ means that the item is assigned to bidder 2, and so forth, up to $a_m$, where the item is assigned to bidder $m$. To further declare our alternatives space, the mechanism is designed based on deterministic allocation rule, where $A \in \{0,1\}^n$ and $\Sigma_{i=1}^n a_i = 1$, in the example above. Obviously, $A^{-i} \subseteq A$.

**Definition (VCG mechanism).** Given reported valuation profile $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n)$, where $\hat{v}_i(i \in [1,n])$ is a function that maps alternatives to real number, representing the value that agent $i$ assigns to each alternative, the *Vickrey-Clarke-Groves (VCG) mechanism* on a set of alternatives $A$ is defined by:

- a choice $x(\hat{v}) \in \arg\max_{a \in A} \sum_{i \in N} \hat{v}_i(a)$, with selected alternative $a^* = x(\hat{v})$
- a payment rule $t_i$, where the payment by agent $i$ is

$$t_i(\hat{v}) = \sum_{j \in N \setminus \{i\}} \hat{v}_j(a^{-i}) - \sum_{j \in N \setminus \{i\}} \hat{v}_j(a^*),$$

  with $a^{-i} \in \arg\max_{a \in A^{-i}} \sum_{j \in N \setminus \{i\}} \hat{v}_j(a)$, where $A^{-i}$ is the set of alternatives when $i$ is not present.

Ties can be broken arbitrarily when choosing $a^*$ or $a^{-i}$ (for any $i$).

**Final Goal**. The goal for this paper is to design an auction mechanism to maximize the social welfare, which could be defined as follow:
Given reported valuation profile $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n)$ and the allocation function $x$, we define the social welfare by summing the utility for each participants:

- the utility for $participant_i$:

$$u_i = u(x(\hat{v}), \hat{v}_i, t_i(\hat{v})) = x(\hat{v}) \times \hat{v}_i - t_i(\hat{v})$$

- social welfare is the summation of the utilities for all participants(including the seller, represented by $u_0 = \Sigma_{i=1}^n t_i(\hat{v})$):

$$\Sigma_{i=0}^n u_i = \Sigma_{i=1}^n x(\hat{v}) \times \hat{v}_i - \Sigma_{i=1}^n t_i(\hat{v}) + \Sigma_{i=1}^n t_i(\hat{v}) = \Sigma_{i=1}^n x(\hat{v}) \times \hat{v}_i$$

In the VCG mechanism regarding auction scenario, agent $i$ pays the *opportunity cost* imposed on the other agents; i.e., the amount by which the total reported value of others is reduced as a result of selecting alternative $a^*$ compared to the alternative $a^{-i}$ that would be selected otherwise. When $a^* = a^{-i}$, agent $i$ is said to be *non-pivotal* and makes no payment; otherwise, agent $i$ is said to be *pivotal* on the decision and we may have $t_i(\hat{v}) \neq 0$.
The VCG mechanism to sell a single item is just the Second-Price-Sealed-Bid(SPSB) auction.

### 1.3 Auction Scenario

Assuming there is a running auction consisting of $m$ rounds, to allocate the identical but multiple items to a series of bidders $N = \{1, ..., n\}$ $(m, n \in \mathbb{N})$, and each bidder has their intrinsic value vector $\hat{v}_i = \{v_{i_1}, v_{i_2}, ... v_{i_m}\}$ ($v_{i_k} \in \mathbb{R}^+$ for all $k \in \{1, 2, \ldots, m\}$, $i \in N$) for bidder $i$ in $m$ rounds. It is worth to note that generally speaking, the valuation for the same item would be decreasing as the time goes up, which is constrained by:

$$v_{i_1} \geq v_{i_2} \geq \ldots \geq v_{i_m}$$

We use a matrix $\mathbf{V} = [\hat{v}_1^\top . \hat{v}_2^\top, \ldots, \hat{v}_n^\top] \in \mathbb{N}^{m \times n}$ to denote the set of intrinsic values for all bidders, which contains all maximum values that bidders could offer for a specific item in each round. For the scenario of the auction in this paper, we narrow the allocation rule to be sealed direct and deterministic revelation, which means the bidder has boolean variables in each round, where 0 means losing and 1 means wining, and in each round, there exists one and only one winner, whose alternative variable is 1, while other bidders' variable are all 0. Other than that, the allocation rule derives based on the intrinsic value matrix gave before the beginning of the auction, which means all bidding values have been assured and fixed before the first round begins, while no new bidders would enter the auction, so the number of participants is fixed.
As the auction mechanism designer, the information known is the bidding values, which are denoted by a $n$-tuple $b_t$ =

$\{b_{t_1}, b_{t_2}, ..., b_{t_n}\}$for bidding value in $t$-th round. Therefore, there is a matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$ recording all bidding values, which is used by designer to allocate the item to the winner in each round. In the scenario we discuss in this paper, the matrix **B** is fixed before the first round starts, until the whole auction finishes. Moreover, because the items are all identical but repetitive for all rounds, the winner for round $t$ won't participate the rounds after $t$, which holds:

$$\begin{cases} \sum_{j=1}^{m} \mathbf{A}_{i,j} \leq 1, \quad \forall i \in \{1,\ldots,n\} \\ \sum_{i=1}^{n} \mathbf{A}_{i,j} \leq 1, \quad \forall j \in \{1,\ldots,m\} \end{cases}$$

Based on the matrix **B**, the goal of the mechanism is to determine the allocation rule set $\mathbf{A} \in [0,1]^{n\times m}$, where 1 denotes the bidder wins and 0 denotes that he doesn't, through which to maximize the social welfare:

$$\mathbb{WELFARE} = \mathbf{V} \times \mathbf{A}$$

After determining the allocation to each winner, the paym ent rule $P_i = \mathbf{Payment}(\mathbf{B}, \mathbf{A})$ ($i \in [1, n]$) is a function based on bidding value matrix **B** and allocation matrix **A**.

In our hypotheses, we assume that every bidder is myopic and selfish. So they will only design their bid strategy $\mathbf{B_{:,i}}$ to maximize their personal utility $u_i$:

$$u_i = \Sigma_{j=1}^{m}\mathbf{V}_{ji} \times \mathbf{A}_{ij} - P_i$$

### 1.4 Mechanism Design Constraints

Every rational auction mechanism should satisfy the following prerequisites: Allocative Efficiency, Incentive Compatibility and Individual Rationality. The definitions are demonstrated as following:

**Definition (Allocation Rule)**. The mechanism ensures that the allocation $\mathbf{A}^* \in \mathbf{A}$ maximizes the total value across all bidders. This is achieved by solving the optimization problem:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} \left(\mathbf{V} \times \mathbf{A}\right),$$

This guarantees that the resources are allocated to those who value them the most, thereby maximizing social welfare.

**Definition (Incentive Compatibility)**. The VCG mechanism incentivizes bidders to truthfully report their valuations. For each bidder $i$, the utility achieved by reporting their true valuation $v_i$ is at least as great as the utility achieved by reporting any other valuation $\hat{v}_i$. Mathematically, this property is expressed as:

$$u_i(b_i = v_i, b_{-i}) \geq u_i(b_i \neq v_i, b_{-i}),$$

where $v_{-i}$ represents the valuations of all other bidders. This property ensures truthful bidding as the optimal strategy.

**Definition (Individual Rationality)**. The VCG mechanism guarantees that the participation for each bidder is better than they are not, ensuring their participation in the auction is rational. Specifically, if a bidder doesn't participate, there is no payment for him. For bidder $i$, this condition is given by:

$$\Sigma_{j=1}^{m}\mathbf{V}_{ji} \times \mathbf{A}_{ij} - P_i \geq \Sigma_{j=1}^{m}\mathbf{V}_{ji} \times \mathbf{A}_{ij}^{-}i = 0$$

where $P_i$ is the payment made by bidder $i$. This implies that no bidder is worse off by participating in the auction.

### 1.5 Potential Problem

In our hypothesis, there exists a strategic bidder $i$ who obtains a crystal ball, which could reveal all bidding information:

$$(\mathbf{V^{-i}}, \mathbf{B^{-i}})$$

where $\mathbf{V^{-i}}$ means the intrinsic valuation matrix without the $i$-th column, and respectively $\mathbf{B^{-i}}$ means the bidding matrix without the $i$-th column, before the auction starts. Bidder $i$ is able to adjust his bidding strategy $\mathbf{B_{:,i}}$ to maximize his utility $u_i$. Other than the strategic bidder $i$, to simplify the scenario, all other bidders are honest bidders, which means they would bid their real value in each round. Thus we have:

$$\mathbf{V^{-i} = B^{-i}}$$

Based on the hypothesis we give above, the strategic bidder might impair the overall social welfare in order to maximize his own utility if he has the crystal ball and know all information before.

**Example** To illustrate the problem we are trying to define, consider the example provided in Table 1, where bidders $A$, $B$, and $C$ participate in a three-round auction. Suppose bidder $A$, as a strategic bidder, the mechanism adopts SPSB payment rule:

- If bidder $A$ bids honestly and wins in the first round, their utility is $u = v_A - \text{second-highest bid} = 22 - 18 = 4$. $\mathbb{WELFARE} = 22 + 12 + 8 = 42$.
- If bidder $A$ delays their win to the second round, their utility becomes $u = v_A - \text{second-highest bid} = 15 - 10 = 5$. $\mathbb{WELFARE} = 18 + 15 + 8 = 41$
- If bidder $A$ delays their win to the third round, their utility becomes $u = v_A - \text{second-highest bid} = 10 - 0 = 10$. $\mathbb{WELFARE} = 18 + 10 + 10 = 38$

This example demonstrates how strategic bidding can impair the social welfare, contravene to the constraint of **Incentive Compatibility**. A robust auction mechanism must, therefore, address these challenges by satisfying all of the constraints to maintain the stable social welfare.

Table 1. Example of Multi-Round Auctions

| Bidder | 1st Round | 2nd Round | 3rd Round |
|---|---|---|---|
| A | 22 | 15 | 10 |
| B | 18 | 12 | 7 |
| C | 15 | 10 | 8 |

Other than the impair to social welfare, simply implementing sequential second price payment rule in the scenario of multi-round auction would introduce regrets, which would be experienced by the strategic bidder due to his dishonest bidding strategy.

**Definition (Regret)**. The regret is defined by the difference between the maximum utility the bidder could achieve and the utility if he bids honestly, given the fixed bidding strategies from all other bidders. We first define $b_i$ which represents the $m$ elements bidding vector for bidder $i$ in this multi-round auction, and $A^*$ represents the optimal allocation given every $b_i$. $V_{-i}$ represents the true value matrix other than bidder $i$ which is fixed when we calculate the regret for bidder $i$ solely. Matrix $V$ is the same as matrix $V_{-i}$ when bidder $i$ bids his true value, as $b_i = v_i$. $P_i$ represents the payment that bidder $i$ needs to pay. So, the regret for bidder $i$ could be defined as:

$$\mathbf{Regret_i} = \arg\max_{\mathbf{b_i}} \mathbf{u_i}\left(\mathbf{b_i}, \mathbf{A^*}, \mathbf{V_{-i}}, \mathbf{P_i}\right) - \mathbf{u_i}(\mathbf{A^*}, \mathbf{V}, \mathbf{P_i}),$$

It's worth noting that the regret is always positive, because any rational person would bid honestly to reach zero regret if there is no spaces to earn more utility.

## 2 Mechanism

In any auction mechanism design, there are two key components to be determined: who wins and how much the winner should pay, corresponding to the two mechanism rules, allocation rules and payment rules. In this section, we will illustrate our rules in detail.

### 2.1 Allocation Rule

Based on the constraints and goals defined in section 1, in order to reach the highest social welfare, which is the summation of all winner's utilities and the seller's utility, we run the **KM algorithm** which derives from **Hungarian algorithm**, to generate the maximum social welfare and the allocation rule which could achieve the maximum utility by giving a cubic matrix.

Initially, KM algorithm is to output the minimum cost in bipartite graph, using dept first search or breadth first search algorithm on a square matrix. If$m \neq n$, we pad the deficient line(or row) with 0, making sure it is square. And then, we transfigure the matrix **B** to be **C=-B**, which leads to output to be the maximum instead of the minimum but the same allocation strategy. Here is the overall scheme for KM algorithm and the pseudocode in 1

**KM Algorithm**

Input: Bidding matrix $\mathbf{B} = [b_{ji}] \in \mathbb{R}^{m\times n}$

Output: Maximum Social Welfare SW and allocation matrix $\mathbf{A} = [a_{ij}] \in \{0,1\}^{n\times m}$

**Algorithm 1** Kuhn–Munkres (Hungarian) Algorithm for Maximum Weight Matching

**Require:** Bidding matrix $\mathbf{B} = [b_{ji}] \in \mathbb{R}^{m\times n}$
**Ensure:** Allocation matrix $\mathbf{A} = [a_{ij}] \in \{0,1\}^{n\times m}$ and maximum social welfare $SW$

1: Pad $\mathbf{B}$ with zeros to make it square if $m \neq n$
2: Construct negative matrix $\mathbf{C} = -\mathbf{B}$
3: Subtract the minimum of each row from all elements in that row
4: Subtract the minimum of each column from all elements in that column
5: **repeat**
6: Cover all zeros in the matrix using a minimum number of horizontal and vertical lines
7: **if** number of lines equals matrix size **then**
8: An optimal allocation is possible
9: **else**
10: Find the smallest uncovered element, subtract it from all uncovered elements, and add it to elements covered twice
11: **end if**
12: **until** an optimal allocation is found
13: Use the zero positions to construct the optimal allocation matrix $\mathbf{A}$
14: Compute $SW = \sum_{i,j} a_{ij} \cdot b_{ji}$

## 2.2 Payment Rule

The amount that the winner need to pay derives from VCG definition, which is the externality(or opportunity cost) that the winner causes due to his winning. The payment value $p_i$ refers to the amount that bidder $i$ needs to pay:

$$p_i = \sum_{c=0}^{m} \sum_{r=1,r\neq i}^{n} v_{c,r} \times a_{r,c}^{-i} - \sum_{c=0}^{m} \sum_{r=1,r\neq i}^{n} v_{c,r} \times a_{r,c}^{*} \tag{3}$$

$$a_{r,c}^{*} \in \mathbf{A}^{*} = \arg \max_{\mathrm{A}\in[0,1]^{n\times m}} \mathbf{V}\cdot\mathbf{A} \tag{4}$$

$$a_{r,c}^{-i} \in \mathbf{A}^{-i*} = \arg \max_{\mathrm{A}^{-\mathrm{i}}\in[0,1]^{(n-1)\times m}} \mathbf{V^{-i}}\cdot\mathbf{A^{-i}} \tag{5}$$

In equation 1, it follows the principle of VCG: what the winner needs to pay is that the summation of values for the rest bidders minus the social welfare if the winner is not present from the round he wins to the end. Also the outcome about the allocation comes from the section 2.1, generally speaking, it derives from the specific allocation matrix that could maximize the current social welfare based on specific scenario. For example, $\mathbf{A^{-i}}$ means the allocation matrix when bidder $i$ is not present, while $\mathbf{A}^{*}$ means the allocation rule that maximizes the social welfare in the scenario when bidder $i$ is present.

**Example** To comprehensively demonstrate the new payment rule in specified multi-rounds auction scenario, we consider the same example provided in Table 2, where the same bidders $X$, $Y$, and $Z$ participate in a three-round auction and their values for each rounds are shown in the table 2. To help denote the meaning for each part in the Equation 3, we define:

$$\begin{cases} S_i^{-} = \sum_{c=0}^{m} \sum_{r=1,r\neq i}^{n} v_{c,r} \times a_{r,c}^{-i} \\ S_i^{*} = \sum_{c=0}^{m} \sum_{r=1,r\neq i}^{n} v_{c,r} \times a_{r,c}^{*} \end{cases}$$

Winners for each should pay:

- After running KM algorithm on table 2, the winners for this three-rounds auction are bidder $X$ for round 1, bidder $Y$ for round 2 and bidder $Z$ for round 3, which leads to the social welfare to be 42, maximum.

- Bidder $X$ is the pivotal bidder and wins in round 1 because removing $X$ from the alternative set $\mathbf{A}$ (denoted as $\mathbf{A}^{-X}$, which consists only of bidders $Y$ and $Z$ in round 1) would change the outcome to $Y$ winning instead (Because when removing bidder $X$ from this game and only considering bidder $Y$ and bidder $Z$ for round 1 and 2, the maximum social welfare is achieved by assigning the item to bidder $Y$ for first round and bidder $Z$ for the second round). This indicates that $X$'s presence directly influences the result. The $S_i^- = 18 + 10 = 28$ and the $S_i^* = 12 + 8 = 20$, so how much $X$ needs to pay in round 1 is $28 - 20 = 8$.
- Similarly, if $X$ wins in the first round, $Y$ is the pivotal bidder in round 2 for the same reason described above. How much $Y$ needs to pay is $(22 + 10) - (22 + 8) = 2$.
- In the final round, because there are no other bidders to compete, to maximize the social welfare, $Z$ needs to pay nothing and he could win this round and get the item without any compensation. So the expected social welfare is $\mathbb{WELFAREF} = 22 + 12 + 8 = 42$.

This example illustrates how the new payment rules operate and determines the amount each winner must pay in each round under a specific scenario. We will further state and prove additional advantages of this mechanism compared to the previous SPSB mechanism in a multi-round auction setting.

Table 2. Example of New Payment Rule

| Bidder | 1st Round | 2nd Round | 3rd Round |
|---|---|---|---|
| X | 22 | 15 | 10 |
| Y | 18 | 12 | 7 |
| Z | 15 | 10 | 8 |

## 2.3 Property Proof

As we cited before, the VCG mechanism needs to satisfy three properties in section 1.4. Now, we can prove the properties: *incentive compatibility, allocation-efficient, and individual rationality.*

**Proof.** Consider agent $i$ with valuation vector $\hat{v}_i$, and fix the reports $\mathbf{V}^{-i}$ of others. For incentive compatibility, consider a misreport $\hat{v'}_i$, with $\hat{v}_i \neq \hat{v'}_i$. Define allocation function $x(\mathbf{V})$, which outputs the allocation matrix $\mathbf{A}$ based on the valuation matrix, and $p_i(\mathbf{V})$ function, which outputs the price that bidder $i$ needs to pay. Let $\mathbf{A}^* = x(\hat{v}_i, \mathbf{V}^{-i})$, $\mathbf{A'} = x(\hat{v'}_i, \mathbf{V}^{-i})$, and $\mathbf{A}^{-i} = \arg\max_{\mathbf{A_a} \in \{\mathbf{0,1}\}^{(\mathrm{n-1})\times \mathrm{m}}} \mathbf{V^{-i}} \cdot \mathbf{A_a}$. We have the difference between utilities:

$$
\begin{aligned}
&\hat{v}_i \cdot \mathbf{A}^*_{:,i} - p_i(\hat{v}_i, \mathbf{V}^{-i}) - (\hat{v'}_i \cdot \mathbf{A'}_{:,i} - p_i(\hat{v'}_i, \mathbf{V}^{-i})) \\
&= \hat{v}_i \cdot \mathbf{A}^*_{:,i} + S_i^* - S_i^- - \left(\hat{v'}_i \cdot \mathbf{A'}_{:,i} + S_i^{'} - S_i^-\right) \\
&= \max_{\mathbf{A} \in \mathbb{A}^{n\times m}} \left(\hat{v}_i \cdot \mathbf{A}_{:,i} + S_i^*\right) - \left(\hat{v'}_i \cdot \mathbf{A'}_{:,i} + S_i^{'}\right) \geq 0.
\end{aligned} \tag{4}
$$

Here, inequity 4 follows from canceling out mutual terms and writing out the optimal choice rule of the VCG mechanism (used to select $\mathbf{A}^*$), and the result is weakly positive since $\mathbf{A'} \in \mathbb{A}^{n\times m}$. Allocation efficiency follows immediately from incentive compatibility, since the choice rule maximizes total reported value.

For individual rationality, and still with $\mathbf{A}^* = x(\hat{v}_i, \mathbf{V}^{-i})$, we have

$$
\begin{aligned}
\hat{v}_i \cdot \mathbf{A}^*_{:,i} - p_i(\hat{v}_i, \mathbf{V}^{-i}) - \hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} &= \hat{v}_i \cdot \mathbf{A}^*_{:,i} + S_i^* - S_i^- - \hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} \\
&= \max_{\mathbf{A} \in \mathbb{A}^{n\times m}} \left(\hat{v}_i \cdot \mathbf{A}_{:,i} + S_i^*\right) - \left(\hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} + S_i^-\right) \qquad (5) \\
&\geq \max_{\mathbf{A}^{-i} \in \mathbb{A}^{(n-1)\times m}} \left(\hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} + S_i^*\right) - \left(\hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} + S_i^-\right) \geq 0. \qquad (6)
\end{aligned}
$$

where in equation 5, we write out the choice rule of the VCG mechanism (used to select $\mathbf{A}^*$), the inequality 6 follows from $\mathbb{A}^{(n-1)\times m} \subseteq \mathbb{A}^{n\times m}$, and the result is weakly positive since $\mathbf{A}^{-i} \in \mathbb{A}^{(n-1)\times m}$.

For the auction scenario in this paper, we have $\hat{v}_i \cdot \mathbf{A}^{-i}_{:,i} = 0$ (zero value for no trade), and rationality provides $\hat{v}_i \cdot \mathbf{A}^*_{:,i} - p_i(\hat{v}_i, \mathbf{V}^{-i}) \geq 0$, so that the payment is no more than the value from trade.

## 3 Simulation

To verify the optimization for the new proposed mechanism, we focus on eliminating the regrets defined in section 1.5. The primitive motivation is due to the existence of regrets, the strategic bidder is motivated to bid untruthfully to gain more profits, which would impair the social welfare. So in order to make sure multi-rounds auction impels each bidder bids honestly, the new mechanism must satisfy the properties of VCG mechanism, mentioned at section 1.4.

### 3.1 Example for simulation

To better understand what should be done in the simulation, we are going to illustrate an example in this section. See the same value matrix as before, the table 3. Assuming bidder $X$ is strategic—meaning they are aware of the bidding strategies of others

Table 3. Example of Simulation

| Bidder | 1st Round | 2nd Round | 3rd Round |
|---|---|---|---|
| X | 22 | 15 | 10 |
| Y | 18 | 12 | 7 |
| Z | 15 | 10 | 8 |

(represented by rows $Y$ and $Z$ in this example)—they may choose to bid untruthfully. Specifically, $X$ might intentionally lose in early rounds and aim to win in a particular round that maximizes their utility. The result we need to observe is the strategic bidder's regret, through the VCG-combined mechanism and the sequential second price(SSP) mechanism.

**SSP Mechanism** Each round follows a second-price auction format, meaning the highest bidder wins but pays the amount of the second-highest bid among the left bidders.

(1) *$X$ wins on round 1.* $X$ bids $[10^{10}, 0, 0]$, $X$ needs to pay 18, utility $U = 22 - 18 = 4$.

(2) *$X$ wins on round 2.* $X$ bids $[0, 10^{10}, 0]$, $X$ needs to pay 10, because bidder $Y$ wins at the first round. So, utility $U = 15 - 10 = 5$.

(3) *$X$ wins on round 3.* $X$ bids $[0, 0, 10^{10}]$, $X$ needs to pay 0, because bidder $Y$ and $Z$ all win at previous rounds. So, utility $U = 10 - 0 = 10$.

If $X$ bids honestly, his utility would be 4, winning at round 1. So the regrets for $X$ using SSP in this auction scenario would be $Regret_X = 5 - 4 = 1$.

**VCG-combined Mechanism**

(1) *$X$ wins on round 1.* $X$ bids $[10^{10}, 0, 0]$, $X$ needs to pay $(18 + 10) - (12 + 8) = 8$, utility $U = 22 - 8 = 14$.

(2) *$X$ wins on round 2.* $X$ bids $[0, 10^{10}, 0]$, $X$ needs to pay $(18 + 10) - (18 + 8) = 8$, utility $U = 15 - 2 = 13$.

(3) *$X$ wins on round 3.* $X$ bids $[0, 0, 10^{10}]$, $X$ needs to pay $(18 + 10) - (18 + 10) = 0$, utility $U = 10 - 0 = 10$.

If $X$ bids honestly, his utility would be 14, winning at round 1. So the regrets for $X$ using SSP in this auction scenario would be $Regret_X = 14 - 14 = 0$.

Given this example, we can find that the VCG-based mechanism effectively eliminates regrets, no matter at which rounds the strategic bidder would win. We run a python simulation to verify this result.

### 3.2 Code Simulation

In order to verify the approach for VCG-combined mechanism, we design a simulation to compare the regrets of the new mechanism and the SSP mechanism. In general, we consider an auction scenario with $n = 20$ bidders and $m = 20$ rounds. The valuation matrix $V$ is generated such that each entry $v_{i,j} \sim Uniform(0, 1)$. In each mechanism simulation, we focus on each bidder and let him be the strategic bidder, finding his highest utility by manipulating his bidding strategy and calculating his regrets in this auction, while others are honest. Finally, we will compare the average regrets of VCG-based mechanism and SSP mechanism, over 10 times simulation, in the line charts, and sort the regrets in the plot in descending order, to help understand the optimization of regrets. For the full simulation code, visit the GitHub repository: resource

### 3.3 Result Analysis

We present both the plots and data as averages over 10 simulation runs. The average regrets are sorted in descending order in the following plots. For more data output, please check Appendices A 5 and B 5.

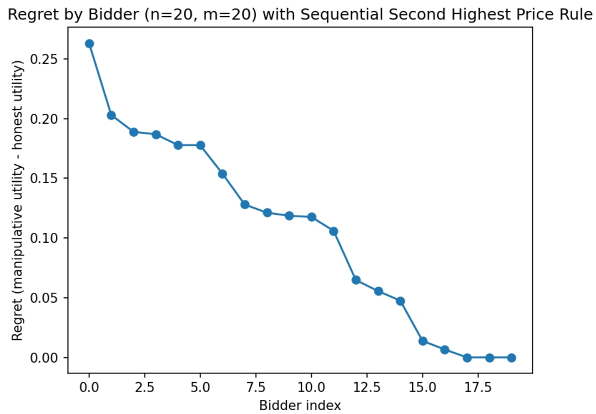


Fig. 1. regrets for SSP

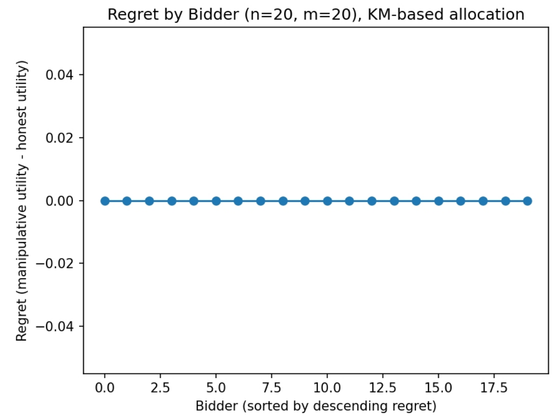


Fig. 2. regrets for VCG-combined

Based on the two line charts, we can find that our new VCG-combined mechanism significantly removes the existence of regrets, which means even a bidder knows other bidders' strategies ahead; he cannot earn any profit by bidding untruthfully. In other words, it satisfies the incentive compatibility, because every bidder could earn his best utility by being honest.

## 4 Conclusion

In this paper, we proposed an auction mechanism tailored for a multi-round auction scenario, employing principles from the VCG mechanism. The proposed mechanism, on the one hand, maximizes social welfare by allocating items to bidders such that the total utility for bidders and sellers is optimized. On the other hand, the payment rule is based on the "externality" each winning bidder imposes on other participants, ensuring incentive compatibility by preventing bidders from increasing their profit through dishonest bidding.

We developed a simulation in which bidders act strategically with complete information about others' valuations prior to the auction. The simulation measured the difference between bidders' best achievable utilities through strategic manipulation and the utilities gained through honest bidding, defined as regret. Results indicated that under a fixed bidding strategy, the proposed mechanism exhibits zero regret, demonstrating incentive compatibility.

Moreover, based on the characteristics of the KM *Hungarian* algorithm and the predetermined value matrix, the allocation rule consistently achieves maximum social welfare. Nevertheless, the proposed auction mechanism is defined within strict constraints, including a fixed valuation matrix prior to the auction, descending valuations for each bidder across rounds, and a rule limiting each bidder to at most one round of winning.

## 5 Future Work

There are also numerous scenarios warrant further exploration regarding mechanism design to address the fundamental questions introduced earlier: determining the auction winner and calculating the appropriate payments. Specifically, it is important to investigate whether existing mechanisms remain effective under dynamic conditions, such as multi-round auctions involving dynamic bidder behavior. Questions arise about how to achieve equilibrium and optimally balance social welfare and individual utilities when the bidding matrix is unknown at the outset, or when participants are allowed to freely join and exit the auction.

Moreover, in increasingly complex scenarios where bidder participation dynamically fluctuates across auction rounds, the VCG mechanism may no longer yield optimal results. Consequently, further research is essential to develop alternative mechanism designs and innovative protocols that effectively address these challenges.

Additionally, there is considerable scope to enhance the efficiency of the algorithm itself. For instance, applying *linear programming* techniques, rather than relying on brute-force enumeration methods, could better accommodate more sophisticated constraints.

Such advancements would enable customization beyond mere value maximization, incorporating more nuanced auction rules and diverse objectives.

## References


[1] Vicki M Coppinger, Vernon L Smith, and Jon A Titus. 1980. Incentives and behavior in English, Dutch and sealed-bid auctions. *Economic inquiry* 18, 1 (1980), 1–22.
[2] Matheus VX Ferreira and S Matthew Weinberg. 2020. Credible, truthful, and two-round (optimal) auctions via cryptographic commitments. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 683–712.
[3] Amos Fiat, Kira Goldner, Anna R Karlin, and Elias Koutsoupias. 2016. The fedex problem. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. 21–22.
[4] Amin Nezarat and GH Dastghaibifard. 2015. Efficient nash equilibrium resource allocation based on game theory mechanism in cloud computing by using auction. *PloS one* 10, 10 (2015), e0138424.
[5] Noam Nisan and Amir Ronen. 2007. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research* 29 (2007), 19–47.
[6] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press. https://books.google.com/books?id=YCu2alSw0w8C
[7] Mallesh M. Pai and Max Resnick. 2024. *Dynamic Transaction Fee Mechanism Design*. Technical Report. Special Mechanisms Group. Accessed: 2025-04-19.
[8] Aldo Rustichini. 1999. Minimizing regret: The general case. *Games and Economic Behavior* 29, 1-2 (1999), 224–243.
[9] Wikipedia contributors. 2024. Linear programming — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Linear_programming. Accessed: 2025-04-19.
[10] Yu Ye, Xiao Ke, and Zhiyong Yu. 2021. A Cost Matrix Optimization Method Based on Spatial Constraints under Hungarian Algorithm. In *Proceedings of the 6th International Conference on Robotics and Artificial Intelligence* (Singapore, Singapore) *(ICRAI '20)*. Association for Computing Machinery, New York, NY, USA, 134–139. doi:10.1145/3449301.3449324


## A Full Simulation Output for SSP

### A.1 Best Deviation Utility per Bidder(Average over 10 simulation)

*Sorted Descending*

```
Bidder 18 best deviation utility: 0.269709
Bidder 14 best deviation utility: 0.247023
Bidder 5 best deviation utility: 0.212251
Bidder 12 best deviation utility: 0.203577
Bidder 16 best deviation utility: 0.197491
Bidder 13 best deviation utility: 0.192834
Bidder 6 best deviation utility: 0.190938
Bidder 10 best deviation utility: 0.167587
Bidder 7 best deviation utility: 0.176675
Bidder 0 best deviation utility: 0.140770
Bidder 15 best deviation utility: 0.121511
Bidder 19 best deviation utility: 0.119559
Bidder 17 best deviation utility: 0.106197
Bidder 1 best deviation utility: 0.104286
Bidder 4 best deviation utility: 0.048915
Bidder 2 best deviation utility: 0.046692
Bidder 8 best deviation utility: 0.056553
Bidder 11 best deviation utility: 0.024210
Bidder 9 best deviation utility: 0.011442
Bidder 3 best deviation utility: 0.008193
```

### A.2 Regrets by Bidder (Sorted)

```
Bidder 18: Regret = 0.263190
Bidder 12: Regret = 0.202952
```

```
Bidder 5: Regret = 0.188898
Bidder 14: Regret = 0.186863
Bidder 16: Regret = 0.177810
Bidder 6: Regret = 0.177606
Bidder 13: Regret = 0.153936
Bidder 10: Regret = 0.128006
Bidder 7: Regret = 0.121210
Bidder 0: Regret = 0.118617
Bidder 15: Regret = 0.117602
Bidder 17: Regret = 0.106031
Bidder 19: Regret = 0.064841
Bidder 1: Regret = 0.055536
Bidder 4: Regret = 0.047558
Bidder 2: Regret = 0.013860
Bidder 11: Regret = 0.006625
Bidder 8: Regret = 0.000000
Bidder 3: Regret = 0.000000
Bidder 9: Regret = 0.000000
```

### A.3 Payments

```
[0.57124078 0.87565699 0.09877131 0.         0.98212642 0.27666831
 0.75943183 0.32669583 0.03144834 0.01985273 0.51651889 0.9503352
 0.80732757 0.46039659 0.69006659 0.92891748 0.41457021 0.99781164
 0.64481174 0.18392491]
```

### A.4 Honest Utilities

```
[0.02215358 0.04874964 0.0328318  0.00819284 0.00135631 0.02335281
 0.01333185 0.05546484 0.05655251 0.01144245 0.03958099 0.01758557
 0.00062557 0.0388983  0.06016008 0.00390889 0.01968138 0.00016646
 0.00651862 0.05471828]
```

### A.5 Regrets

```
[0.11861677 0.05553628 0.01385981 0.         0.04755847 0.18889777
 0.17760578 0.12121042 0.         0.         0.12800558 0.0066247
 0.2029515  0.15393584 0.18686301 0.11760216 0.17781009 0.10603071
 0.26318993 0.0648407 ]
```

### A.6 Value Matrix

```
[[0.99625751 0.94989236 0.88155512 0.80973563 0.74624568 0.70552338
  0.63449884 0.61525675 0.59925995 0.59339435 0.56648216 0.55560709
  0.48491263 0.45625778 0.37040375 0.27801747 0.23954166 0.10950168
  0.07707959 0.06662472]
 [0.98070977 0.97750551 0.94771095 0.92891748 0.92440663 0.91223906
  0.75117244 0.66070583 0.64696394 0.61846642 0.56415635 0.49027708
  0.48606588 0.3078619  0.27812201 0.23819504 0.14323654 0.13100205
```

```
  0.10882236 0.03405597]
 [0.90586505 0.89861333 0.85843961 0.79108018 0.77756835 0.65342181
  0.61082612 0.5781895  0.56065883 0.52056586 0.39504125 0.3881648
  0.32753261 0.32669583 0.27666831 0.16047732 0.13160311 0.07813995
  0.05010965 0.00160818]
 [0.95738301 0.87894831 0.87222252 0.8204265  0.80788754 0.7729297
  0.75686839 0.53022146 0.37896056 0.33661401 0.31386148 0.29972999
  0.24222969 0.22521451 0.20367724 0.17314113 0.06472189 0.0312217
  0.01985273 0.00819284]
 [0.98621232 0.98348274 0.86924709 0.85454972 0.82346874 0.77196369
  0.6788341  0.67048872 0.5748411  0.57250855 0.44731075 0.35366807
  0.31795598 0.28922243 0.2016917  0.11509501 0.09519128 0.07375071
  0.06876751 0.02462163]
 [0.9775672  0.9560913  0.94007716 0.92021264 0.87565699 0.75139736
  0.6593884  0.61992499 0.61256987 0.5647393  0.49475687 0.40038941
  0.33661252 0.30496983 0.30002112 0.27219711 0.27049668 0.21835868
  0.21761966 0.21225058]
 [0.98582324 0.94335327 0.91974511 0.91562943 0.8165694  0.80732757
  0.77276369 0.7389743  0.70648924 0.65699091 0.62139335 0.56278122
  0.55939883 0.39300324 0.35711088 0.33643639 0.28970895 0.22102114
  0.09208379 0.0653765 ]
 [0.90631178 0.90125111 0.86085175 0.81010609 0.77730728 0.754318
  0.62558368 0.61485659 0.60056514 0.501822   0.47604203 0.46039659
  0.41457021 0.38216067 0.37051659 0.33510626 0.27544657 0.19747852
  0.19519298 0.08816592]
 [0.90683223 0.90468886 0.77375939 0.76637047 0.75242369 0.67940739
  0.66747045 0.64442848 0.56687581 0.33011623 0.32788971 0.32547716
  0.31177467 0.23944776 0.1883132  0.18392491 0.09877131 0.08800085
  0.04350738 0.00780503]
 [0.88529616 0.88217124 0.73773034 0.73242339 0.71597028 0.6885613
  0.62120411 0.4320053  0.41831318 0.36176475 0.34031419 0.33470672
  0.31738723 0.28964034 0.26068636 0.16669492 0.06032677 0.03144834
  0.03129518 0.00890253]
 [0.82909914 0.77251871 0.75035811 0.71958562 0.70757068 0.6709775
  0.66973591 0.62168119 0.61282943 0.57124078 0.55609988 0.5205115
  0.43457405 0.39088286 0.28486773 0.27589571 0.20331477 0.19903492
  0.08887141 0.07026296]
 [0.98057591 0.97204406 0.96792077 0.82777836 0.82722713 0.76804426
  0.66285724 0.62963811 0.47609911 0.44863423 0.31568866 0.22701383
  0.180809   0.16401033 0.15299511 0.06810693 0.06613879 0.05565861
  0.0407653  0.01445035]
 [0.99781164 0.90635595 0.90018167 0.85806686 0.85344996 0.80795314
  0.72183918 0.69596261 0.68488371 0.64036791 0.61982638 0.55484046
  0.53752802 0.5302729  0.4001244  0.12852867 0.09636972 0.07113978
  0.06404097 0.0385211 ]
 [0.98758662 0.98212642 0.861643   0.77533941 0.7676098  0.74507194
  0.7125539  0.69006659 0.64481174 0.54808804 0.51651889 0.4992949
  0.42683874 0.41493748 0.36332393 0.26689833 0.25689262 0.22428248
  0.18607113 0.13276477]
 [0.96156198 0.92620794 0.89989971 0.81520773 0.79224174 0.78837719
  0.75943183 0.75022667 0.59506849 0.56296371 0.53951042 0.53560435
  0.5071945  0.49777268 0.43493043 0.4137358  0.30426241 0.27847144
  0.09649132 0.02686908]
 [0.99752009 0.98175244 0.9503352  0.93282637 0.89301544 0.86741376
  0.62334317 0.6155918  0.61384588 0.59747566 0.54431612 0.54239112
  0.50853612 0.47801158 0.30441596 0.28021913 0.22028236 0.13837485
  0.0277208  0.01650316]
 [0.97581751 0.79295565 0.71722721 0.71383987 0.6521049  0.5388188
  0.52902534 0.48947825 0.48768176 0.47134607 0.44264017 0.43686683
  0.4342516  0.3496581  0.30633363 0.26180631 0.23455295 0.22040618
  0.2173442  0.03576652]
 [0.9979781  0.89184963 0.89120946 0.77943537 0.76480699 0.70694403
  0.6884658  0.61092416 0.54208319 0.48140962 0.32225506 0.27340393
  0.26073024 0.24391297 0.23028398 0.2236255  0.19994935 0.13764551
  0.10489548 0.08706315]
```

```
[0.90324906 0.84383532 0.77274223 0.76042194 0.71914365 0.69372185
 0.68579481 0.67545732 0.65133036 0.64642899 0.56333653 0.53310802
 0.49745972 0.44853134 0.39252396 0.37863881 0.36847986 0.23238191
 0.14038681 0.06982095]
[0.95449757 0.92047668 0.86488558 0.85804952 0.75069764 0.61629962
 0.60133134 0.58405211 0.58217243 0.43129898 0.41006359 0.35671552
 0.32674956 0.30772992 0.2717562  0.23864318 0.22000499 0.15100731
 0.10626767 0.06443152]]
```

## B Full Simulation Output for VCG-Combined Mechanism

### B.1 Best Deviation Utility per Bidder(Average over 10 simulations)

*Sorted Descending*

```
Bidder 7 best deviation utility = 0.456804
Bidder 18 best deviation utility = 0.279492
Bidder 14 best deviation utility = 0.274894
Bidder 12 best deviation utility = 0.294274
Bidder 5 best deviation utility = 0.269423
Bidder 19 best deviation utility = 0.266434
Bidder 8 best deviation utility = 0.263069
Bidder 11 best deviation utility = 0.254799
Bidder 6 best deviation utility = 0.251720
Bidder 2 best deviation utility = 0.244892
Bidder 9 best deviation utility = 0.238888
Bidder 16 best deviation utility = 0.194732
Bidder 17 best deviation utility = 0.193343
Bidder 0 best deviation utility = 0.191480
Bidder 3 best deviation utility = 0.180122
Bidder 4 best deviation utility = 0.140122
Bidder 10 best deviation utility = 0.132676
Bidder 15 best deviation utility = 0.126625
Bidder 13 best deviation utility = 0.124202
Bidder 1 best deviation utility = 0.050014
```

### B.2 Regrets by Bidder (All Zero)

```
Bidder 0: Regret = 0.000000
Bidder 1: Regret = 0.000000
Bidder 2: Regret = 0.000000
Bidder 3: Regret = 0.000000
Bidder 4: Regret = 0.000000
Bidder 5: Regret = 0.000000
Bidder 6: Regret = 0.000000
Bidder 7: Regret = 0.000000
Bidder 8: Regret = 0.000000
Bidder 9: Regret = 0.000000
Bidder 10: Regret = 0.000000
Bidder 11: Regret = 0.000000
```

```
Bidder 12: Regret = 0.000000
Bidder 13: Regret = 0.000000
Bidder 14: Regret = 0.000000
Bidder 15: Regret = 0.000000
Bidder 16: Regret = 0.000000
Bidder 17: Regret = 0.000000
Bidder 18: Regret = 0.000000
Bidder 19: Regret = 0.000000
```

### B.3 Payments

```
[0.74635034 0.25519474 0.5070482  0.34857234 0.40275161 0.09046344
 0.19731336 0.46617999 0.28500725 0.42957883 0.85273885 0.
 0.578285   0.03554544 0.69520216 0.13400512 0.00290181 0.62838741
 0.17191741 0.66007964]
```

### B.4 Honest Utilities

```
[0.19147958 0.05001397 0.24489164 0.180122   0.14012233 0.26942287
 0.25171952 0.45680359 0.2630693  0.23888785 0.1326754  0.25479945
 0.29427379 0.12420241 0.27489354 0.12662472 0.1947322  0.19334324
 0.27949212 0.26643372]
```

### B.5 Regrets

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

### B.6 Value Matrix

```
[[9.62688592e-01 9.37829913e-01 7.75384979e-01 7.36435895e-01
  5.93023847e-01 5.56536504e-01 5.15451479e-01 3.76338162e-01
  3.32930363e-01 3.21861482e-01 2.73620047e-01 1.84174876e-01
  1.58096532e-01 1.49551396e-01 1.35416814e-01 1.02823783e-01
  9.64921657e-02 9.18293473e-02 7.29718558e-02 6.15713673e-02]
 [8.41315821e-01 7.77390036e-01 7.01025960e-01 6.59700183e-01
  5.96125790e-01 4.60964478e-01 4.40584723e-01 4.35192895e-01
  4.05526922e-01 3.53391020e-01 3.48955329e-01 3.35021221e-01
  3.05208711e-01 1.87155360e-01 1.22003631e-01 1.17748149e-01
  8.85145198e-02 2.87031796e-02 2.42686931e-02 1.69348225e-02]
 [9.92612806e-01 9.91241976e-01 8.61383680e-01 8.43255268e-01
  8.11423694e-01 8.01113333e-01 7.51939834e-01 6.37229231e-01
  6.24217391e-01 6.09410217e-01 5.37421343e-01 3.93011530e-01
  3.75984131e-01 3.11320475e-01 1.98823901e-01 1.36064211e-01
  1.16536279e-01 1.05893477e-01 2.61382821e-02 1.92416252e-02]
 [9.99497205e-01 9.14056857e-01 8.54095964e-01 8.09029255e-01
  6.93471799e-01 6.56468161e-01 5.77987754e-01 5.59660724e-01
  5.56381798e-01 5.54001523e-01 5.28694341e-01 4.06956907e-01
  3.84065151e-01 3.53832295e-01 3.14768915e-01 2.58373371e-01
  7.35270437e-02 5.86339574e-02 3.17069779e-02 3.57535308e-03]
 [9.91888081e-01 8.35447206e-01 7.99115007e-01 6.99847925e-01
  6.34779286e-01 6.21024848e-01 6.05402631e-01 5.93557088e-01
  5.50417194e-01 5.42873937e-01 4.85985160e-01 3.08665954e-01
  2.97225051e-01 2.16208757e-01 2.13058473e-01 1.35040446e-01
  1.33623570e-01 8.09524341e-02 1.95993112e-02 7.18957568e-03]
 [9.46830297e-01 9.43197187e-01 9.11287437e-01 8.47783210e-01
```

```
 8.34040711e-01 7.88608981e-01 7.70714985e-01 7.18236670e-01
 6.99001697e-01 5.98192064e-01 5.03355815e-01 4.92103820e-01
 4.86171888e-01 4.53789195e-01 4.39435575e-01 3.64239563e-01
 3.59886309e-01 3.01235214e-01 2.43909996e-01 6.73795595e-02]
[9.33397968e-01 9.33080158e-01 9.23407474e-01 8.93821093e-01
 8.53559101e-01 7.61325557e-01 7.58767713e-01 6.91290112e-01
 5.84282051e-01 5.37886444e-01 5.22594282e-01 5.06594288e-01
 4.56162148e-01 4.49032878e-01 4.12990727e-01 3.03975502e-01
 1.42492405e-01 1.19248035e-01 1.12745117e-01 5.71096930e-02]
[9.93042978e-01 9.80203316e-01 9.75590231e-01 9.71450528e-01
 9.70651362e-01 9.66629123e-01 9.54889599e-01 9.22983580e-01
 8.08909661e-01 6.79323998e-01 6.17132849e-01 4.13119064e-01
 3.41497969e-01 1.98217761e-01 1.73195213e-01 1.50434262e-01
 9.44755994e-02 7.39505322e-02 4.03374156e-02 2.52096376e-02]
[9.51031909e-01 8.21704696e-01 8.08939779e-01 6.97403328e-01
 6.66365678e-01 6.47470072e-01 6.22908544e-01 5.66604379e-01
 5.63886176e-01 5.59962631e-01 5.50092057e-01 5.48076550e-01
 4.34248101e-01 3.73695203e-01 3.65794041e-01 2.81072867e-01
 1.61355326e-01 9.61571338e-02 7.76640254e-02 8.28347329e-03]
[9.82156330e-01 9.80179340e-01 9.34090014e-01 7.70234029e-01
 7.69670986e-01 7.61920678e-01 7.23479653e-01 6.75798656e-01
 6.68466682e-01 6.20316519e-01 5.87460196e-01 2.89290197e-01
 2.34585397e-01 2.11949705e-01 9.97031283e-02 8.33291545e-02
 7.23367748e-02 6.14729785e-02 4.24608904e-02 3.78892751e-03]
[9.85414250e-01 8.51346943e-01 7.37606792e-01 6.75336930e-01
 5.48600178e-01 4.27346103e-01 4.14522955e-01 3.76444395e-01
 2.47521678e-01 2.25190969e-01 2.13065641e-01 1.96579190e-01
 1.87580509e-01 1.87138507e-01 1.74860680e-01 1.66223250e-01
 1.48156964e-01 1.37179527e-01 1.07492873e-01 1.01531094e-01]
[9.96356363e-01 8.68773784e-01 8.29333592e-01 7.45632658e-01
 7.36359389e-01 7.12922451e-01 6.97311373e-01 6.09756829e-01
 5.57103354e-01 5.24355516e-01 4.60344753e-01 4.51105667e-01
 4.44603989e-01 4.28400526e-01 3.51443840e-01 3.27588828e-01
 3.25564764e-01 2.74453634e-01 2.57701262e-01 2.54799449e-01]
[9.89139529e-01 9.86674457e-01 9.65919584e-01 8.84684735e-01
 8.79488049e-01 8.72558789e-01 7.72205432e-01 7.26646901e-01
 7.11030904e-01 6.01088322e-01 5.88293273e-01 5.15690589e-01
 4.92301246e-01 4.91587151e-01 4.13917163e-01 1.97470983e-01
 1.07865198e-01 9.84442605e-02 4.12986313e-02 1.35779019e-02]
[9.76941265e-01 8.22955165e-01 8.15618293e-01 7.84282052e-01
 7.22493563e-01 7.02487411e-01 5.96824713e-01 5.06059176e-01
 3.81456813e-01 3.53673492e-01 3.01278880e-01 2.68435103e-01
 2.61684085e-01 2.61256987e-01 2.59969098e-01 2.58207532e-01
 1.73095441e-01 1.59747852e-01 8.07193878e-02 5.70579285e-02]
[9.96378024e-01 9.83190880e-01 9.70095699e-01 8.65871292e-01
 7.53869115e-01 6.87434664e-01 5.82884182e-01 5.72207669e-01
 5.17337485e-01 4.49381112e-01 3.22754263e-01 2.82831386e-01
 2.79686438e-01 2.72222402e-01 2.53210724e-01 9.56983383e-02
 9.44134040e-02 5.94612533e-02 4.26734994e-02 6.15345749e-05]
[8.43207973e-01 8.26791932e-01 8.15812666e-01 7.22114269e-01
 6.47854218e-01 6.24730428e-01 6.00928331e-01 5.54370193e-01
 5.38929745e-01 5.29376324e-01 4.25770187e-01 4.10703985e-01
 3.81819460e-01 3.13009120e-01 2.98542124e-01 2.60629837e-01
 1.46003904e-01 1.32934042e-01 1.08405806e-01 3.27976857e-02]
[9.65848930e-01 9.22955998e-01 8.71778284e-01 8.42533382e-01
 8.04345594e-01 7.19736703e-01 6.70685050e-01 6.50788543e-01
 6.03762260e-01 5.45523403e-01 5.18516383e-01 4.16812519e-01
 3.97627136e-01 3.63520748e-01 3.32583638e-01 2.95245951e-01
 2.85195640e-01 2.30277641e-01 1.97634014e-01 1.80417580e-01]
[9.94262867e-01 8.99457491e-01 8.33640830e-01 8.26289126e-01
 8.21730654e-01 7.39505624e-01 6.67801133e-01 6.59523232e-01
 6.15852130e-01 5.31356728e-01 4.60455000e-01 4.38223337e-01
 4.34705930e-01 2.81263897e-01 2.45968648e-01 2.07802961e-01
 1.96079254e-01 4.81583764e-02 4.61426590e-02 3.75474768e-02]
[9.21468833e-01 9.00354392e-01 8.66577313e-01 8.65652068e-01
```

```
 8.50475218e-01 8.14076747e-01 6.93220915e-01 6.88316093e-01
 6.62441291e-01 6.13090274e-01 5.64701427e-01 5.53389864e-01
 5.10120201e-01 4.71050967e-01 4.51409528e-01 3.78330003e-01
 3.52919140e-01 1.03120600e-01 8.25864958e-02 5.64904088e-02]
[9.99640955e-01 9.95099479e-01 9.36475804e-01 9.26513361e-01
 8.94821134e-01 7.84523100e-01 7.44338496e-01 7.00213937e-01
 5.08545172e-01 5.03435904e-01 4.69941090e-01 4.44928944e-01
 2.89926760e-01 2.75743168e-01 1.63895834e-01 9.87764868e-02
 4.46903651e-02 2.64255039e-02 2.37117839e-02 1.54727403e-02]]
```