RWTH Aachen University
Chair of Management Science
04.07.2024
Communicated by Prof. Dr. rer. nat. Britta Peis

Tailoring a Matroid-Constrained Ascending Vickrey Auction to Representable Matroids

Bachelor's Thesis
in partial fulfillment of the requirements
for the degree of
Bachelor of Science
in Computer Science

submitted by
Luca Kempkes (422262)

Supervisor:	Niklas Rieken, M. Sc.
First Examiner:	Prof. Dr. rer. nat. Britta Peis
Second Examiner:	Prof. Dr. rer. nat. Christina Büsing

Abstract

In their paper titled "An Ascending Vickrey Auction for Selling Bases of a Matroid", Bikhchandani et al. propose an auction with numerous desirable properties [1]. However, their auction relies on the concept of a matroid oracle. This oracle is a theoretical construct that abstracts from the underlying type of matroid and the specific matroid operations employed by the auction. For practical application, the auction should operate on real matroid descriptions directly. Representable matroids are types of matroids that can be characterized by compact descriptions such as matrices or graphs, making them suitable for modeling real-world settings. This thesis demonstrates how the auction by Bikhchandani et al. can efficiently operate on the compact descriptions of various classes of representable matroids directly, without relying on an oracle. We achieve this by showing how the matroid operations of finding cocircuits, deletion, and contraction work on the compact descriptions of linear matroids, graphic matroids, planar matroids, gammoids, partition matroids, and uniform matroids. We analyze the worst-case time complexities of our operations and implement the auction along with our adapted operations in `Python`. Based on this implementation, we conduct a running time comparison to evaluate the real-world performance of the auction across the different classes of representable matroids.

Contents

1 Introduction **1**

1.1 Related Work . . . 3
1.2 Contribution . . . 5
1.3 Overview . . . 5

2 Background **7**

2.1 Auction Theory . . . 7
2.1.1 Fundamentals . . . 7
2.1.2 VCG Auction . . . 9
2.2 Matroid Theory . . . 11
2.2.1 Fundamentals . . . 11
2.2.2 Representable Matroids . . . 13
2.2.3 Descriptions . . . 17

3 The Auction by Bikhchandani et al. **19**

3.1 Economic Model . . . 19
3.2 The Auction . . . 20
3.2.1 Unit-Step Auction . . . 21
3.2.2 Long-Step Auction . . . 25
3.3 Properties . . . 25
3.3.1 Welfare Maximization . . . 25
3.3.2 Truthful Ex-Post Equilibrium . . . 26
3.3.3 Polynomial Running Time . . . 29

4 Tailoring the Auction to Representable Matroids **31**

4.1 Finding Cocircuits . . . 31
4.1.1 Dual Approach . . . 31
4.1.2 Primal Approach . . . 33
4.2 Adaptations . . . 34
4.2.1 Uniform Matroid . . . 35

4.2.2 Partition Matroid 38
4.2.3 Planar Matroid 42
4.2.4 Graphic Matroid 48
4.2.5 Gammoid 54
4.2.6 Linear Matroid 68
4.3 Time Complexities 74

5 Running Time Comparison **75**
5.1 Environment 75
5.2 Methodology 75
5.3 Limitations 78
5.4 Results 79
5.5 Discussion 86

6 Conclusion **89**

Bibliography **91**

1 Introduction

Consider an auction for selling bundles of indivisible, unique goods to buyers. Specifically, we focus on the case where the auctioneer is constrained to sell a set of elements that is independent in some underlying matroid.

A matroid, as detailed in Section 2.2, is a structure that generalizes the idea of linear independence in vector spaces to more abstract settings. In the context of this auction, a matroid involves a set of elements (the goods) and a collection of subsets of these elements (potential combinations of goods), called independent sets, that satisfy certain axioms. The auction can only allocate combinations of goods that form an independent set in this matroid.

In this auction, the bidders have non-negative valuations that are additively separable in money and goods. Each item won provides the buyer with utility equal to their valuation for the acquired item. The primary goal of this auction is to maximize social welfare, which refers to allocating the goods in a way that maximizes total utility across all bidders.

Achieving maximum social welfare requires accurate information about the valuations, which are usually private. Therefore, this information must be inferred from the submitted bids. In this context, a bid is the value a bidder offers to the auctioneer for an item or a combination of items, signaling their willingness to pay. Thus, bidders should provide bids that reflect their true valuation of the goods. Consequently, the auction should incentivize truthful bidding. Furthermore, the auction should be computationally efficient, operating within polynomial time.

Overall, we consider an auction that maximizes welfare under truthful bidding, incentivizes truthful bidding, and operates within polynomial time under matroid constraints. The sealed-bid VCG Auction, introduced in Section 2.1.2, possesses these desired properties [2, 3, 4]. However, there are reasons to prefer an ascending

auction format over a sealed-bid auction.

In this context, an ascending auction is an auction where the price is openly announced and where bidders report their demand at this price. The price is raised as long as there is excess demand from the bidders. Otherwise, the auction concludes. Ascending auctions offer several benefits over sealed-bid auctions: they are more transparent, reveal less private information about the bidder's valuations, and generally incur lower communication and computation costs. They may also be advantageous when bidder valuations are slightly interdependent [5, 6].

Bikhchandani et al. propose an ascending auction that matches the described setting and possesses the desired properties [1]. However, this auction is conceptualized through the theoretical construct of a matroid oracle. As detailed in Section 2.2.3, a matroid oracle functions as a black box by responding to queries, such as determining whether a given set is independent within the matroid. Although all matroids can be described without an oracle, these general descriptions may require exponential space in the number of items. Consequently, Bikhchandani et al. rely on a matroid oracle to avoid this issue. To efficiently run the auction in real-world scenarios, however, it should run on compact matroid descriptions directly without depending on an oracle.

Compact matroid descriptions, as explained in Section 2.2.3, are used to describe particular classes of matroids known as representable matroids. These descriptions provide a concise way to characterize the independent sets of representable matroids. By restricting the auction to representable matroids, we eliminate the problem of superpolynomial description sizes. Compact descriptions can take various forms, including matrices, graphs (represented by structures like adjacency lists, incidence matrices, or sets of edges), or sequences of sets and numbers. However, performing matroid operations on these descriptions can be complex.

This thesis tailors the auction of Bikhchandani et al. to representable matroid classes, allowing it to operate directly on their compact descriptions.

1.1 Related Work

This thesis builds upon existing research on ascending auctions. In this section, we review the work most closely related to our auction setting. Additionally, we consider research that deals with matroid operations on compact matroid descriptions.

The valuation functions in our setting satisfy the gross substitutes condition. Fundamental research on ascending auctions with gross substitute valuations was conducted by Kelso and Crawford [7] and Gul and Stacchetti [8]. While these auctions maximize social welfare when bidders behave truthfully, they do not incentivize truthful bidding. Furthermore, these auctions are always market-clearing, meaning they always allocate the entire set of items.

Note that the auction in our setting is market-clearing when constrained by the matroid where the set of all items is independent, also known as the free matroid. Given that the valuations in our setting are non-negative, we can assume w.l.o.g. that the welfare-maximizing set is an inclusion-wise maximal independent set, so in this case the set of all items. Since the auction in our setting is welfare-maximizing, it is market-clearing when constrained by the free matroid. Additionally, our auction applies to scenarios where only a subset of the items should be allocated, making it more general than pure market-clearing auctions in this respect.

Given that the valuations in our setting satisfy the gross substitutes condition, the ascending auctions proposed by Parkes and Ungar [9], Ausubel and Milgrom [10], de Vries et al. [11], and Mishra and Parkes [12] apply. These auctions maximize social welfare and incentivize truthful bidding, provided appropriate consistency checks are performed. However, these auctions require an exponential number of steps in the number of items.

Ausubel analyzes ascending auctions in environments with multiple items and bidders, considering both multi-unit [5] and single-unit [13] settings. While these auctions achieve efficient outcomes in equilibrium, they again are designed to be market-clearing and therefore differ from our matroid-constrained auction.

Bikhchandani et al. [1] propose an auction where the set of feasible allocations is constrained by a matroid. This auction can be applied to our setting and has

the desired properties. However, the auction is conceptualized through a matroid oracle. For practical application, it should operate on compact matroid descriptions directly, without relying on an oracle.

More recently, Peis and Rieken [14] have simplified the main proofs in the paper by Bikhchandani et al., demonstrating that the auction is welfare-maximizing and charges Vickrey prices. The fact that the auction charges Vickrey prices is fundamental to showing that truthful bidding results in an ex-post equilibrium, thereby for showing that the auction incentivizes truthful bidding.

Although the auction by Bikhchandani et al. can be extended to Polymatroids via Helgason's reduction [15], this reduction is pseudo-polynomial in the highest quantity of any item. In his PhD thesis, which extends the work by Bikhchandani et al., Raach [16] proposes an auction designed to sell a basis of a polymatroid while achieving polynomial running time.

Next, we review research that deals with matroid operations on compact matroid descriptions. Specifically, the auction by Bikhchandani et al. involves the matroid operations of finding cocircuits, deletion, and contraction. While there is research on applying these operations to compact matroid descriptions, there is a noticeable gap, particularly regarding the key operation of finding cocircuits.

The textbook by Oxley [17] serves as the main foundation for matroid theory in this thesis. It demonstrates the application of contraction and deletion on the descriptions of some classes of representable matroids, specifically uniform, graphic, and linear matroids. However, his work is not tailored to the auction setting and does not directly address the operation of finding cocircuits on compact matroid descriptions.

Stallmann [18] offers a concise overview of matroid algorithmics on representable matroids. He examines the time complexities of fundamental problems in matroid theory for various classes of representable matroids, including partition matroids, gammoids, planar matroids, (co-)graphic matroids, and linear matroids. Nonetheless, he does not go into detail on the operations of contraction and deletion and does not specifically address the problem of finding cocircuits.

Albrecht [19] focuses on gammoids and explores the derivation of the dual gammoid description. This lays the groundwork for finding cocircuits on compact gammoid descriptions. However, he does not provide information on performing contraction and deletion.

Notably, the open-source mathematics software system `SageMath` [20] fully supports operations on two classes of representable matroids: linear and graphic matroids. Although `SageMath` provides all necessary operations, it does so inefficiently. Instead of operating directly on the matroid descriptions, it converts the descriptions to more general and inflated descriptions. This approach is particularly inefficient for more complex matroid instances, as detailed in Section 2.2.3.

1.2 Contribution

This thesis bridges the gap to real-world applications by tailoring the auction of Bikhchandani et al. to multiple classes of representable matroid, allowing it to operate directly on their compact descriptions. For this, we show how the matroid operations employed by the auction work on these compact descriptions. We also refine the matroid operations to meet the unique requirements of the auction, improving their overall efficiency. Additionally, we analyze and compare the theoretical time complexities of our adapted matroid operations. Furthermore, we implement the auction by Bikhchandani et al. and our custom adaptations in `Python` and conduct a running time analysis.

1.3 Overview

Overall, the structure of this thesis is as follows: Chapter 2 establishes the background on auction theory and matroid theory. In Chapter 3, we present the auction by Bikhchandani et al. and provide simplified proofs of its properties based on the analysis by Peis and Rieken [14]. In contrast, Chapters 4 and 5 contain our original contributions. Specifically, Chapter 4 describes how the auction is tailored to several representable matroids, and Chapter 5 showcases the running time analysis.

2 Background

2.1 Auction Theory

2.1.1 Fundamentals

An auction is a market mechanism for allocating items among bidders by offering them up for bid, taking bids, and mapping the bids to an outcome. This outcome includes an allocation of items to bidders and the prices the bidders have to pay. The market involves a set of items E, a set of bidders N, an auctioneer, and a seller. The auctioneer oversees the bidding process, presents the items for sale, and declares the winning bids, while the seller offers the items for sale. In some cases, the roles of auctioneer and seller may be performed by the same entity. Furthermore, every bidder j has a valuation function $v_j : 2^E \to \mathbb{R}_+$ that maps each bundle $S \subseteq E$ to a number representing how much the bidder values this bundle. Definition 2.1 more formally defines auction mechanisms [21].

Definition 2.1. Given a set of possible allocations Ω and a set of bidders N with private valuations $v_j : \Omega \to \mathbb{R}_+$ and bids $b_j : \Omega \to \mathbb{R}_+$ for all $j \in N$. The bids can be considered valuation functions as well, but they are known to the auctioneer and openly communicate the willingness to pay for a bundle. A *bid profile* is a vector of bids $\mathbf{b} = (b_j)_{j \in N}$. An *auction mechanism* is defined by a tuple (x, p) where:

- The *allocation rule* $x : \mathbb{R}_+^N \to \Omega$ is a mapping of bid profiles to allocations.
- The *payment rule* $p : \mathbb{R}_+^N \to \mathbb{R}_+^N$ is a mapping of bid profiles to prices for each bidder.

The most common allocation rule simply awards items to the highest bidder. This is also the allocation rule considered throughout this thesis. A widely used payment rule is pay-as-bid, or first-price, where the winner pays the amount of their highest bid. This payment rule is easy to understand and therefore popular in practical settings. Another common payment rule involves Vickrey prices, where a winner

pays the net effect their presence has on other bidders. We provide a more detailed explanation of these prices in Section 2.1.2 and show how they can lead to desirable auction properties.

The settings of auctions can vary from single-item auctions, where only one item is up for bid, to multi-item auctions, where several goods are auctioned simultaneously. These settings extend to single-unit and multi-unit auctions, which differentiate based on whether one unit or multiple units of the same item are sold.

Valuation functions can take different forms. For instance, concave valuations indicate diminishing marginal utility, meaning each additional unit of the item is valued less than the previous one. In contrast, linear valuations imply that the perceived value increases proportionally with the quantity of the good. Convex valuations represent increasing marginal utility, where the value of each additional unit increases with the quantity.

Auctions can have different objectives. One objective is revenue maximization, which refers to achieving the highest possible price for the auctioned goods. Another objective is the purpose of price discovery, a process through which the market price for goods or services is determined based on the interplay of supply and demand. However, this thesis focuses on the goal of maximizing social welfare.

Welfare maximization ensures that resources are allocated to generate the greatest overall benefit by achieving the highest total utility across all bidders. Formally, the goal is to find an allocation $a^* \in \arg\max_{a \in \Omega} \sum_{j \in N} v_j(a)$. However, note that when the valuations are private, the auctioneer can only find a maximizing allocation regarding the bids, therefore $b^* \in \arg\max_{a \in \Omega} \sum_{j \in N} b_j(a)$. In practical terms, welfare maximization aims to distribute items in a way that maximizes collective satisfaction. For example, governments use welfare-maximizing auctions to allocate electromagnetic spectrum to telecom companies, ensuring that the companies can generate the most value from the acquired spectrum.

Auctions can be categorized into static and dynamic auctions. In static auctions, also known as sealed-bid auctions, bidders only submit the bids once, without knowledge of the bids of their competitors. In a dynamic auction, on the other hand, bidding occurs over several rounds. This format allows bidders to observe the progression of

the auction price and the bids of other participants, enabling them to adjust their bidding strategies throughout the auction.

Strategies define how bidders bid in auctions, considering their valuations of the auctioned item, their information about the valuations of other bidders, and the auction rules. A dominant strategy guarantees the best possible outcome for the bidder, regardless of the actions taken by other participants. In contrast, a weakly dominant strategy ensures an outcome that is at least as good as any other strategy the bidder could choose.

When all participants in an auction perfectly align with their strategies, such that no single bidder benefits from changing their approach, the auction reaches an equilibrium. Different equilibrium concepts exist, such as a dominant strategy equilibrium, where all bidders have found a dominant strategy and, consequently, no reason to change it. A related but slightly weaker equilibrium is the ex-post equilibrium. It occurs when no bidder would want to change their strategy even after learning about the actual valuations of all other bidders. This condition implies that each bidder's strategy is optimal regardless of the specific strategies adopted by others, as long as these strategies are consistent with some underlying valuation function. In other words, for each bidder, there exists an underlying valuation function such that the bids reflect the true valuations of this function. The strength of the ex-post equilibrium is that assumptions about other valuations become irrelevant, as the strategy remains the best choice even with complete information about the other valuations.

2.1.2 VCG Auction

The auction by Bikhchandani et al. incentivizes truthful behavior by charging Vickrey prices. These prices correspond to the Vickrey auction or its generalization to multiple items, the Vickrey-Clarke-Groves (VCG) auction [2, 3, 4].

Definition 2.2. Given the set of possible allocations Ω and a set of bidders N with the bid profile $\mathbf{b} = (b_j)_{j \in N}$. The *VCG mechanism* is characterized by any allocation rule that chooses an allocation $a^* = x(\mathbf{b})$ with:

$$x(\mathbf{b}) \in \arg\max_{a \in \Omega} \sum_{j \in N} b_j(a)$$

and the payment rule for all $j \in N$:

$$p_j(\mathbf{b}) = \max_{a \in \Omega} \sum_{i \in N \setminus \{j\}} b_i(a) - \sum_{i \in N \setminus \{j\}} b_i(a^*)$$

Lemma 2.3. *Truthful bidding is a dominant strategy in the VCG mechanism.*

Proof. The VCG mechanism selects an allocation of items

$$a^* \in \arg\max_{a \in \Omega} \sum_{j \in N} b_j(a)$$

and charges bidder j the net effect their presence has on other bidders, that is:

$$p_j = \max_{a \in \Omega} \sum_{i \in N \setminus \{j\}} b_i(a) - \sum_{i \in N \setminus \{j\}} b_i(a^*)$$

The net utility u_j that bidder j gains is the real utility of the items they have won, minus the price they have paid:

$$\begin{aligned} u_j &= v_j(a^*) - p_j \\ &= v_j(a^*) - \left(\max_{a \in \Omega} \sum_{i \in N \setminus \{j\}} b_i(a) - \sum_{i \in N \setminus \{j\}} b_i(a^*) \right) \\ &= v_j(a^*) + \sum_{i \in N \setminus \{j\}} b_i(a^*) - \max_{a \in \Omega} \sum_{i \in N \setminus \{j\}} b_i(a) \end{aligned}$$

In the expression above, the last term $\max_{a \in \Omega} \sum_{i \in N \setminus \{j\}} b_i(a)$ is independent of the bids of bidder j. Thus, bidder j maximizes their utility by maximizing the first two terms. Suppose bidder j behaves truthfully, then $b_j = v_j$ and:

$$\begin{aligned} \sum_{j \in N} b_j(a^*) &= b_j(a^*) + \sum_{i \in N \setminus \{j\}} b_i(a^*) \\ &= v_j(a^*) + \sum_{i \in N \setminus \{j\}} b_i(a^*) \end{aligned}$$

Recall that a^* is an allocation that maximizes $\sum_{j \in N} b_j(a)$. Therefore, a^* also maximizes $v_j(a^*) + \sum_{i \in N \setminus \{j\}} b_i(a^*)$, and consequently, it maximizes u_j.
Thus, each bidder j maximizes their utility by reporting their valuations truthfully, making truthful bidding a dominant strategy in the VCG mechanism. □

2.2 Matroid Theory

2.2.1 Fundamentals

First introduced in 1935 by Hassler Whitney, matroids are mathematical structures that capture the notion of independence [22]. We recall some basic definitions from matroid theory relevant to this thesis. We mostly use the standard matroid notations by Oxley [17].

Definition 2.4. A *matroid* is an ordered pair $M = (E, \mathcal{I})$, where E is a finite set, called the *ground set*, and $\mathcal{I}$ is a collection of subsets of E, known as the *independent sets*. The independent sets $\mathcal{I}$ adhere to the following axioms:

I1. $\emptyset \in \mathcal{I}$.

I2. If $I_1 \in \mathcal{I}$ and $I_2 \subseteq I_1$, then $I_2 \in \mathcal{I}$.

I3. For any $I_1, I_2 \in \mathcal{I}$ with $|I_1| < |I_2|$, there exists an element $e \in I_2 \setminus I_1$ such that $I_1 \cup e \in \mathcal{I}$.

We write $E(M)$ and $\mathcal{I}(M)$ to denote the ground set and the independent sets of the matroid M.

Definition 2.5. A set $S \subseteq E(M)$ with $S \notin \mathcal{I}(M)$ is called a *dependent set* of M.

Definition 2.6. A *circuit* is an inclusion-wise minimal dependent set. A *loop* is a circuit of cardinality 1. The set of all circuits is denoted by $\mathcal{C}(M)$.

Definition 2.7. A *basis* B is an inclusion-wise maximal independent set. The set of all bases is denoted by $\mathcal{B}(M)$.

Definition 2.8. The *rank* $r(S)$ of a subset $S \subseteq E(M)$ represents the size of the largest independent set contained within S. Formally, $r(S) = \max\{|I| : I \subseteq S \text{ and } I \in \mathcal{I}\}$.
The rank function satisfies the following properties:

I1. $r(S) \geq 0$ for all $S \subseteq E(M)$.

I2. If $S \subseteq T \subseteq E(M)$, then $r(S) \leq r(T)$.

I3. For all $S, T \subseteq E(M), r(S) + r(T) \geq r(S \cup T) + r(S \cap T)$.

I4. $r(\emptyset) = 0$.

Definition 2.9. Let $M = (E, \mathcal{I})$ be a matroid. The *dual* of M, denoted by M^*, is another matroid on the same ground set E but with the set of bases
$\mathcal{B}(M^*) = \{E \setminus B : B \in \mathcal{B}(M)\}$.

The following definitions are fundamental to this thesis, as the operations of finding cocircuits, deletion, and contraction are precisely the matroid operations employed by the auction of Bikhchandani et al. and are therefore relevant to Chapter 4.

Definition 2.10. The circuits of the dual matroid M^* are called *cocircuits* of M and denoted by $C^* \in \mathcal{C}^*(M)$.

Definition 2.11. The *deletion* of a set X in a matroid $M = (E, \mathcal{I})$ yields the matroid
$M \setminus X = (E \setminus X, \mathcal{I}_d)$ with $\mathcal{I}_d = \{I \subseteq E \setminus X : I \in \mathcal{I}(M)\}$.

Definition 2.12. The *contraction* of a set X in M equals the deletion of X in M^*, thus $M/X = (M^* \setminus X)^*$.

Definition 2.13. A *minor* of M is a matroid that is derived from M solely through a series of contractions and deletions. A class of matroids is *closed under minors* if every minor of a matroid in the class also belongs to the class.

Lemma 2.14 (Follows from Proposition 3.1.8 and Corollary 3.1.25 in [17])**.**
Let $M = (E, \mathcal{I})$ and $e \in E$.

- *If e is a loop, then $M/e = M \setminus e$.*
- *If e is not a loop, then $M/e = (E \setminus \{e\}, \mathcal{I}_c)$ with $\mathcal{I}_c = \{I \subseteq E \setminus \{e\} : I \cup \{e\} \in \mathcal{I}(M)\}$.*

Lemma 2.15 (Adapted from Lemma 1.1.3 in [17])**.** *Let $C_1, C_2 \in \mathcal{C}(M)$ with $C_1 \neq C_2$ and $e \in C_1 \cap C_2$. Then there exists $C \in \mathcal{C}(M)$ with $C \subseteq (C_1 \cup C_2) \setminus \{e\}$.*

Lemma 2.16 (Adapted from p.74 in [17])**.** *A cocircuit is a minimal subset $C^* \subseteq E(M)$ that intersects every basis of M.*

Lemma 2.17 (Adapted from Lemma 1 in [14])**.** *Let $C^* \in \mathcal{C}^*$ and $e \in E(M)$. If $e \notin C^*$, then C^* is the union of cocircuits in $M \setminus e$.*

Lemma 2.18 (Adapted from Lemma 2 in [14])**.** *If C^* is a cocircuit of $M \setminus e$, then either C^* or $C^* \cup \{e\}$ is a cocircuit of M.*

2.2.2 Representable Matroids

Definition 2.19. A matroid M is called $\mathbb{F}$-*representable* if there exists a matrix A over a field $\mathbb{F}$ such that M is the linear matroid (see Definition 2.20) associated with A. A matroid is called *representable* if it is $\mathbb{F}$-representable over some $\mathbb{F}$.

We define various classes of matroids and explore their hierarchical relationships, demonstrating that all of these classes are representable. Figure 2.3 illustrates this hierarchy. Notably, these classes can all be described by a matrix, but they may have descriptions that characterize the independent sets even more concisely.

Definition 2.20. Let A be an $m \times n$ matrix over a field $\mathbb{F}$ and let v_i represent the i^{th} column vector of A.
A *linear matroid* is a tuple $M = (E, \mathcal{I})$, where $E = \{1, 2, \ldots, n\}$, and

$$\mathcal{I} = \{I \subseteq E : \text{ vectors } \{v_i\}_{i \in I} \text{ are linearly independent}\}.$$

Lemma 2.21. *Every linear matroid is a matroid.*

Proof. We show that linear matroids satisfy the three matroid axioms in Definition 2.4.
I1. The empty set is trivially linearly independent in any vector space.
I2. Every subset of a linearly independent set of vectors remains linearly independent since removing vectors cannot introduce any new dependency.
I3. Consider two linearly independent sets of vectors X and Y with $|X| < |Y|$. We need to show that there is an element $y \in Y$ such that $X \cup \{y\}$ is linearly independent.
Suppose no such y exists, then every $y \in Y$ would be linearly dependent on X. However, this implies $Y \subseteq \text{span}(X)$, which contradicts the assumption that $|X| < |Y|$ since a set of vectors with fewer elements cannot span a linearly independent set with more elements. □

Example 2.1. Consider the 3×4 matrix:

$$A = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}.$$

The corresponding linear matroid is $M = (E, \mathcal{I})$ with $E = \{1, 2, 3, 4\}$ and
$\mathcal{I} = \{\{1, 2, 4\}, \{1, 3, 4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{1\}, \{2\}, \{3\}, \{4\}, \emptyset\}$.

Definition 2.22. A *multigraph* G is a tuple $G = (V, E)$ where V is a set of vertices and E is a multiset of edges. Consequently, a multigraph is a graph that allows parallel edges.

Definition 2.23. Let $G = (V, E)$ be an undirected multi-graph.
A *graphic matroid* is a tuple $M = (E, \mathcal{I})$ with

$$\mathcal{I} = \{I \subseteq E : \ I \text{ is acyclic in } G\}.$$

Lemma 2.24 (Adapted from Proposition 5.1.2 in [17])**.** *Every graphic matroid is linear.*

Refer to Section 5.2.1 for a detailed explanation of how to convert graphic matroid descriptions to linear matroid descriptions.

Definition 2.25. A *planar graph* is a graph that can be embedded in the plane $\mathbb{R}^2$ such that no two edges intersect except at their endpoints.

Definition 2.26. Let $G = (V, E)$ be an undirected, planar multi-graph.
A *planar matroid* is a tuple $M = (E, \mathcal{I})$ with

$$\mathcal{I} = \{I \subseteq E : \ I \text{ is acyclic in } G\}.$$

Lemma 2.27. *Every planar matroid is graphic.*

Proof. The structures of undirected, planar multigraphs are substructures of undirected multigraphs. □

Example 2.2. Consider the planar graph depicted in Figure 2.1. The corresponding graphic (and planar) matroid is $M = (E, \mathcal{I})$ with $E = \{\{1,2\},\{1,3\},\{2,3\}\}$ and $\mathcal{I} = \{\{\{1,2\},\{1,3\}\},\{\{1,2\},\{2,3\}\},\{\{1,3\},\{2,3\}\},\{\{1,2\}\},\{\{1,3\}\},\{\{2,3\}\},\emptyset\}$.

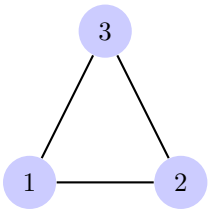


Figure 2.1: A planar graph with edges $E = \{\{1,2\},\{1,3\},\{2,3\}\}$.

Definition 2.28. Let $D = (V, \mathcal{A})$ be a directed graph where V is a set of vertices and $\mathcal{A}$ is a set of directed edges. Let $X, Y \subseteq V$ be sets of vertices.
A *routing* from X to Y is a family of vertex-disjoint paths connecting each vertex $x \in X$ to a distinct vertex $y \in Y$ in the graph D.

Definition 2.29. Let $D = (V, \mathcal{A})$ be a directed graph where V is a set of vertices and $\mathcal{A}$ is a set of directed edges. Let $E \subseteq V$ be the set of starting vertices and $T \subseteq V$ the set of destination vertices.
A *gammoid* is a tuple $M = (E, \mathcal{I})$ with

$$\mathcal{I} = \{I \subseteq E : \text{ there is a routing from } I \text{ to } T\}.$$

Lemma 2.30 (Adapted from Proposition 3.9 in [23])**.** *Every gammoid is linear.*

Example 2.3. Consider starting vertices $E = \{1, 2, 3\}$, destination vertices $T = \{4, 5\}$, and the directed graph in Figure 2.2. The corresponding gammoid is $M = (E, \mathcal{I})$ with $E = \{1, 2, 3\}$ and $\mathcal{I} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \emptyset\}$.

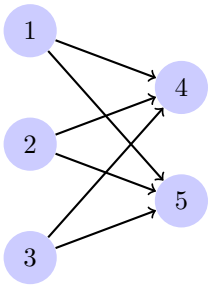


Figure 2.2: Directed graph with vertices $V = \{1, 2, 3, 4, 5\}$.

Definition 2.31. Let E be a finite set and k a non-negative integer.
A *uniform matroid* is a tuple $M = (E, \mathcal{I})$ with

$$\mathcal{I} = \{I \subseteq E : |I| \leq k\}.$$

Lemma 2.32. *Every uniform matroid is a gammoid.*

Proof. Suppose M_u is a uniform matroid with ground set E and non-negative integer k. Consider the gammoid M_g with starting vertices E, destination vertices $T = \{t_1, t_2, \ldots, t_k\}$ with $E \cap T = \emptyset$, and the directed graph $D = (V, \mathcal{A})$ with $V = E \cup T$ and $\mathcal{A} = \{(e, f) : \text{for all } e \in E \text{ and for all } f \in T\}$. We show that M_u and M_g are the same matroid.

Case 1: Suppose $I \in \mathcal{I}(M_u)$. Then, $|I| \leq k$. Since $|T| = k$ and there is an edge from each starting to each destination vertex, there is a family of vertex-disjoint paths connecting each starting vertex $e \in I$ to a different destination vertex $f \in T$. Therefore, there is a routing from I to T, and consequently, $I \in \mathcal{I}(M_g)$.
Case 2: Suppose $I \notin \mathcal{I}(M_u)$. Then, $|I| > k$. Since $|T| = k$, there are not enough destination vertices to connect each starting vertex $e \in I$ to a different destination vertex $f \in T$. Therefore, there cannot exist a routing from I to T, and consequently, $I \notin \mathcal{I}(M_g)$. □

Example 2.4. Let $E = \{a, b, c\}$ and k = 2. The corresponding uniform matroid is $M = (E, \mathcal{I})$ with $E = \{a, b, c\}$ and $\mathcal{I} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a\}, \{b\}, \{c\}, \emptyset\}$.

Definition 2.33. Given matroids $M_1, M_2, \ldots, M_k$ with disjoint ground sets $E_1, E_2, \ldots, E_m$ and families of independent sets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_m$.
The *direct sum* of these matroids $M = M_1 \oplus M_2 \oplus \ldots \oplus M_m$ is a matroid with ground set $E = E_1 \cup E_2 \cup \ldots \cup E_m$ and the family of independent sets

$$\mathcal{I} = \{I \subseteq E : I \cap E_i \in \mathcal{I}_i \text{ for all } 1 \leq i \leq m\}.$$

Definition 2.34. Let $P_1, P_2, \ldots, P_m$ be a partition of a finite set E and let $k_1, k_2, \ldots, k_m$ be non-negative integers.
A *partition matroid* is a tuple $M = (E, \mathcal{I})$ with

$$\mathcal{I} = \{I \subseteq E : |I \cap P_i| \leq k_i \text{ for all } 1 \leq i \leq m\}.$$

Lemma 2.35. *Every uniform matroid is a partition matroid.*

Proof. Note that partition matroids can be expressed as direct sums of uniform matroids. Since a uniform matroid can trivially be viewed as a direct sum of itself, any uniform matroid is a partition matroid. □

Lemma 2.36. *Every partition matroid is a gammoid.*

Proof. By Lemma 2.32, every uniform matroid is a gammoid. Since gammoids are closed under direct sum operations [19], it follows that partition matroids, being direct sums of uniform matroids, are also gammoids. □

Example 2.5. Let $E = \{a, b, c, d, e, f\}$ be partitioned in $P_1 = \{a, b\}, P_2 = \{c, d, e\}$, $P_3 = \{f\}$ and let $k_1 = 1, k_2 = 1, k_3 = 1$. The corresponding partition matroid is

$M = (E, \mathcal{I})$ with $E = \{a, b, c, d, e, f\}$ and $\mathcal{I} = \{\{a, c, f\}, \{a, d, f\}, \{a, e, f\}, \{b, c, f\}, \{b, d, f\}, \{b, e, f\}, \{a, c\}, \{a, d\}, \{a, e\}, \{a, f\}, \{b, c\}, \{b, d\}, \{b, e\}, \{b, f\}, \{c, f\}, \{d, f\}, \{e, f\}, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \emptyset\}$.

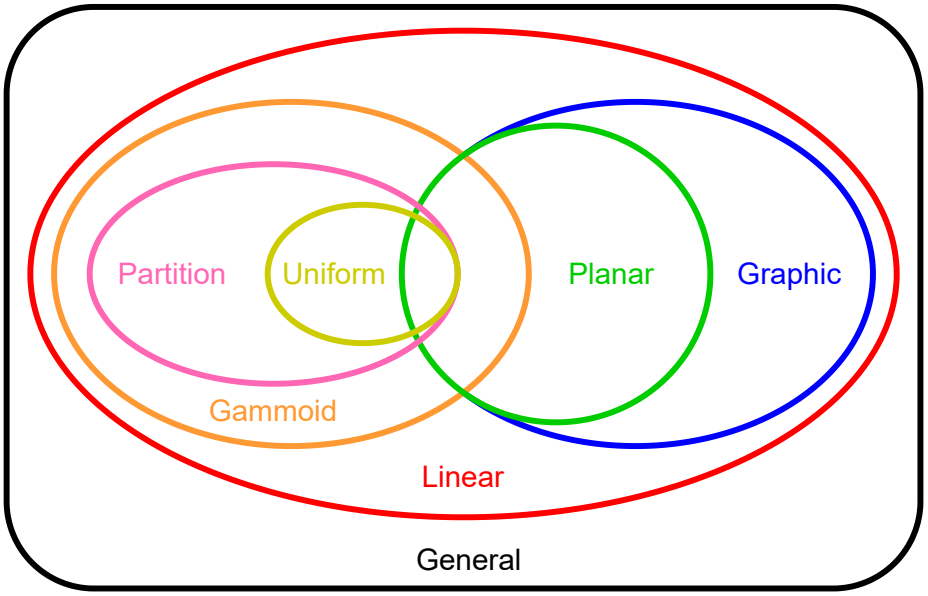


Figure 2.3: Hierarchy of matroid classes without considering scale [18]. Representable matroids include linear matroids and all their subclasses.

2.2.3 Descriptions

Matroids can be described in various ways. While these descriptions may define the same matroid, they often differ significantly from a computational point of view. Three main approaches have emerged for describing matroids.

Consider a matroid $M = (E, \mathcal{I})$ and a deterministic Turing machine. The first approach describes M to the Turing machine via families of subsets on E, such as its independent sets, circuits, cocircuits, and bases [24]. Although this method is straightforward and universally applicable to all matroid classes, it can result in an input size that is superpolynomial in $|E|$ [25], and therefore impractical.

The second method is more widely used and addresses the issue of inflated inputs.

It abandons the traditional deterministic Turing machine model and instead uses an altered version equipped with an oracle. This oracle can provide information about a subset of the ground set within polynomial time. For example, an independence oracle determines if a specified set is independent in polynomial time. Bikhchandani et al. use a cocircuit oracle in their auction, which returns in polynomial time whether a set is a cocircuit [1].

The problem with the first two approaches is that, while good for theoretical analysis, they are difficult to apply to real-world situations. The third approach abandons the ability to accept all matroids as input and restricts the input to representable matroids. As discussed in Section 2.2.2, representable matroids have compact descriptions. A uniform matroid, for instance, can be described by its ground set E and integer k. This means that the input is relatively small. Furthermore, representable matroids are well-suited for real-world situations as many problems can be modeled through representable matroids. For example, minimal network planning problems can be modeled by graphic matroids. Additionally, scheduling problems can be reduced to matching problems in bipartite graphs, which in turn can be modeled using gammoids.

3 The Auction by Bikhchandani et al.

We present the auction by Bikhchandani et al. [1] and show that the auction maximizes social welfare under truthful bidding. Moreover, we show that the auction incentivizes truthful bidding by showing that truthful bidding results in an ex-post equilibrium. Additionally, we prove that the auction operates within polynomial time.

3.1 Economic Model

The auction aims to allocate a bundle of indivisible, unique goods among various bidders while maximizing social welfare. Let E be the set containing the goods and N the set containing the bidders. The set E is partitioned into sets $(E_j)_{j \in N}$ and we assume that a bidder j is exclusively interested in the items E_j. As demonstrated in the final paragraph of this section, this assumption is made without loss of generality.

If bidder j acquires element $e \in E_j$, it provides them the value $v_j(e) \in \mathbb{R}_+$ and a set $S \subseteq E_j$ provides them the value $v_j(S) \coloneqq \sum_{e \in S} v_j(e)$. As no two bidders demand the same good, thus making it clear to whom a valuation belongs, we sometimes write $v(e)$ instead of $v_j(e)$.

Furthermore, a matroid $M = (E, \mathcal{I})$ constrains the feasible allocations of goods to bidders. The seller is only allowed to sell combinations of elements of E that form an independent set in this matroid. As the valuations are all non-negative, we assume w.l.o.g. that the feasible solutions are inclusion-wise maximum independent sets and, therefore, bases. Thus, the seller is constrained to sell a basis of M. The matroid in this auction is provided through a cocircuit oracle.

We also assume that no bidder controls a cocircuit before the auction begins. This means that $C^* \not\subseteq E_j$ for all $C^* \in \mathcal{C}^*(M)$ and for all $j \in N$. This assumption is without loss of generality. Suppose such a cocircuit C^* exists. For a random element $e \in C^*$, we add a parallel element e' and replace M with $M' = (E', \mathcal{I}')$, where $E' = E \cup \{e'\}$ and $\mathcal{I}' = \mathcal{I} \cup \{I \cup \{e'\} : I \cup \{e\} \in \mathcal{I}, e \notin I\}$. Furthermore, we add a new bidder that values e' at 0. This ensures that C^* is no longer a cocircuit since there exists a basis containing e' instead of e. However, since the item gets dropped at the beginning of the auction, this does not change the overall outcome.

The assumption that no two bidders demand the same good is without loss of generality as well. In cases where two participants are interested in the same item e, one can add a parallel element e' and specify that the first agent exclusively demands e, while the second agent is solely interested in e'. By replacing M with $M' = (E', \mathcal{I}')$, where $E' = E \cup \{e'\}$ and $\mathcal{I}' = \mathcal{I} \cup \{I \cup \{e'\} : I \cup \{e\} \in \mathcal{I}, e \notin I\}$, each independent set contains at most one of $\{e, e'\}$. Consequently, at most one of $\{e, e'\}$ can be sold.

3.2 The Auction

The auction by Bikhchandani et al. is based on Dawson's cocircuit greedy algorithm for finding the maximum weight basis of a matroid [26].

Algorithm 1: Dawsons Cocircuit Greedy Algorithm

Input: matroid M and distinct item valuations

Output: max-weight basis of M

```
B ← ∅
while there exists C* ∈ C*(M) with C* ∩ B = ∅ do
    let b_{C*} be the highest valued element in C*
    B ← B ∪ {b_{C*}}
```

Result: B

The auction builds on this algorithm by only awarding an element to a bidder j if bidder j controls a cocircuit in a specific minor of M; thus, if there exists a cocircuit C^* in the minor with $C^* \subseteq E_j$. Moreover, the auction consistently awards the highest-valued element of such a cocircuit under truthful bidding. We show in Section 3.3 that this results in a welfare-maximizing basis.

The general idea of the ascending auction is as follows: By price increments, the bidders give up interest in the items they value less than the current price. Hence, these items are deleted from the matroid. Because of the deletion, some other bidder might gain control of a cocircuit. In such a case, this bidder is awarded an element of their choice in this cocircuit, which is the highest-valued element if the bidder behaves truthfully. Furthermore, this awarded item is contracted and added to the final basis. The bidder loses control over this cocircuit through the contraction and the auction continues.

3.2.1 Unit-Step Auction

The unit-step auction increases the price by one unit at a time. While this is straightforward, it is less efficient than the long-step auction with larger price increments. Additionally, it only works with integer valuations.

Auction 2: Unit-Step Auction

Input: matroid M with ground set E and a cocircuit oracle,
partition $E = \bigcup_{j \in N} E_j$ with $C^* \nsubseteq E_j$ for all $C^* \in \mathcal{C}^*(M)$ and all $j \in N$,
non-negative integer valuations

Output: max-weight basis of M

```
1  p ← 0, B ← ∅
2  while E(M) ≠ ∅ do
3      ask bidders to determine F ← {f ∈ E : v(f) = p}
4      foreach f ∈ F do
5          M ← M \ {f}
6          foreach j ∈ N do
7              if E_j contains a cocircuit C* then
8                  ask bidder j to choose b ∈ C*
9                  award b to bidder j and charge them p
10                 M ← M/{b}, B ← B ∪ {b}
11     p ← p + 1
```

Result: B

This version of the auction is slightly different from the auction by Bikhchandani et al. to make it more suitable for the implementation in Chapter 5. The most notable

change is in lines 6-7, where we introduce a `foreach` loop and an `if` statement that replace the original line:

while there exists a bidder j and a cocircuit C^* of M with $C^* \subseteq E_j$ **do**

To show that the replacement is justified, we first establish Lemma 3.1.

Lemma 3.1. *Let M be a matroid where no bidder controls a cocircuit, thus $C^* \nsubseteq E_j$ for all bidders j and all $C^* \in \mathcal{C}^*(M)$. Then, each bidder controls at most one cocircuit in $M \setminus f$.*

Proof. Suppose no bidder controls a cocircuit in M but $M \setminus f$ contains two cocircuits C_1^*, C_2^* with $C_1^* \neq C_2^*$ and $C_1^*, C_2^* \in E_j$ for some bidder $j \in N$. Then by Lemma 2.18, C_1^* or $C_1^* + f$ is a cocircuit of M and C_2^* or $C_2^* + f$ is a cocircuit of M. Since no bidder controls a cocircuit in M and $C_1^*, C_2^* \in E_j$, C_1^* and C_2^* cannot be cocircuits in M. Thus, $C_1^* + f$ and $C_2^* + f$ must be the two cocircuits in M. But then $C_1^* + f$ and $C_2^* + f$ both contain f. Hence, by Lemma 2.15 applied to the dual matroid, there exists a cocircuit $C^* \subseteq ((C_1^* + f) \cup (C_2^* + f)) \setminus \{f\} = C_1^* \cup C_2^* \subseteq E_i$. This contradicts the assumption that no bidder controls a cocircuit in M. □

By Lemma 3.1, no bidder can control more than one cocircuit after an element f is deleted. Furthermore, the contraction in line 10 cannot lead to new cocircuits [17]. Therefore, it is sufficient to perform a single oracle call per bidder.

Example 3.1. Let $M = (E, \mathcal{I})$ be a matroid with $E = \{1, 2, 3, 4\}$ and

$$\mathcal{I} = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}, \{1\}, \{2\}, \{3\}, \{4\}, \emptyset\}.$$

Observe that M is the uniform matroid with the ground set E and $k = 2$. Suppose that the matroid is provided through a cocircuit oracle. To visualize this, we state the set of cocircuits at any point in the auction. Initially, the set of cocircuits is given by:

$$\mathcal{C}^* = \{\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}\}.$$

In this example, we do not specify how contraction and deletion affect the cocircuits, and we assume that the cocircuit oracle manages these operations for us. Consider the bidders $\{A, B, C\}$ and their item valuations as presented in Table 3.1. Observe

that each bidder is interested in different items and no bidder controls a cocircuit. Furthermore, we assume that the bidders behave truthfully.

Bidder	$v(1)$	$v(2)$	$v(3)$	$v(4)$
A	1			
B		5		
C			6	2

Table 3.1: Item valuations for bidders A, B, and C.

$\mathbf{p = 0}$: The cocircuits are $\mathcal{C}^*(M) = \{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\}\}$, i.e., all 3-element subsets of E. No bidder controls a cocircuit, since neither $E_A = \{1\}$, $E_B = \{2\}$, nor $E_C = \{3,4\}$ contains a cocircuit. Therefore, the price is incremented by one unit.

$\mathbf{p = 1}$: The valuation of bidder A for item 1 is reached. Thus, bidder A does not gain additional utility when acquiring item 1. Bidder A drops interest item 1, and item 1 is deleted.

The new cocircuits are $\mathcal{C}^*(M \setminus \{1\}) = \{\{2,3\},\{2,4\},\{3,4\}\}$.

Now, Bidder C controls a cocircuit since $E_C = \{3,4\}$ is a cocircuit in itself. Bidder C chooses item 3 in this cocircuit since $v_C(3) = 6 > 2 = v_C(4)$. Therefore, item 3 is awarded to the bidder C at a price of 1 and contracted.

The new cocircuits are $\mathcal{C}^*(M \setminus \{1\}/\{3\}) = \{\{2,4\}\}$.

No bidder controls a cocircuit, since neither $E_B = \{2\}$ nor $E_C = \{3,4\}$ contains the set $\{2,4\}$. Therefore, the price is incremented by one unit.

$\mathbf{p = 2}$: The valuation of bidder C for item 4 is reached. Bidder C drops interest in item 4 and item 4 is deleted.

The new set of cocircuits is $\mathcal{C}^*(M \setminus \{1,4\}/\{3\}) = \{\{2\}\}$.

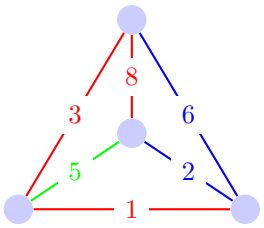
Bidder B controls a cocircuit, since $E_B = \{2\}$ is a cocircuit in itself. Bidder B chooses item 2 and item 2 is awarded to bidder B at a price of 2. Furthermore, item 2 is contracted.

Now, all elements are either deleted or awarded and the auction concludes.

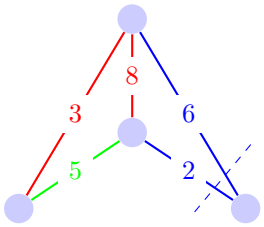
The welfare-maximizing basis is $B = \{2,3\}$.

However, as explained in Section 2.2.3, describing matroids via families of subsets of E or through oracles is impractical. It also remains open how the operations of deletion and contraction are performed. Example 3.2 demonstrates how the auction can operate on the compact description of a representable matroid, namely a graphic matroid. To see how this can be implemented computationally, see Section 4.2.4.

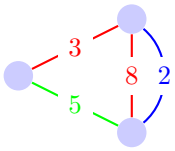
Example 3.2. Figure 3.1 illustrates the auction on a graphic matroid with three bidders: red, green, and blue. The colors of the edges indicate the associated bidder, and the labels indicate the valuations. Note that in a graphic matroid, cocircuits correspond to minimal cuts, deletion refers to edge deletion, and contraction refers to edge contraction. For more details on these operations, refer to Section 4.2.4.



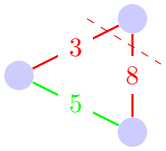
(a) p=0: No bidder controls a cocircuit.



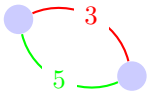
(b) p=1: The edge with a weight of 1 is deleted. Now blue controls a cocircuit.



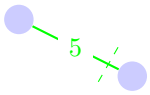
(c) p=1: The edge with a weight of 6 is awarded to bidder blue and contracted.



(d) p=2: The edge with a weight of 2 is deleted. Now red controls a cocircuit.



(e) p=2: The edge with a weight of 8 is awarded to bidder red and contracted.



(f) p=3: The edge with a weight of 5 is awarded to bidder green and contracted. The auction concludes.

Figure 3.1: Example of Auction 2 on a graphic matroid with bidders: red, green, and blue. The colors of the edges indicate the associated bidder, and the labels indicate the valuations.

3.2.2 Long-Step Auction

The unit-step auction does not operate in polynomial time in the number of items $|E|$ because the running time depends on the highest item valuation. Furthermore, the unit-step auction has the flaw that it does not work with non-integer valuations since the price always increases by one unit.
The long-step auction addresses these problems by adjusting how the price is incremented. Instead of incrementing in unit steps, the auction asks the bidders to report the valuation of the lowest-valued item they are still interested in buying.
Thus, line 11 is replaced with:

$$\text{Ask each bidder } j \text{ to determine } u_j = \min_{e \in E_j} v_e > p$$

$$p \leftarrow \min_{j \in N} u_j$$

Furthermore, a consistency check needs to be implemented. Otherwise, an untruthful bidder could report a valuation of $u_j = p + 2^{-t}$ for every round t and prolong the auction indefinitely.
The consistency check ensures that bidders drop their interest in items if the reported minimum valuation is reached.
We replace line 3 with:

$$\text{ask bidders to determine } F \leftarrow \{f \in E : v(f) = p\},$$
$$\text{every bidder } j \text{ with } u_j = p \text{ must announce at least one element}$$

3.3 Properties

We present the simplified analysis by Peis and Rieken [14].

3.3.1 Welfare Maximization

We show that Auction 2 is welfare-maximizing if bidders reveal their valuations truthfully.

Definition 3.2. Let $M = (E, \mathcal{I})$ be a matroid and $w : E \to \mathbb{R}$ a weight function. An independent set $I \in \mathcal{I}$ is *max-weight basis extendable* in M if there exists a basis $B^* \in \mathcal{B}(\mathcal{M})$ that has maximum weight and $I \subseteq B^*$.

Lemma 3.3 (Adapted from Lemma 3 in [14]). *Let $I \in \mathcal{I}(M)$ be max-weight basis extendable in M, and let e be an element of maximum weight in some cocircuit C^* of M/I. Then $I \cup \{e\}$ is max-weight basis extendable in M.*

Theorem 3.4 (Adapted from Theorem 4 in [14]). *Auction 2 returns a welfare-maximizing allocation under truthful bidding.*

Proof. Let B be the set that contains the awarded elements during the auction and assume that all bidders are bidding truthfully. We show that Auction 2 returns a max-weight basis by showing that B maintains max-weight basis extendable throughout the auction.
Base Case: The auction starts with $B = \emptyset$. The empty set is a subset of any basis, and therefore max-weight basis extendable in M.
Induction Hypothesis: Assume that at some stage of the auction, the set B of awarded elements is max-weight basis extendable. Let D be the set that contains the deleted elements at this stage.
Induction Step: The auction adds an item e to the set B only if e has the maximum value among all items in some cocircuit of $M \setminus D/B$. Since $v(e) \geq v(f)$ for all $f \in D$, it follows from Lemma 2.18 that e has the maximum weight in some cocircuit of M/B. By Lemma 3.3, the set $B \cup \{e\}$ is max-weight basis extendable. □

3.3.2 Truthful Ex-Post Equilibrium

Theorem 3.4 has limitations since bidders can decide against bidding truthfully. We demonstrate that the auction by Bikhchandani et al. does incentivize truthful bidding. Unfortunately, unlike the VCG auction, truthful bidding is not a dominant strategy. The problem lies in the dynamic nature of this auction, which allows bidders to observe and respond to the actions of their competitors, potentially modifying their bids to punish or reward others based on observed behaviors. This interactivity means no single strategy can consistently dominate, as another bidder's reactive behavior can counter any strategy. Nevertheless, we show that truthful bidding is an ex-post equilibrium strategy.

Lemma 3.5 (Adapted from Theorem 4 in [14]). *Auction 2 charges Vickrey Prices.*

Proof. Let f_e be the element that is deleted in the Auction 2 right before the element e is awarded. Let B^* be the max-weight basis returned by Auction 2. To show that the prices f_e in the auction are Vickrey Prices, we show for all $j \in N$ that

$B_j^* = B^* \setminus E_j \cup \{f_e : e \in B^* \cap E_j\}$ is a basis of maximum value in the matroid $M \setminus E_j$.

First, we show that each B_j^* has the same cardinality as B^*.
Since, by Lemma 3.1, every bidder can control at most one cocircuit, only a single element can be awarded to any bidder after the deletion of an element f_e. Thus, every f_e replaces at most one item in B^* and $|B^*| = |B_j^*|$.

Next, we show that each B_j^* has maximimum weight in $M \setminus E_j$.
Let $B_j = B \setminus E_j \cup \{f_e : e \in B \cap E_j\}$ where B is the set of awarded elements maintained throughout the auction. We show that each B_j constantly maintains max-weight basis extendable in $M \setminus E_j$ throughout the auction, similar to how we showed for Theorem 3.4 that B constantly maintains max-weight basis extendable in M. This implies for all $j \in N$ that B_j^* is a maximum weight basis of $M \setminus E_j$.
Base Case: The auction starts with $B_j = \emptyset$. The empty set is a subset of any basis, and therefore max-weight basis extendable in $M \setminus E_j$.
Induction Hypothesis: Assume that at some stage of the auction, the set B_j is max-weight basis extendable in $M \setminus E_j$. Let B be the set containing all awarded elements and let D be the set containing the deleted elements at this stage of the auction.
Induction Step:
Case 1: Assume that an element $e \notin E_j$ is awarded to a bidder i with $i \neq j$. We show that $B_j \cup \{e\}$ is max-weight basis extendable in $M \setminus E_j$.
Note that $v(e) \geq v(f)$ for all $f \in D$. Since e is awarded to some bidder i with $i \neq j$, e has maximum value in some cocircuit C^* in $M \setminus D/B$ with $C^* \subseteq E_i$. According to Lemma 2.18, e has maximum value in a cocircuit of M/B. Furthermore, Lemma 2.17 establishes that e retains the maximum value element within a cocircuit of $M \setminus E_j/B$. By Lemma 3.3, the set $B_j \cup \{e\}$ is max-weight basis extendable in $M \setminus E_j$.
Case 2: Assume that an element $e \in E_j$ is awarded to bidder j. We show that $B_j \cup \{f_e\}$ is max-weight basis extendable in $M \setminus E_j$.
Note that $v(e) \geq v(f)$ for all $f \in D$. Since e is awarded to bidder j, e is an item of maximum value in some cocircuit C^* of $M \setminus D/B$ with $C^* \subseteq E_j$. According to Lemma 2.18, e has maximum value in some cocircuit of M/B. Furthermore, note that $f_e \notin E_j$ as then by Lemma 2.18 either C^* or $C^* + f_e$ would have been a cocircuit entirely contained in E_j before removing f_e. Thus, by Lemma 2.17, f_e has maximum value in some cocircuit of $M \setminus E_j/B$. By Lemma 3.3, $B_j \cup \{f_e\}$ is max-weight basis

extendable in $M \setminus E_j$. □

Lemma 3.6 (Adapted from Lemma 15 in [1])**.** *Suppose each bidder is bidding truthfully. Let s be the strategy where bidder j bids truthfully according to their true valuation, and let $\hat{s}$ represent a strategy where bidder j bids exactly according to a different valuation. Then, the payoff from strategy s is at least as great as the payoff from strategy $\hat{s}$.*

Proof. The proof is analogous to the proof of Lemma 2.3. □

Lemma 3.7 (Adapted from Lemma 16 in [1])**.** *Suppose bidder j bids according to some arbitrary strategy s and all other bidders are bidding truthfully according to their real valuations. Then, there exists some valuation function $\hat{v}$, not necessarily the real valuation, such that the payoff of bidding exactly according to this valuation $\hat{v}$ is at least as great as the payoff from using strategy s.*

Proof. Suppose bidder j bids according to some arbitrary strategy s and all other bidders are bidding truthfully according to their real valuations.
Assume bidder j is awarded the elements $\{e_1, \ldots, e_k\}$ in that order. Let p^* be the highest price reached before the auction concluded.

For all $e \in E_j$, we define a new valuation $\hat{v}_j$ with:

$$\hat{v}_j(e) = \begin{cases} (1 + p^*) + (k - l) & \text{if } e = e_l \in \{e_1, \ldots, e_k\} \\ \text{price } p \text{ where bidder } j \text{ dropped interest in } e & \text{if } e \in E_j \setminus \{e_1, \ldots, e_k\} \end{cases}$$

Consequently, any awarded element has a higher valuation than any dropped element, so $\hat{v}_j(e) > \hat{v}_j(f)$ for all $e \in \{e_1, \ldots, e_k\}$ and $f \in E_j \setminus \{e_1, \ldots, e_k\}$.
Furthermore it holds that $\hat{v}_j(e_i) > \hat{v}_j(e_l)$ for all $1 \leq i < l \leq k$.

Let $\hat{s}$ be the strategy where bidder j bids exactly according to $\hat{v}_j$.
We show that bidder j using strategy $\hat{s}$ does not change the outcome of the auction.

The only difference between strategies s and $\hat{s}$ is that under strategy s, bidder j might announce an element e in line 3 that he is still awarded later, while under strategy $\hat{s}$, bidder j never drops interest in the items $\{e_1, \ldots, e_k\}$.
Let $f_1, \ldots f_l$ be the critical elements that are announced together with e in line 3 and whose deletion leads to bidder j controlling a cocircuit and element e being

awarded.
In both strategies, the deletion of $f_1, \dots f_l$ leads to bidder j controlling a cocircuit, so element e is awarded at exactly the same time.
Consequently, the outcome of the auction is the same for all bidders, regardless whether bidder j uses strategy s or $\hat{s}$. □

Theorem 3.8 (Adapted from Theorem 17 in [1])**.** *For any profile of valuations, the profile of truthful bidding strategies is an ex-post equilibrium of Auction 2.*

Proof. Suppose bidder j bids according to some arbitrary strategy s and all other bidders are bidding truthfully according to their real valuations. By Lemma 3.7, there exists some valuation function $\hat{v}$ for bidder j, not necessarily their true valuation function, such that the payoff of bidding exactly according to this valuation $\hat{v}$ is at least as great as the payoff from using strategy s.
Lemma 3.6 implies that the payoff from bidding truthfully to the real valuation function yields the same payoff as from bidding according to $\hat{v}$.
Hence, under these conditions, no bidder has an incentive to change their strategy after the auction outcomes are realized. □

3.3.3 Polynomial Running Time

Theorem 3.9. *The long-step auction goes through a polynomial number of steps in the number of items $|E|$.*

Proof. The outer `while` loop is executed no more than $|E|$ times because the consistency check guarantees that, in each iteration, at least one critical element is reported in line 3 and subsequently contracted in line 5. Similarly, the outer `foreach` loop is limited to at most $|E|$ executions, corresponding to the maximum number of critical elements that might be reported in line 3. The inner `foreach` loop is executed at most $|N|$ times while making a polynomial time oracle call in line 7. Since $|N| \leq |E|$, the auction only goes through a polynomial number of steps in the number of items $|E|$. □

4 Tailoring the Auction to Representable Matroids

To directly run Auction 2 on compact matroid descriptions, we show how the matroid operations used in the auction operate on these descriptions. The operations employed by the auction are: finding cocircuits, deletion, and contraction.

4.1 Finding Cocircuits

We use two different approaches for determining if a bidder j controls a cocircuit in a matroid M, so if there exists $C^* \in \mathcal{C}^*(M)$ with $C^* \subseteq E_j$.
Note that by Lemma 3.1, no bidder can control more than one cocircuit at any time during the auction. Thus, it is sufficient to identify a single, unique cocircuit within a given set E_j.

4.1.1 Dual Approach

By definition, cocircuits of M are circuits of the dual matroid M^*. Consequently, a straightforward approach is to derive the description of the dual matroid and to determine whether a bidder j controls a circuit in M^*, specifically, whether there exists a circuit $C \in \mathcal{C}(M^*)$ such that $C \subseteq E_j$. Algorithm 3 outlines the procedure for identifying circuits after the dual description has been arrived. However, it is important to note that while Algorithm 3 provides a general procedure, circuits can possibly be identified more efficiently. For instance, in the case of a uniform matroid, circuits can be identified in constant time.
Once the dual description is derived, it can be reused throughout the auction, provided the class is closed under deletion and contraction. Thus, the initial effort to derive the dual is amortized over the entire auction period. Furthermore, searching for a circuit in the dual requires only examining the elements in E_j, as shown in Algorithm 3. This is in contrast to the primal approach, where finding a cocircuit

requires examining all of E.

Algorithm 3: Finding a Circuit in the Dual Matroid in Auction 2

Input: matroid $M^* = (E, \mathcal{I}^*)$, $E_j \subseteq E$

Output: circuit $C \in \mathcal{C}(M^*)$ with $C \subseteq E_j$ or $\emptyset$ if no such circuit exists

```
1 if E_j ∈ I* then
2 |   C ← ∅
3 else
4 |   C ← E_j
5 |   foreach e ∈ C do
6 |   |   if C \ {e} ∉ I* then
7 |   |   |   C ← C \ {e}
```

Result: C

Lemma 4.1. *Algorithm 3 correctly identifies a circuit $C \in \mathcal{C}(M^*)$ in Auction 2.*

Proof. If E_j is independent in M^*, then E_j does not contain a circuit and Algorithm 3 returns $C = \emptyset$. Otherwise, E_j is dependent in M^* and E_j contains a circuit. By Lemma 3.1, any bidder j controls at most one cocircuit in the matroid M, and therefore, at most one circuit in M^*. Algorithm 3 identifies this circuit by iteratively removing all elements from E_j whose deletion does not cause E_j to become independent. □

In this dual-based approach, only the description of M^* is needed, and the original description of M can be disregarded. This implies that the operations of deletion and contraction are reversed; deletion in M corresponds to contraction in M^*, and vice versa.

Note, however, that the dual approach only directly applies to matroid classes that are closed under duality. For instance, according to Lemma 4.40, the class of graphic matroids is not closed under duality. If a class is not closed under duality, the matroid description must be converted to that of a more general matroid class that is closed under duality. This conversion can be complex and introduces a new layer of abstraction, which we would like to avoid. For example, we aim to model network planning problems using graphic matroids directly, as running the auction on a more general description complicates the association with the real-world setting. Additionally, there may be cases where it is inefficient or even unknown how to

derive the dual description.

However, the dual approach can potentially be slower than the primal approach, even though finding cocircuits requires examining fewer elements. For instance, in cases where the dual description is significantly larger than the original description. For linear matroids, as shown in Section 4.3, the rows in the dual description equal $|E| - r$, where r is the rank of the original matrix. This means that for a matrix with few rows and many columns, the dual description would have the same amount of columns but many rows. Furthermore, while the initial effort of deriving the dual matroid description is amortized, it is not always justified. For example, when almost all cocircuit lookups result in a contraction, the time spent finding cocircuits is relatively low compared to the other operations in the auction. In such scenarios, the effort required to derive the dual matroid may outweigh the time savings achieved in cocircuit lookups compared to using the primal approach.

4.1.2 Primal Approach

Even if the dual matroid is not constructible or not efficiently constructible, it is still possible to identify the cocircuits by following Algorithm 4.

Lemma 4.2. *Let E_j be the elements of interest to a bidder j.*
There exists a cocircuit C^* in $M = (E, \mathcal{I})$ with $C^* \subseteq E_j$ if and only if

$$r(E) > r(E \backslash E_j).$$

Proof. Recall that cocircuits are the inclusion-wise minimal sets that intersect all bases of the matroid. A bidder j controls a cocircuit in M if and only if there exists $C^* \in \mathcal{C}^*$ with $C^* \subseteq E_j$, therefore if and only if $E_j \cap B \neq \emptyset$ for all $B \in \mathcal{B}(M)$. Furthermore, $E_j \cap B \neq \emptyset$ for all $B \in \mathcal{B}(M)$ if and only if deleting E_j removes at least one element from each basis B, hence if $r(E) > r(E \setminus E_j)$. □

Lemma 4.3. *Let E_j be the elements of interest to a bidder j.*
If there exists a cocircuit C^* in $M = (E, \mathcal{I})$ with $C^* \subseteq E_j$, then

$$C^* = \{e \in E_j : r((E \setminus E_j) \cup \{e\}) > r(E \setminus E_j)\}.$$

Proof. Suppose there exists a cocircuit C^* in $M = (E, \mathcal{I})$ with $C^* \subseteq E_j$. By Lemma 4.2, this means that $r(E) > r(E \setminus E_j)$. More precisely, since each bidder can

control at most one cocircuit at any time in the auction by Lemma 3.1, it holds that $r(E) = r(E \setminus E_j) + 1$. Now, let $e \in E$ be an element with $r((E \setminus E_j) \cup \{e\}) > r(E \setminus E_j)$. Then, e is part of the inclusion-wise minimal set in E_j whose deletion decreases the rank of the matroid. Thus, $e \in C^*$. □

Algorithm 4: Finding a Cocircuit in the Original Matroid in Auction 2

Input: matroid $M = (E, \mathcal{I})$, $E_j \subseteq E$

Output: cocircuit $C^* \in \mathcal{C}^*(M)$ with $C^* \subseteq E_j$ or $\emptyset$ if no such cocircuit exists

1 $S \leftarrow \emptyset$
2 **foreach** $e \in E \setminus E_j$ **do**
3 **if** $S \cup \{e\} \in \mathcal{I}$ **then**
4 $S \leftarrow S \cup \{e\}$
5 $C^* \leftarrow \emptyset$
6 **foreach** $e \in E_j$ **do**
7 **if** $S \cup \{e\} \in \mathcal{I}$ **then**
8 $C^* \leftarrow C^* \cup \{e\}$

Result: C^*

Lemma 4.4. *Algorithm 4 correctly identifies a cocircuit $C^* \in \mathcal{C}^*(M)$ in Auction 2.*

Proof. By Lemma 4.2, a bidder j controls a cocircuit in a matroid M if and only if $r(E) > r(E \setminus E_j)$. In lines 1 to 4, Algorithm 4 identifies an inclusion-wise maximal independent set contained in $E \setminus E_j$, and therefore determines $r(E \setminus E_j)$. In lines 5 to 8, all elements $C^* = \{e \in E_j : r((E \setminus E_j) \cup \{e\}) > r(E \setminus E_j)\}$ are identified. If E_j does not contain a cocircuit, then $r(E \setminus E_j) = r(E)$, and therefore $C^* = \emptyset$. Otherwise, C^* is a cocircuit as per Lemma 4.3. □

4.2 Adaptations

We adhere to the following schema for each matroid class and analyze the worst-case time complexity for each operation.

- **Description:** Explains how the matroid is described throughout the auction process. This includes the compact description of the specific matroid class but may also involve auxiliary data structures for more efficient operations.
- **Dual Matroid:** This section explores whether the matroid class is closed under duality and, if so, how to derive the description of the dual matroid.

- **Cocircuit:** Explains the method for identifying a cocircuit. When the class is closed under duality, we consistently use the dual approach, as it is generally more efficient. However, as discussed in Section 4.1, there may be specific instances where the primal approach is preferred, even when the class is closed under duality.
- **Deletion:** Describes how matroid deletion is performed.
- **Contraction:** Describes the procedure for matroid contraction.

4.2.1 Uniform Matroid

Description

A compact description of the uniform matroid is the ground set E together with the integer k. By definition, a subset $I \subseteq E$ is independent if and only if $|I| \leq k$. We do not use any auxiliary data structures for describing the uniform matroid.

Dual Matroid

The class of uniform matroids is closed under duality.

Lemma 4.5. *Let M be a uniform matroid on the ground set E with the non-negative integer k. Then, M^* is the uniform matroid on ground set E and the non-negative integer $k^* = |E| - k$.*

Proof. Since bases are maximal independent sets, we have $\mathcal{B}(M) = \{B \subseteq E : |B| = k\}$. Consequently, the dual bases are $\mathcal{B}(M^*) = \{B \subseteq E : |B| = |E| - k\}$. This implies that the independent sets of M^* are $\mathcal{I}(M^*) = \{I \subseteq E : |I| \leq |E| - k\}$. Therefore, M^* is a uniform matroid with ground set E and integer $k^* = |E| - k$. □

Example 4.1. Let M be a uniform matroid with ground set $E = \{a, b, c, d\}$ and $k = 3$. Then, M^* is the uniform matroid with the same ground set E and $k^* = 1$.

Theorem 4.6. *The dual description of a uniform matroid is derived in $O(1)$ time.*

Proof. By Lemma 4.5, deriving the dual matroid description involves determining the size of the ground set E and performing a subtraction operation. Typically, the size of the set E is stored as metadata within the set data structure, allowing both operations to be executed in $O(1)$. □

Cocircuit

Since uniform matroids are closed under duality, we apply the dual approach.
To determine if a set E_j contains a cocircuit in M, we determine if E_j contains a circuit in M^*.

Lemma 4.7. *Let M be a uniform matroid with ground set E and non-negative integer k. Then, the circuits of M are precisely the subsets of E with size $k+1$.*

Proof. By definition, the subsets of E with size $k+1$ are dependent. Furthermore, all subsets of size less than or equal to k are independent. It follows that the minimal dependent sets, and therefore the circuits, are precisely the subsets of size $k+1$. □

Example 4.2. Let M be a uniform matroid with ground set $E = \{a, b, c, d\}$ and $k = 3$. A bidder j controls a circuit in M if and only if $E_j = \{a, b, c, d\}$.

Theorem 4.8. *Determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$ in a uniform matroid and identifying this cocircuit takes $O(1)$ time.*

Proof. By Theorem 4.6, the dual description is constructed in $O(1)$. Furthermore, according to Lemma 4.7, finding a circuit in a set S involves determining the size of S and a comparison operation, both requiring $O(1)$ time. Thus, by applying the dual approach, determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$, and identifying this cocircuit, is performed in $O(1)$ time. □

Deletion

Since we use the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. However, this means that the deletion operation in the auction corresponds to a contraction operation on our dual description. Lemma 4.9 shows how contraction can be performed on the dual matroid description.

Lemma 4.9. *Let M be a uniform matroid with ground set E and non-negative integer k. Let $e \in E$. Then, M/e is the uniform matroid with ground set $E \setminus \{e\}$ and non-negative integer $k' = \max(k-1, 0)$.*

Proof. Suppose e is a loop in M. This implies that $k = 0$ and, by Lemma 2.14, $M/e = M \setminus e$. By Lemma 4.11, this means that M/e is the uniform matroid with ground set $E \setminus \{e\}$ and integer $k' = 0$.

Otherwise, e is not a loop in M. This means that, by Lemma 2.14, $\mathcal{I}(M/e) = \{I \subseteq E \setminus \{e\} : I \cup \{e\} \in \mathcal{I}(M)\}$. By the definition of uniform matroids, $\mathcal{I}(M) = \{I \subseteq E : |I| \leq k\}$. Therefore, $\mathcal{I}(M/e) = \{I \subseteq E \setminus \{e\} : |I| \leq k-1\}$. □

Example 4.3. Let M be a uniform matroid with ground set $E = \{a, b, c, d\}$ and $k = 3$. Then, $M \setminus b$ is the uniform matroid with ground set $E \setminus \{b\} = \{a, c, d\}$ and $k' = 2$.

Theorem 4.10. *The deletion operation in Auction 2 for a uniform matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, deleting an element e in a matroid M corresponds to contracting e in the dual matroid M^*. According to Lemma 4.9, contraction in M^* involves removing e from the ground set E and a subtraction operation. Removing an element from the ground set has a worst-case time complexity of $O(|E|)$, assuming E is stored as a hash set, and subtraction is performed in $O(1)$ time. Thus, the deletion operation is performed in $O(|E|)$ time. □

Contraction

Using the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. Consequently, the contraction operation in the auction corresponds to a deletion operation in the dual description. Lemma 4.11 shows how deletion can be performed on the dual matroid description.

Lemma 4.11. *Let M be a uniform matroid with ground set E and non-negative integer k. Let $e \in E$. Then, $M \setminus e$ is the uniform matroid with ground set $E \setminus \{e\}$ and the same non-negative integer k.*

Proof. By the definition of uniform matroids, $\mathcal{I}(M) = \{I \subseteq E : |I| \leq k\}$, and by the definition of deletion, $\mathcal{I}(M \setminus e) = \{I \in \mathcal{I}(M) : e \notin I\}$. Thus, for uniform matroids, it follows that $\mathcal{I}(M \setminus e) = \{I \subseteq E \setminus \{e\} : |I| \leq k\}$. □

Example 4.4. Let M be a uniform matroid with ground set $E = \{a, b, c, d\}$ and $k = 3$. Then, $M \setminus b$ is the uniform matroid with ground set $E \setminus \{b\} = \{a, c, d\}$ and the same integer $k = 3$.

Theorem 4.12. *The contraction operation in Auction 2 for a uniform matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, contracting an element e in a matroid M corresponds to deleting e in the dual matroid M^*. According to Lemma 4.11, deletion in M^* means removing e from the ground set E. This operation has a worst-case time complexity of $O(|E|)$, assuming E is stored as a hash set. □

4.2.2 Partition Matroid

Description

A compact description of a partition matroid is a partition $P_1, P_2, \ldots, P_m$ of the ground set E and non-negative integers $k_1, k_2, \ldots, k_m$. A subset $S \subseteq E$ is independent if and only if $|S \cap P_i| \leq k_i$ for all $1 \leq i \leq m$. We do not use any auxiliary data structures for representing the partition matroid.

Dual Matroid

The class of partition matroids is closed under duality.

Lemma 4.13. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and the non-negative integers $k_1, k_2, \ldots, k_m$. Then, M^* is represented by the same partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1^* = |P_1| - k_1, k_2^* = |P_2| - k_2, \ldots, k_m^* = |P_m| - k_m$.*

Proof. By definition, a set $I \subseteq E$ is independent in M if and only if $|I \cap E_i| \leq k_i$ for each $1 \leq i \leq m$. Consider a subset $S \subseteq E$. For S to be independent in the dual matroid M^*, it must be included in a basis of M^*. So $S \in \mathcal{I}(M^*)$ if there exists $B \in \mathcal{B}(M^*)$ with $S \subseteq B$, and therefore, if $E \setminus B \subseteq E \setminus S$. Note that B is a basis of M^* if and only if $E \setminus B$ is a basis of M. Consequently, $S \in \mathcal{I}(M^*)$ if $E \setminus S$ contains a basis of M. Thus, $S \in \mathcal{I}(M^*)$ if $|(E \setminus S) \cap P_i| \geq k_i$ for all $1 \leq i \leq m$, and therefore, if $|S \cap P_i| \leq |P_i| - k_i$ for all $1 \leq i \leq m$. Hence, the description of M^* consists of the same partition $P_1, P_2, \ldots, P_m$ but with the rule that a set is independent if it contains no more than $|P_i| - k_i$ elements from each partition P_i. □

Example 4.5. Let M be a partition matroid on the ground set $E = \{a, b, c, d, e, f\}$ with the partition $P_1 = \{a, b\}, P_2 = \{c, d, e\}, P_3 = \{f\}$ and associated integers $k_1 = 2, k_2 = 2, k_3 = 1$. Then, M^* is the partition matroid with the same ground set and partition together with the associated integers $k_1^* = 0, k_2^* = 1, k_3^* = 0$.

Theorem 4.14. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1, k_2, \ldots, k_m$. The dual matroid description is derived in $O(m)$ time.*

Proof. According to Lemma 4.13, deriving the dual matroid description involves the following two constant time operations for each $1 \leq i \leq m$: determining the size of P_i and performing the subtraction $|P_i| - k_i$. Therefore, the overall time complexity is $O(m)$. □

Cocircuit

Since partition matroids are closed under duality, we apply the dual approach.
To determine if a set E_j contains a cocircuit in M, we determine if E_j contains a circuit in M^*.

Lemma 4.15. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1, k_2, \ldots, k_m$. Then, the circuits of M are sets $C \subseteq E$ with $|P_l \cap C| = k_l + 1$ for some $1 \leq l \leq m$ and $P_g \cap C = \emptyset$ for all $1 \leq g \leq m$ with $g \neq l$.*

Proof. Let $C \subseteq E$ with $|P_l \cap C| = k_l + 1$ for some $1 \leq l \leq m$ and $P_g \cap C = \emptyset$ for all $g \neq l$ with $1 \leq g \leq m$. Since $|P_l \cap C| = k_l + 1$ for some $1 \leq l \leq m$, C is dependent in M. Let $e \in C$. Then, $C \setminus \{e\}$ is independent because $|P_l \cap (C \setminus \{e\})| = k_l$ and $|P_g \cap (C \setminus \{e\})| = 0 \leq k_g$ for all $1 \leq g \leq m$ with $g \neq l$. Thus, C is a minimal dependent set in M and, consequently, a circuit of M. □

Example 4.6. Let M be a partition matroid with the partition $P_1 = \{a, b\}, P_2 = \{c, d, e\}, P_3 = \{f\}$ and associated integers $k_1 = 2, k_2 = 2, k_3 = 1$.
A bidder j controls a circuit in M if and only if $E_j = P_2 = \{c, d, e\}$.

Theorem 4.16. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1, k_2, \ldots, k_m$. Determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$ in the partition matroid and identifying this cocircuit takes $O(m)$ time.*

Proof. By Theorem 4.14, the dual description is constructed in $O(m)$. However, since the dual description is reused throughout the auction, the time for deriving the dual description is negligible. Furthermore, according to Lemma 4.15, finding a circuit in a set S involves the following operations for each $1 \leq i \leq m$: performing the intersection $S \cap P_i$, determining the size of this intersection, and a comparison

operation, all requiring $O(1)$ time. Thus, by applying the dual approach, determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$, and identifying this cocircuit, is performed in $O(m)$ time. □

Deletion

Since we use the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. However, this means that the deletion operation in the auction corresponds to a contraction operation on our dual description. Lemma 4.17 shows how contraction can be performed on the dual matroid description.

Lemma 4.17. *Let M be a partition matroid with the partition $P_1, \ldots, P_m$ of the ground set E and non-negative integers $k_1, \ldots, k_m$. Let $e \in P_l$ for some $1 \leq l \leq m$. Then, M/e is represented by the partition $P_1, \ldots, P_l \setminus \{e\}, \ldots, P_m$ of the ground set $E \setminus \{e\}$ and non-negative integers $k_1, \ldots, k'_l = \max(k_l - 1, 0), \ldots, k_m$.*

Proof. Let $e \in P_l$ for some $1 \leq l \leq m$.
Suppose e is a loop in M. This implies that $k_l = 0$ and, by Lemma 2.14, $M/e = M \backslash e$. By Lemma 4.19, this means that M/e is the partition matroid with ground set $E \backslash \{e\}$ and integers $k_1, \ldots, k'_l = 0, \ldots k_m$.
Otherwise, e is not a loop in M. By the definition of partition matroids, a set $I \subseteq E$ is independent in M if and only if $|I \cap P_i| \leq k_i$ for all $1 \leq i \leq m$. By Lemma 2.14, $\mathcal{I}(M/e) = \{I \subseteq E \setminus \{e\} : I \cup \{e\} \in \mathcal{I}(M)\}$. Therefore, a set S is independent in M/e if $S \subseteq E \setminus \{e\}$ with $|S \cap P_l| \leq k_l - 1$ and $|S \cap P_i| \leq k_i$ for all $1 \leq i \leq m$ with $i \neq l$. □

Example 4.7. Let M be a partition matroid with the partition $P_1 = \{a, b\}, P_2 = \{c, d, e\}, P_3 = \{f\}$ and associated integers $k_1 = 2, k_2 = 2, k_3 = 1$.
Then, M/d is the partition matroid with the partition $P_1 = \{a, b\}, P_2 \setminus \{d\} = \{c, e\}, P_3 = \{f\}$ and associated integers $k_1 = 2, k'_2 = 1, k_3 = 1$.

Theorem 4.18. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1, k_2, \ldots, k_m$. The deletion operation in Auction 2 for the partition matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, deleting an element e in a matroid M corresponds to contracting e in the dual matroid M^*. According to Lemma 4.17, contraction in M^* involves removing e from the P_i that contains e, and decrementing k_i

by 1. While the subtraction takes $O(1)$ time, removing the element has a worst-case time complexity of $O(|E|)$, assuming the partition is stored as a list of hash sets. Furthermore, identifying the P_i that contains e takes $O(m)$ time. Since $m \leq |E|$, deletion in the partition matroid is performed in $O(|E|)$ time. □

Contraction

Using the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. Consequently, the contraction operation in the auction corresponds to a deletion operation in our dual description. Lemma 4.19 shows how deletion can be performed on the dual matroid description.

Lemma 4.19. *Let M be a partition matroid with the partition $P_1, \ldots, P_m$ of the ground set E and non-negative integers $k_1, \ldots, k_m$. Let $e \in P_l$ for some $1 \leq l \leq m$. Then, $M \setminus e$ is represented by the partition $P_1, \ldots, P_l \setminus \{e\}, \ldots, P_m$ of the ground set $E \setminus \{e\}$ and the same non-negative integers $k_1, \ldots, k_m$.*

Proof. By the definition of partition matroids, a set $I \subseteq E$ is independent in M if and only if $|I \cap P_i| \leq k_i$ for all $1 \leq i \leq m$. Since $\mathcal{I}(M \setminus e) = \{I \in \mathcal{I}(M) : e \notin I\}$, a set S is independent in $M \setminus e$ if $S \subseteq E \setminus \{e\}$ and $|S \cap P_l| \leq k_l$ for all $1 \leq i \leq m$. □

Example 4.8. Let M be a partition matroid with the partition $P_1 = \{a, b\}$, $P_2 = \{c, d, e\}, P_3 = \{f\}$ and associated integers $k_1 = 2, k_2 = 2, k_3 = 1$.
Then, $M \setminus d$ is the partition matroid with the partition $P_1 = \{a, b\}, P_2 \setminus \{d\} = \{c, e\}, P_3 = \{f\}$ and the same integers $k_1 = 2, k_2 = 2, k_3 = 1$.

Theorem 4.20. *Let M be a partition matroid with the partition $P_1, P_2, \ldots, P_m$ and non-negative integers $k_1, k_2, \ldots, k_m$. The contraction operation in Auction 2 for the partition matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, contracting an element e in a matroid M corresponds to deleting e in the dual matroid M^*. According to Lemma 4.19, deletion in M^* involves removing e from the P_i that contains e, which has a worst-case time complexity of $O(|E|)$, assuming the partition is stored as a list of hash sets. Furthermore, identifying the P_i that contains e takes $O(m)$ time. Since $m \leq |E|$, deletion in the partition matroid is performed in $O(|E|)$ time. □

4.2.3 Planar Matroid

Description

A compact description of a planar matroid is an undirected planar graph $G = (V, E)$ where V is a set of vertices and E is a set of edges.
However, we represent the graph using an adjacency list rather than a set of edges to run the auction more efficiently.

Definition 4.21. An *adjacency list* for a graph $G = (V, E)$ is a list L of $|V|$ elements. Each element $L[v]$ is itself a collection of vertices that are adjacent to v, i.e., there exists an edge between v and each vertex in $L[v]$.

Example 4.9. Consider the undirected graph in Figure 4.1. The corresponding corresponding adjacency list is shown in Figure 4.2.

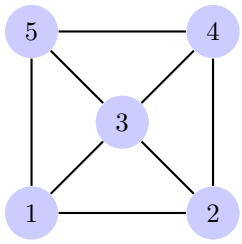


Figure 4.1: Undirected graph

Vertex	Adjacent Vertices
1	2, 3, 5
2	1, 3, 4
3	1, 2, 4, 5
4	2, 3, 5
5	1, 3, 4

Figure 4.2: Adjacency list

An adjacency list is more efficient than storing the graph as a list of edges because we use the Depth-First Search algorithm for finding cocircuits. An adjacency list facilitates exploring adjacent vertices, a central operation in Depth-First Search.

Definition 4.22. *Depth-First Search* is a graph traversal algorithm that starts at a selected node and explores as far as possible along each branch before backtracking. This algorithm uses a stack, either as a data structure or implicitly via recursion, to keep track of nodes to be visited. As it visits each node, it marks it as discovered, and upon exploring all adjacent nodes, it marks the node as finished and backtracks to explore new branches.

Lemma 4.23 (Adapted from p. 566 in [27])**.** *Let $G = (V, E)$ be an undirected graph represented by an adjacency list. The time complexity of performing Depth-First Search on G is $O(|E| + |V|)$.*

Dual Matroid

The class of planar matroids is closed under duality.

Definition 4.24. A *planar embedding* of a graph is a way of drawing the graph on a plane such that no two edges intersect except at their endpoints.

Definition 4.25. A *face of a planar graph* is a region of the planar embedding, bounded by edges and vertices. This includes both the bounded regions within the graph and the unbounded region surrounding it.

Definition 4.26. Given a planar graph G, its *dual graph* G^* is constructed by associating a vertex f^* in G^* with each face f of G. An edge connects two vertices f^* and g^* in G^* if and only if their corresponding faces f and g are separated by an edge in G.

Lemma 4.27 (Adapted from Lemma 2.3.7 in [17])**.** *Let M be a planar matroid associated with a planar graph G. Then, M^* is a planar matroid with graph G^*.*

Example 4.10. A planar graph, depicted in blue, and its corresponding dual graph, depicted in red, are illustrated in Figure 4.3.

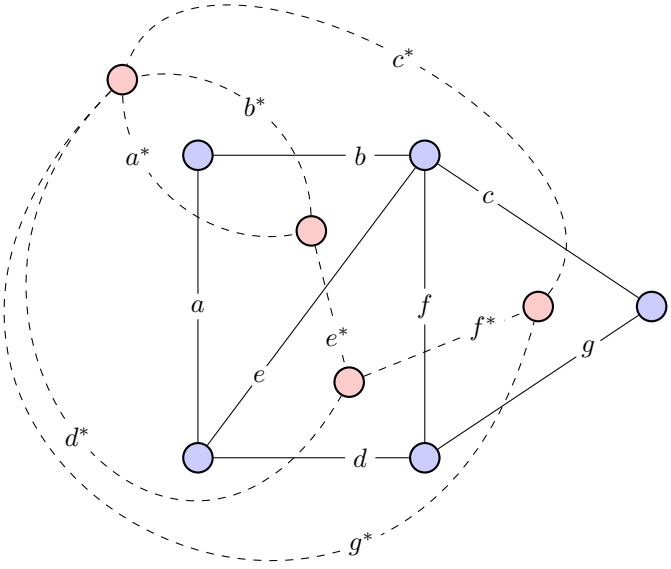


Figure 4.3: A planar graph in blue and its corresponding dual graph in red.

Theorem 4.28. *The dual matroid description of the planar matroid is derived in $O(|E| + |V|)$ time.*

Proof. The Boyer-Myrvold algorithm constructs a planar embedding in $O(|V|)$ [28]. Using Depth-First Search, the set of all faces in the planar embedding is identified in $O(|E| + |V|)$ according to Lemma 4.23. This is done by identifying cycles, which correspond to the boundaries of the faces. For each face in the planar embedding, a node in the dual graph is created. According to Euler's formula for planar graphs, $|V| - |E| + |F| = 2$, where F is the set of faces [29]. Therefore, $|F| \leq |E|$, and creating the new nodes takes $O(|E|)$ time. Furthermore, each edge is examined to determine the two faces it separates, requiring $O(|E|)$ time. Thus, the overall time complexity for deriving the dual graph is $O(|E| + |V|)$. By Lemma 4.27, this means that the dual description of a planar matroid is derived in $O(|E| + |V|)$ time. □

Cocircuit

Since planar matroids are closed under duality, we use the dual approach. To determine if E_j contains a cocircuit in the matroid M, we determine if E_j contains a circuit in the dual matroid M^*.

Lemma 4.29 (Follows from Proposition 1.1.7 in [17])**.** *Let M be a planar matroid associated with the graph $G = (V, E)$. The set of circuits $\mathcal{C}$ of M is exactly the set of edge cycles of G.*

Example 4.11. Consider the dual graph in Figure 4.3, depicted in red. The cycles and, thus, circuits of the corresponding planar matroid are: $\mathcal{C}(M^*) = \{\{a^*, b^*\}, \{a^*, c^*, f^*, e^*\}, \{b^*, c^*, f^*, e^*\}, \{a^*, d^*, e^*\}, \{b^*, d^*, e^*\}, \{a^*, g^*, f^*, e^*\}, \{b^*, g^*, f^*, e^*\}, \{d^*, g^*, f^*\}, \{c^*, g^*\}, \{d^*, f^*, c^*\}\}$. We can observe that these sets of edges correspond to minimal cuts in the original graph, depicted in blue.

Theorem 4.30. *Determining whether a bidder j controls a cocircuit $C^* \in \mathcal{C}^*$ in a planar matroid and identifying this cocircuit takes $O(|E_j|)$ time.*

Proof. By Theorem 4.28, the dual description of a graph is constructed in $O(|E| + |V|)$. Since the dual description is reused throughout the auction, the time for deriving the dual is amortized and becomes negligible. Moreover, according to Lemma 4.29, determining whether a set E_j contains a cocircuit in M and identifying this cocircuit corresponds to determining if the edges E_j contain a cycle in the dual graph and identifying this cycle.
To check for a cycle in the edges E_j, a Depth-First Search is executed on the subgraph of the dual graph induced by E_j. During this search, vertices are visited and

marked. If a vertex is revisited, a cycle is detected. This cycle is then identified by backtracking from the point of detection to the initial visit of the revisited vertex. According to Lemma 4.23, this process takes $O(|V_j^*| + |E_j|)$ time, where $|V_j^*|$ is the set of vertices in the subgraph of the dual graph induced by the edges E_j. By Definition 4.26, $|F_j| = |V_j^*|$, where F_j is the set of faces in the subgraph of the original graph induced by the edges E_j. According to Euler's formula for planar graphs, $|F_j| \leq |E_j|$ [29]. Hence, the overall time complexity for determining whether a bidder j controls a cocircuit $C^* \in \mathcal{C}^*$, and identifying this cocircuit, is $O(|E_j|)$. □

Deletion

Since we use the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. However, this means that the deletion operation in the auction corresponds to a contraction operation on our dual description. Lemma 4.33 shows how contraction can be performed on the dual matroid description.

Definition 4.31. Let $G = (V, E)$ be an undirected graph and $e = \{v_1, v_2\} \in E$ be an edge. The *edge contraction* of e in G, denoted by G/e, involves the following steps:

1. Remove the edge e from G.
2. Merge the vertices v_1 and v_2 into a single vertex v.
3. Reroute all edges that were incident to v_1 or v_2 to the new vertex v.

Lemma 4.32 (Adapted from Fact 3.2.1 in [17])**.** *Let M be a graphic matroid associated with the undirected graph $G = (V, E)$. Let $e \in E$. Then, M/e is the graphic matroid associated with the graph G/e.*

Lemma 4.33. *Let M be a planar matroid associated with the undirected graph $G = (V, E)$. Let $e \in E$. Then, M/e is the planar matroid associated with the graph G/e.*

Proof. This result follows directly from Lemma 4.32, given that planar matroids are graphic matroids and that planar graphs remain planar upon the contraction of any edge. □

Example 4.12. A planar graph with its adjacency list before and after contracting the edge $e = \{2, 3\}$. The original graph and its adjacency list are depicted in Figure 4.4 and Figure 4.5, respectively. The resulting graph and its adjacency list after edge contraction are shown in Figure 4.6 and Figure 4.7.

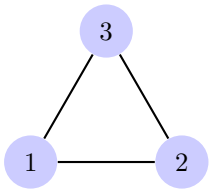


Figure 4.4: Original graph

Vertex	Adjacent Vertices
1	2, 3
2	1, 3
3	1, 2

Figure 4.5: Adjacency list of the original graph

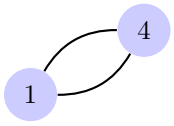


Figure 4.6: Graph after edge contraction

Vertex	Adjacent Vertices
1	4, 4
4	1, 1

Figure 4.7: Adjacency list after edge contraction

Theorem 4.34. *The deletion operation in Auction 2 for the planar matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, deleting an element e in a matroid M corresponds to contracting e in the dual matroid M^*. According to Lemma 4.33, contraction in M^* is equivalent to edge contraction in G^*. Suppose the edge e has the form $e = \{v_1, v_2\}$. Edge contraction of e involves finding and removing e from the adjacency lists of v_1 and v_2. The time complexity of this operation is $O(\deg(v_1)+\deg(v_2))$ because we need to traverse the lists of both vertices to find and remove e. Merging the vertices v_1 and v_2 into a new vertex v involves iterating through the adjacency lists of v_1 and v_2 and updating all connections to point to the new vertex v. This update also has a time complexity of $O(\deg(v_1) + \deg(v_2))$ since each element in the lists of v_1 and v_2 must be updated. Note that $\deg(v_1) + \deg(v_2) \leq 2|V^*|$, where V^* are the vertices of G^*. By Lemma 4.26, $|V^*| = |F|$, where F are the faces of G. According to Euler's formula for planar graphs, $|V| - |E| + |F| = 2$ [29]. Therefore, $|F| \leq |E|$. Consequently, the total time complexity of the edge contraction in G^*, and therefore deletion in the planar matroid, is $O(|E|)$. □

Contraction

Using the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. Consequently, the contraction operation in the auction corresponds to a deletion operation in our dual description. Lemma 4.35 shows how deletion can be performed on the dual matroid description.

Lemma 4.35 (Follows from Fact 3.1.2 in [17])**.** *Let M be a planar matroid associated with the planar, undirected graph $G = (V, E)$. Let $e \in E$. Then, $M \setminus e$ is the planar matroid associated with the graph $G \setminus e = (V, E \setminus \{e\})$.*

Theorem 4.36. *The contraction operation in Auction 2 for the planar matroid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, contracting an element e in a matroid M corresponds to deleting e in the dual matroid M^*. According to Lemma 4.35, deletion in M^* is simply the edge deletion of an edge e, which involves finding and removing e from the adjacency lists of its endpoints v_1 and v_2. The time complexity of this operation is $O(\deg(v_1) + \deg(v_2))$ because we need to traverse the lists of both vertices to find and remove e. Note that $\deg(v_1) + \deg(v_2) \leq 2|V^*|$, where V^* are the vertices of G^*. By Lemma 4.26, $|V^*| = |F|$, where F are the faces of G. According to Euler's formula for planar graphs, $|V| - |E| + |F| = 2$ [29]. Therefore, $|F| \leq |E|$. Consequently, the total time complexity of the edge deletion in G^*, and therefore contraction in the planar matroid, is $O(|E|)$. □

Example 4.13. Consider the undirected graph and its adjacency list depicted in Figures 4.4 and 4.5. We delete the edge $e = \{2, 3\}$. The resulting graph and its adjacency list are shown in Figure 4.8 and Figure 4.9.

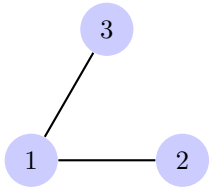


Figure 4.8: Graph after edge deletion

Vertex	Adjacent Vertices
1	2, 3
2	1
3	1

Figure 4.9: Adjacency list after edge deletion

4.2.4 Graphic Matroid

Description

A compact description of a graphic matroid is an undirected graph $G = (V, E)$ where V is a set of vertices and E is a set of edges.

However, we include another data structure in the description of the graphic matroid, known as Union-Find. Union-Find can be used to track the disjoint sets of vertices in a graph that are not interconnected by edges. Thus, Union-Find efficiently tracks the disconnected components of the graph. We use Union-Find to keep track of edge contractions without redirecting edges in the graph directly. Furthermore, we use it to find cocircuits efficiently.

Definition 4.37. *Union-Find* is designed to manage multiple disjoint sets efficiently. In the data structure, every element is part of a set, and each set is represented as a tree. In this tree, each node is an element, and the root of the tree acts as the representative of the set. Union-Find revolves around two main operations:
Find: This operation takes an element as input and traces the parent nodes in the tree until it reaches the root. The root is the representative of the set to which the element belongs and clearly identifies the set.
Union: This operation takes two elements from different sets and merges both sets into one by making one root element the child of the other root element. The root of the new tree becomes the representative of the newly formed set.

Example 4.14. Figure 4.10 shows a Union-Find structure with representatives 3 and 4. Figure 4.11 illustrates the same structure after performing a union operation between an element in $\{1, 2, 3, 6\}$ and an element in $\{4, 5\}$.

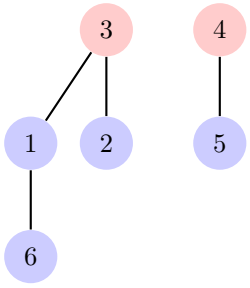


Figure 4.10: Union-Find structure before a union operation

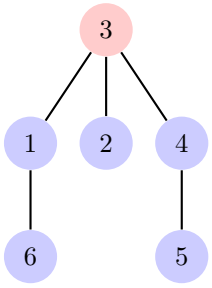


Figure 4.11: Union-Find structure after a union operation

Definition 4.38. A *Union-Find description* of a graphic matroid is a description that consists of a set of edges E and a Union-Find structure with elements for each vertex in the graph. Applying union to an edge $e \in E$ means applying union to the vertices v_1, v_2 with $e = \{v_1, v_2\}$. Initially, all elements in this Union-Find structure are representatives. However, this changes when contractions are applied, as explained in 4.45.

Lemma 4.39. *Let M be a graphic matroid with a set of edges E and a newly created Union-Find structure according to Definition 4.38. A set $I \subseteq E$ is independent if and only if applying union operations to all edges $e \in I$ always merges two distinct sets in the Union-Find structure.*

Proof. ($\Rightarrow$) Suppose I is an independent set in the matroid M, meaning that I forms an acyclic subgraph in the associated graph. In a new Union-Find structure, each vertex is its own representative. Let $e = \{v_1, v_2\} \in I$. The vertices v_1 and v_2 must belong to separate sets in the Union-Find structure, even after applying union to all other $f \in I$ with $f \neq e$ because their only connection is via the edge e since there are no cycles. Therefore, the union operation for each edge $e \in I$ merges the sets containing v_1 and v_2.
($\Leftarrow$) Assume that applying the union operations to all edges $e \in I$ leads to the merging of two distinct sets each time. This indicates that before each union operation, no pair of vertices v_1 and v_2 in any edge $e = \{v_1, v_2\}$ from I are in the same set, implying that there are no alternate paths connecting v_1 and v_2 within I. Consequently, the set I forms an acyclic subgraph and is therefore independent in M. □

Dual Matroid

Lemma 4.40 (Adapted from Proposition 2.3.3 in [17])**.** *The class of graphic matroids is not closed under duality.*

Cocircuit

Since the class of graphic matroids is not closed under duality, we use the primal approach, which revolves around rank calculations. Similar to planar matroids, cocircuits in a graphic matroid intuitively correspond to minimal cuts in the associated undirected graph [17]. We show how these cocircuits can be found computationally,

using a representation that allows for very efficient contractions, as detailed in Section 4.2.4.

Lemma 4.41 (Adapted Proposition 1.3.6 in [17])**.** *Let M be a graphic matroid associated with the undirected graph $G = (V, E)$ and $S \subseteq E$. Then, the rank is given by $r(S) = |V| - \kappa(S)$, where $\kappa(S)$ is the number of disconnected components in the subgraph G induced by the edges S.*

Theorem 4.42. *Determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$ in the graphic matroid and identifying this cocircuit takes $O(|E| \cdot \alpha(|V|))$ time.*

Proof. We use the primal approach, which is based on calculating the matroid rank. By Lemma 4.41, the rank of a graphic matroid relates to the number of connected components in the graph. Therefore, the Union-Find structure is well-suited for finding cocircuits. To determine if a bidder j controls a cocircuit, we first calculate $r(E \setminus E_j)$. This is done by evaluating the number of disconnected components κ in the subgraph induced by $E \setminus E_j$. The Union-Find structure achieves this by performing a union operation for each pair of vertices connected by an edge in $E \setminus E_j$, ensuring vertices connected by any such edge share the same representative. The number of different representatives then indicates the number of disconnected components.
Next, we check the edges $e \in E_j$ and determine whether v_1 and v_2 with $e = \{v_1, v_2\}$ belong to the same component by checking if they have the same representative. If they do not, uniting v_1 and v_2 would decrease κ and increase the rank, indicating that bidder j controls a cocircuit according to Lemma 4.2. By Lemma 4.3, all edges e where the associated vertices do not have the same representative are part of the cocircuit.
Using a Union-Find structure with the optimizations of union by rank and path compression, each union and find operation has a worst-case time complexity of $O(\alpha(|V|))$, where α is the inverse Ackermann function [27]. Since we perform $|E|$ of these operations overall, identifying a cocircuit in the graphic matroid takes $O(|E| \cdot \alpha(|V|))$ time. □

Example 4.15. Consider the graphic matroid M associated with the undirected graph in Figure 4.12 and suppose a bidder j is interested in the edges $E_j = \{\{1,2\},\{2,3\},\{2,4\},\{3,5\}\}$. Intuitively, we can already observe that the edges $C^* = \{\{1,2\},\{2,3\},\{2,4\}\}$ form a minimal cut in the graph and are therefore a cocircuit. However, we will now demonstrate the computational approach using Union-Find structures.

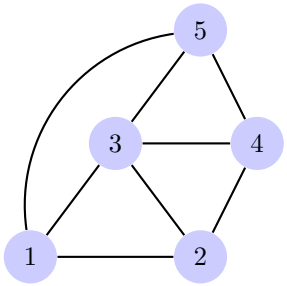


Figure 4.12: Undirected graph with $V = \{1,2,3,4,5\}$.

By Definition 4.38, the initial description of M for the auction consists of the set of edges $E = \{\{1,2\},\{1,5\},\{1,3\},\{2,3\},\{2,4\},\{3,4\},\{3,5\},\{4,5\}\}$ and the Union-Find structure in Figure 4.13.



Figure 4.13: Union-Find structure with each element acting as its representative.

We determine if E_j contains a cocircuit by applying the union operation to all vertices belonging to the edges $E \setminus E_j = \{\{1,5\},\{1,3\},\{3,4\},\{4,5\}\}$. This yields the Union-Find structure depicted in Figure 4.14.

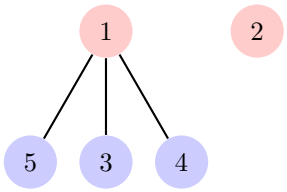


Figure 4.14: Union-Find structure in Figure 4.13 after applying the union operation to all vertices belonging to the edges $E \setminus E_j$.

Now, we determine for all $e \in E_j$ whether or not the two vertices connected by e share the same representative. All edges $e \in E_j$ where the associated vertices do not share a representative are part of the cocircuit. Therefore $C^* = \{\{1,2\},\{2,3\},\{2,4\}\}$.

Deletion

We use the primal approach. Unlike the dual approach, this means the deletion operation in the auction directly corresponds to a deletion operation on the graphic matroid description.

Lemma 4.43. *Let M be a graphic matroid with a set of edges E and a Union-Find structure according to Definition 4.38. Let $e \in E$. Then, $M \setminus e$ is the graphic matroid with the set of edges $E \setminus \{e\}$ and the same Union-Find structure.*

Proof. By Lemma 4.39, a set $I \subseteq E$ is independent if applying union operations to all edges $f \in I$ always merges two distinct sets in the Union-Find structure. By the definition of deletion, $\mathcal{I}(M \setminus e) = \{I \subseteq E \setminus \{e\} : I \in \mathcal{I}(M)\}$. Therefore, $I' \in \mathcal{I}(M \setminus e)$ if $I' \subseteq E \setminus \{e\}$ and if applying the union operations to all edges $f' \in I'$ always merges two distinct sets in the original Union-Find structure. Consequently, $M \setminus e$ is the graphic matroid with the set of edges $E \setminus \{e\}$ and the same Union-Find structure. □

Theorem 4.44. *The deletion operation in Auction 2 for the graphic matroid is performed in $O(|E|)$ time.*

Proof. The graphic matroid is represented by a set of edges and a Union-Find structure. By Lemma 4.43, graphic matroid deletion of an element e involves deleting the edge e in the set of edges E. This operation has a worst-case time complexity of $O(|E|)$, assuming that E is stored as a hash set. □

Example 4.16. Consider the graphic matroid M associated with the edges $E = \{\{1,2\},\{1,3\},\{2,3\}\}$ and some Union-Find structure. Then, $M \setminus \{1,3\}$ is the graphic matroid associated with the edges $E \setminus \{\{1,3\}\} = \{\{1,2\},\{2,3\}\}$ and the same Union-Find structure.

Contraction

We use the primal approach, so the contraction operation in the auction directly translates to a contraction operation on the graphic matroid description.

Lemma 4.45. *Let M be a graphic matroid with a set of edges E and a Union-Find structure according to Definition 4.38. Let $e = \{v_1, v_2\} \in E$. Then, M/e is the graphic matroid with the set of edges $E \setminus \{e\}$ and the same Union-Find structure, but with a union operation applied to the elements v_1 and v_2 if $v_1 \neq v_2$.*

Proof. Suppose e is a loop in M. This implies that $v_1 = v_2$ and, by Lemma 2.14, $M/e = M \setminus e$. By Lemma 4.43, this means that M/e is the matroid with the set of edges $E \setminus \{e\}$ and the same Union-Find structure.
Otherwise, e is not a loop in M. By Lemma 4.39, a set $I \subseteq E$ is independent if applying union operations to all edges $f \in I$ always results in merging two distinct sets in the Union-Find structure. By Lemma 2.14, $\mathcal{I}(M/e) = \{I \subseteq E \setminus \{e\} : I \cup \{e\} \in \mathcal{I}(M)\}$.
Let $e \in E$ with $e = \{v_1, v_2\}$ and $S \subseteq E \setminus \{e\}$.
We transform the old Union-Find structure to a new Union-Find structure by applying the union operation to v_1 and v_2. We show that M/e is the graphic matroid associated with the set of edges $E \setminus \{e\}$ and the new Union-Find structure.
Case 1: Suppose $S \cup \{e\} \in \mathcal{I}(M)$. Then, applying the union operations to all edges $f \in S \cup \{e\}$ results in merging two distinct sets in the old Union-Find structure, and consequently, applying the union operations to all edges $f \in S$ always results in merging two distinct sets in the new Union-Find structure. Therefore, $S \in \mathcal{I}(M/e)$, where M/e is the graphic matroid associated with the set of edges $E \setminus \{e\}$ and the new Union-Find structure.
Case 1: Suppose $S \cup \{e\} \notin \mathcal{I}(M)$. Then, applying the union operations to all edges $f \in S \cup \{e\}$ does not result in merging two distinct sets in the old Union-Find structure, and consequently, applying the union operations to all edges $f \in S$ does not always result in merging two distinct sets in the new Union-Find structure. Therefore, $S \notin \mathcal{I}(M/e)$, where M/e is the graphic matroid associated with the set of edges $E \setminus \{e\}$ and the new Union-Find structure. □

Example 4.17. Consider the graphic matroid M with the edges
$E = \{\{1,2\},\{1,3\},\{2,3\}\}$ and the Union-Find structure in Figure 4.15.
$M/\{1,3\}$ is the graphic matroid with the edges $E \setminus \{\{1,3\}\} = \{\{1,2\},\{2,3\}\}$ and the Union-Find structure in Figure 4.16.



Figure 4.15: Union-Find structure before contraction

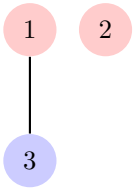


Figure 4.16: Union-Find structure after contraction

Theorem 4.46. *The contraction operation in Auction 2 for the graphic matroid is performed in $O(|E| + \alpha(|V|))$ time.*

Proof. By Lemma 4.45, the contraction of an edge $e = \{v_1, v_2\}$ in a graphic matroid is implemented by applying the union operation to the vertices v_1 and v_2 and removing e from the set of edges E. Assuming the union-find structure employs union by size and path compression, this union operation has a time complexity of $O(\alpha(|V|))$, where α is the inverse Ackermann function [27]. The operation of removing the edge e from E has a worst-case time complexity of $O(|E|)$. Therefore, the contraction operation in a graphic matroid is performed in $O(|E| + \alpha(|V|))$ time. □

4.2.5 Gammoid

Description

A compact description of a gammoid M is a directed graph $D = (V, \mathcal{A})$ where V is a set of vertices and $\mathcal{A}$ a set of directed edges, along with the set of starting vertices $E \subseteq V$, and the set of destination vertices $T \subseteq V$.

However, we transform the directed graph D into a flow network N to identify the independent sets of M. Remember that independent sets in a gammoid correspond to routings from the starting vertices to the destination vertices in D. Finding these routings involves finding families of vertex-disjoint paths in D, which can be reduced to finding maximum flows in flow networks.

Definition 4.47. A *flow network* N is a tuple $N = (D, c, s, t)$ where:

- $D = (V, \mathcal{A})$ is a directed graph with vertices V and directed edges $\mathcal{A}$.
- $c : \mathcal{A} \to \mathbb{Z}_0$ is a function that assigns a *capacity* to each edge.
- $s \in V$ is the *source* node.
- $t \in V$ is the *sink* node.

Definition 4.48. Let $N = (D, c, s, t)$ be a flow network with $D = (V, \mathcal{A})$. A *flow* f for N is a function $f : \mathcal{A} \to \mathbb{R}$. A flow f is called *feasible* if it satisfies the following two constraints:

- **Capacity Constraint:** The flow over any edge cannot exceed its capacity:

$$0 \leq f((u, v)) \leq c((u, v)) \text{ for all } (u, v) \in \mathcal{A}.$$

- **Flow Conservation:** For all $v \in V \setminus \{s, t\}$ the total flow entering v must equal the total flow exiting it:

$$\sum_{u:(u,v)\in\mathcal{A}} f((u,v)) = \sum_{w:(v,w)\in\mathcal{A}} f((v,w)).$$

Definition 4.49. The *value* of a flow f, denoted by $val(f)$, is the net flow out of the source:

$$\text{val}(f) \coloneqq \sum_{u:(s,u)\in\mathcal{A}} f((s,u)) - \sum_{v:(v,s)\in\mathcal{A}} f((v,s)).$$

Definition 4.50. The *maximum flow* $f^*(N)$ in a flow network N is a feasible flow with maximum possible value. We denote this maximum value by $val^*(N)$ with $val^*(N) \coloneqq val(f^*(N))$.

Definition 4.51. Let $N = (D, c, s, t)$ be a flow network with $D = (V, \mathcal{A})$ and flow f. The *residual network* of N with respect to f is a flow network $N^f = (D, c^f, s, t)$ where c^f is called the *residual capacity* and defined as

$$c^f((u,v)) \coloneqq c((u,v)) - f((u,v)) + f((v,u)) \text{ for all } (u,v) \in \mathcal{A}.$$

Definition 4.52. Let $N = (D, c, s, t)$ be a flow network with $D = (V, \mathcal{A})$ and flow f. An *augmenting path* in N with respect to f is a path from s to t in the residual network N^f such that every edge on the path has positive residual capacity. Flow *augmentation* refers to increasing the flow along an augmenting path by the minimum residual capacity of the edges on that path.

Method 5: Ford-Fulkerson Method

Input: flow network $N = (D, c, s, t)$ with $D = (V, \mathcal{A})$

Output: value of the maximum flow $val^*(N)$

1 $f((u,v)) \leftarrow 0$ for all $(u,v) \in \mathcal{A}$

2 **while** *there is an augmenting path p in N with respect to f* **do**

3 augment f along p

Result: $val(f)$

Lemma 4.53 (Follows from [30])**.** *Method 5 correctly identifies the value of the maximum flow in a flow network N.*

Definition 4.54. The variant of Method 5 that discovers the augmenting paths via Breadth-First Search is called the *Edmonds-Karp algorithm*.

Definition 4.55. Let M be a gammoid on a directed graph $D = (V, \mathcal{A})$ with starting vertices E and destination vertices T. The *gammoid flow network* is a flow network $N = (D_N, c, s, t)$ where $D_N = (V', \mathcal{A}')$ and:

- $V' = \{v_{in} : v \in V\} \cup \{v_{out} : v \in V\} \cup \{s\} \cup \{t\}$
- $\mathcal{A}' = \{(v_{in}, v_{out}) : v \in V\} \cup \{(u_{out}, v_{in}) : (u, v) \in \mathcal{A}\} \cup \{(s, v_{in}) : v \in E\} \cup \{(v_{out}, t) : v \in T\}$
- $c((u, v)) = 1$ for all $(u, v) \in \mathcal{A}'$

Example 4.18. Consider a gammoid M with starting vertices $E = \{1, 2\}$, destination vertices $T = \{3, 4\}$, and the associated directed graph D in Figure 4.17. The corresponding gammoid flow network N is depicted in Figure 4.18.

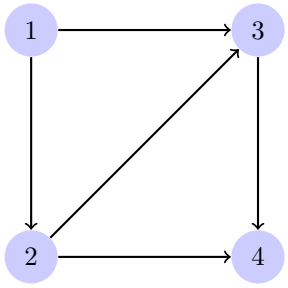


Figure 4.17: Directed graph D

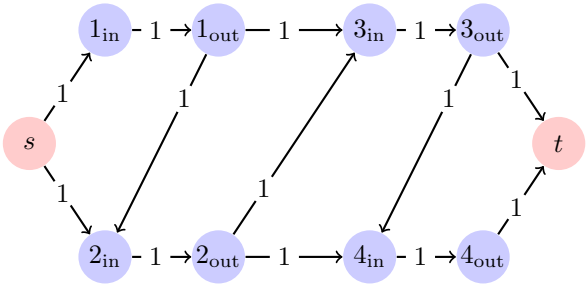


Figure 4.18: Gammoid flow network N

Definition 4.56. Let M be a gammoid with starting vertices E and a gammoid flow network $N = (D, c, s, t)$ with $D = (V, \mathcal{A})$. Let $S \subseteq E$. Then, $N_{|S}$ is the *restriction* $N_{|S} = (D', c, s, t)$ with $D' = (V, \mathcal{A}')$ and

$$\mathcal{A}' = \{(u, v) \in \mathcal{A} : (u \neq s \text{ or } v \neq w_{in}) \text{ and } (u \neq w_{in} \text{ or } v \neq s) \text{ for all } w \in E \setminus S\}.$$

Example 4.19. Let M be a gammoid with starting vertices $E = \{1, 2, 3\}$ and the gammoid flow network N in Figure 4.19. Figure 4.20 shows the flow network restriction $N_{|\{1\}}$.

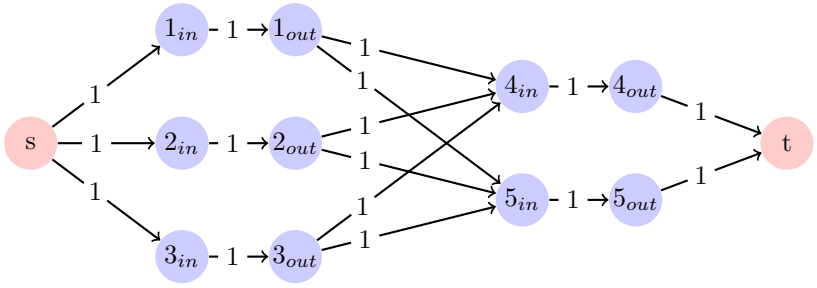


Figure 4.19: Flow network N

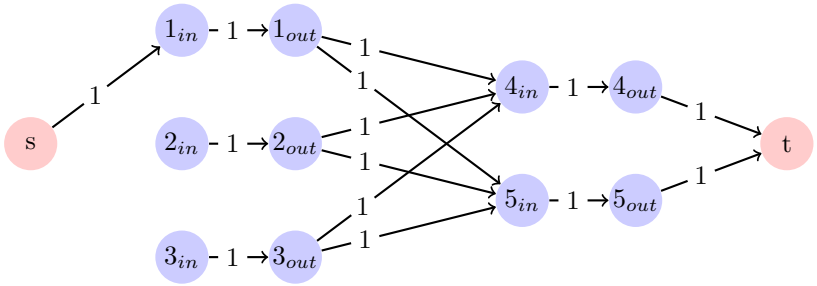


Figure 4.20: Flow network $N_{|\{1\}}$

Lemma 4.57. *Let M be a gammoid with starting vertices E and gammoid flow network N. Let $I \subseteq E$. Then, $I \in \mathcal{I}(M)$ if and only if $val^*(N_{|I}) = |I|$.*

Proof. Let M be the gammoid with starting E, destination vertices T, and the directed graph $D = (V, \mathcal{A})$. Let N be the associated gammoid flow network of M.
($\Rightarrow$) Suppose that $I \in \mathcal{I}(M)$. By the definition of gammoids, there is a routing from I to T in D. Therefore, there is a family of distinct vertex-disjoint paths connecting each $e \in I$ to a different vertex $g \in T$ in the directed graph D. Let f be a new flow with $val(f) = 0$ in N. By Definition 4.55, each path of the form $(v^1, v^2, \ldots, v^m)$ in D corresponds to a path $(s, v^1_{in}, v^1_{out}, v^2_{in}, v^2_{out}, \ldots, v^m_{in}, v^m_{out}, t)$ in N^f. Since the paths in D are vertex-disjoint, the paths in N^f are edge-disjoint. Since each edge has a capacity of 1, augmenting f along the $|I|$ different paths in N^f results in a feasible flow with $val(f) = |I|$. Since the vertex-disjoint paths all originate from vertices $e \in I$, it hold that $f((s, e_{in})) = 1$ for all $e \in I$ and $f((s, g_{in})) = 0$ for all $g \in E \setminus I$. Therefore, we can restrict the flow network N to $N_{|I}$ without affecting the flow f. Consequently, $val^*(N_{|I}) = |I|$.
($\Leftarrow$) Suppose that $val^*(N_{|I}) = |I|$. Let f^* be a maximum flow for $N_{|I}$. By Definition 4.55, each augmented path of the form $(s, v^1_{in}, v^1_{out}, v^2_{in}, v^2_{out}, \ldots, v^m_{in}, v^m_{out}, t)$ in $N^{f^*}_{|I}$ corresponds to a path of the form $(v^1, v^2, \ldots, v^m)$ in D with $v^1 \in I$ and $v^m \in T$. Since for all $u \in V$, the flow between vertices u_{in} and u_{out} in $N_{|I}$ is limited by 1, the

corresponding paths in D are all vertex-disjoint. Therefore, the augmented paths with respect to f^* in $N_{|I}$ correspond to a family of distinct vertex-disjoint paths connecting each $e \in I$ to a different vertex $f \in T$ in the directed graph D. This means that there is a routing from I to T in D. Consequently, $I \in \mathcal{I}(M)$. □

Dual Matroid

Gammoids are closed under duality, and to derive the dual, we use the method outlined by Albrecht [19]. This method revolves around the concept of duality-respecting descriptions.

Definition 4.58. The *opposite graph* D^{opp} of a directed graph D is the graph obtained by reversing the direction of all edges in D.

Definition 4.59. Let M be a gammoid on a directed graph D with starting vertices E and destination vertices T. The description is *duality-respecting* if the dual matroid M^* has the description D^{opp} with the same starting vertices E and destination vertices $T^{opp} = E \setminus T$.

To derive the dual matroid, we transform the description of M into a duality-respecting description. Before it can be converted to a duality-respecting description, however, it needs to be transformed into a basis description.

Definition 4.60. Let M be a gammoid on a directed graph D with starting vertices E and destination vertices T. This description of M is called a *basis description* if $T \in \mathcal{B}(M)$.

Definition 4.61. Let $D = (V, \mathcal{A})$ be a directed graph and let $(e, f) \in \mathcal{A}$. The *swap* of (e, f) in D returns the directed graph $D_{(e,f)} = (V, \mathcal{A}_{(e,f)})$ where

$$\mathcal{A}_{(e,f)} = \{(u, v) \in \mathcal{A} : u \neq e\} \cup \{(f, v) : v \neq f \text{ and } (e, v) \in \mathcal{A}\} \cup \{(f, e)\}.$$

Lemma 4.62. *Let M be a gammoid associated with a directed graph D with starting vertices E and destination vertices T. If $B \subseteq E$ is a basis of M, then there exist vertex-disjoint paths $p_1, \ldots, p_k$ from B to T in D. The identification of B and the paths $p_1, \ldots, p_k$ can be achieved by:*

1. *Converting D into a gammoid flow network N as described in Definition 4.55.*
2. *Identifying a maximum flow f^* in N using the Edmonds-Karp algorithm.*

3. Mapping each augmented path in N^{f^*} back to a vertex-disjoint path in D.

Proof. Suppose f^* is a maximum flow in N. Let B be the set with $f^*((s, v_{in})) = 1$ for all $v \in B$ and $f^*((s, u_{in})) = 0$ for all $u \in E \setminus B$. Therefore, N can be restricted to $N_{|B}$ without affecting the flow f^*. It holds that $val^*(N_{|B}) = |B|$, and by Lemma 4.57, $B \in \mathcal{I}(M)$. Since f^* is a maximum flow in $N = N_{|E}$, B is an inclusion-wise maximal independent set and therefore $B \in \mathcal{B}(M)$. Analogous to the proof for Lemma 4.57, each augmented path in $N_{|B}^{f^*}$ corresponds to a vertex-disjoint path from B to T in D. □

Lemma 4.63 (Follows from Corollary 4.1.2 in [31] and Remark 1.5 in [19])**.** *Let M be a gammoid with starting vertices E, destination vertices T, and basis B. Let $p_1, \ldots, p_k$ be the vertex-disjoint paths from B to T.*
Every gammoid description can be transformed to a basis description by applying the swap operation to all edges along the paths $p_1, \ldots, p_k$ in the reverse order of traversal.

Lemma 4.64 (Adapted from Lemma 2.2 in [19])**.** *Let M be a gammoid with a basis description, so with a directed graph $D = (V, \mathcal{A})$, starting vertices E, and destination vertices $T = B$ with $B \in \mathcal{B}(M)$.*
Then, the duality-respecting description consists of the same starting and destination vertices and the directed graph $D_{dr} = (V_{dr}, \mathcal{A}_{dr})$ with

$$V_{dr} = \{v' \mid v \in V\} \cup \{v \mid v \in T \cup E\}.$$

and

$$\mathcal{A}_{dr} = \{(u', v') \mid (u, v) \in \mathcal{A}\} \cup \{(b', b) \mid b \in T\} \cup \{(e, e') \mid e \in E \setminus T\}.$$

Example 4.20. Consider the gammoid M with starting vertices $E = \{1, 2, 3\}$, destination vertices $T = \{4, 5\}$, and the directed graph shown in Figure 4.21. Observe that this is equivalent to the uniform matroid on the same ground set E and $k = 2$.

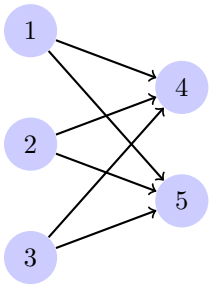
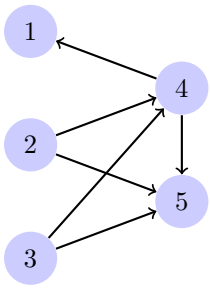
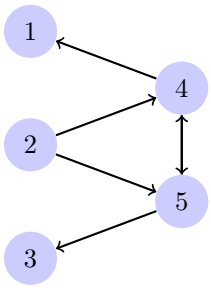


Figure 4.21: Directed graph with vertices $V = \{1, 2, 3, 4, 5\}$.

Suppose that after applying Lemma 4.62, the identified basis is $B = \{1, 3\}$ with single-edge vertex-disjoint paths $p_1 = (1, 4)$ and $p_2 = (3, 5)$. Figure 4.22 shows the conversion to a basis description. This basis description of M is given by starting vertices $E = \{1, 2, 3\}$, destination vertices $T_2 = B = \{1, 3\}$, and the directing graph in Figure 4.22b.



(a) Conversion to $E = \{1, 2, 3\}$ and $T_1 = \{1, 5\}$ by applying the swap operation to p_1.



(b) Conversion to $E = \{1, 2, 3\}$ and $T_2 = \{1, 3\}$ by applying the swap operation to p_2.

Figure 4.22: Conversion to a basis description by applying swap operations according to Definition 4.61.

Applying Lemma 4.64 to the basis description yields the duality-respecting description with starting vertices $E = \{1, 2, 3\}$, destination vertices $T_2 = \{1, 3\}$, and the directed graph shown in Figure 4.23.

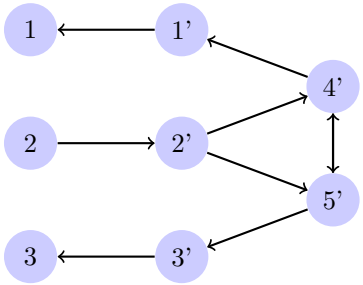


Figure 4.23: Associated directed graph of the duality-respecting description.

By Definition 4.59, the dual matroid M^* is associated with the starting vertices $E = \{1, 2, 3\}$, the destination vertices $T_{opp} = E \setminus T_2 = \{2\}$, and the directed graph shown in Figure 4.24.

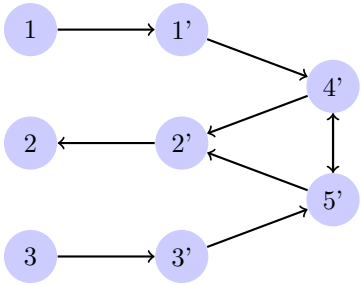


Figure 4.24: Associated directed graph of the dual description.

Theorem 4.65. *Let M be a gammoid with starting vertices E, destination vertices T, and the directed graph $D = (V, \mathcal{A})$. The description of M^* is derived in $O(|V|^2 + |\mathcal{A}|^2)$ time.*

Proof. Transforming the directed graph D into a flow network N for finding vertex-disjoint paths requires creating two new vertices and a new edge for each vertex $v \in V$ and redirecting the edges $(v_1, v_2) \in \mathcal{A}$, which takes $O(|V|+|\mathcal{A}|)$ time. Let $N = (D_n, c, s, t)$ with $D_n = (V_n, \mathcal{A}_n)$. Finding a basis B and the corresponding basis paths involves finding a maximum flow f^* in N and mapping the augmented paths in N^{f^*} back to vertex-disjoint paths in D. Since all capacities are 1, finding the maximum flow takes $O(|V_n| \cdot |\mathcal{A}_n|)$ time using the Edmonds-Karp algorithm [32]. Mapping the augmented paths in N^{f^*} back to vertex-disjoint paths in D requires $O(|\mathcal{A}_n|)$ time. Appling the swap operation to all edges in the basis paths requires $O(|\mathcal{A}|^2)$ time, as all paths together can contain up to $|\mathcal{A}|$ edges, and each swap operation takes $O(|\mathcal{A}|)$ time. Transforming the basis description into a duality-respecting description involves copying D and adding new sinks and vertices, taking $O(|V|+|\mathcal{A}|)$ time. The

process of reversing the edges incurs an additional $O(|\mathcal{A}|)$ time. By Definition 4.55, $|V_n| = 2|V| + 2$ and $|\mathcal{A}_n| = |V| + |\mathcal{A}|$. Consequently, deriving the dual matroid description of a gammoid takes $O(|V_n| \cdot |\mathcal{A}_n| + |\mathcal{A}|^2) = O((2|V|+2) \cdot (|V| + |\mathcal{A}|) + |\mathcal{A}|^2)$ time, which simplifies to $O(|V|^2 + |\mathcal{A}|^2)$ time. □

We analyze the time complexity of all operations in terms of the vertices and edges of the original description of M, even when working on a gammoid flow network associated with the dual M^*. Therefore, we establish Lemma 4.66 to translate the time complexities between these descriptions.

Lemma 4.66. *Let M be a gammoid on a directed graph $D = (V, \mathcal{A})$ with starting vertices E and destination vertices T. Let $N = (D_n, c, s, t)$ with $D_n = (V_n, \mathcal{A}_n)$ be the gammoid flow network associated with the dual M^*. Then, $|V_n| \leq 4|V| + 2$ and $|\mathcal{A}_n| \leq 3|V| + |\mathcal{A}|$.*

Proof. Let $D_d = (V_d, \mathcal{A}_d)$ be the graph associated with the dual M^*. Then, transforming D to D_d involves adding additional vertices for all $e \in E$ and all $f \in T$ to ensure that all starting vertices are vertices with only outgoing and all destination vertices are vertices with only incoming edges. Therefore, $|V_d| \leq |V| + |E| + |T| \leq 2|V|$ and $|\mathcal{A}_d| \leq |\mathcal{A}| + |E| + |T| \leq |\mathcal{A}| + |V|$. By Definition 4.55, $|V_n| = 2|V_d| + 2 \leq 4|V| + 2$ and $|\mathcal{A}_n| = |V_d| + |\mathcal{A}_d| \leq 2|V| + |\mathcal{A}| + |V| = 3|V| + |\mathcal{A}|$. □

Cocircuit

Since gammoids are closed under duality, we apply the dual approach. Thus, we determine if a bidder j controls a cocircuit in M by checking if E_j contains a circuit in the dual matroid M^*. Lemma 4.67 allows us to determine if a bidder j controls a circuit, while Lemma 4.68 enables us to identify this circuit.

Lemma 4.67. *Let M be a gammoid with starting vertices E and gammoid flow network N. A set E_j contains a circuit if and only if $val^*(N_{|E_j|}) < |E_j|$.*

Proof. By Lemma 4.57, it holds that $I \subseteq E$ is independent in M if and only if $val^*(N_{|I}) = |I|$. Note that $val^*(N_{|I}) > |I|$ is impossible, since there are only $|I|$ edges originating from s in $N_{|I}$ and each edge has a capacity of 1. Since a set E_j contains a circuit if and only if E_j is dependent in M, E_j contains a circuit if and only if $val^*(N_{|E_j|}) < |E_j|$. □

Lemma 4.68. *Let M be a gammoid with starting vertices E and gammoid flow network N. Suppose a set E_j contains a circuit $C \in \mathcal{C}(M)$. Let f^* be the maximum flow for $N_{|E_j}$ with $val(f^*) < |E_j|$ and let $e \in E_j$ be an element with $f^*((s, e_{in})) = 0$. Then,*

$$C = \{e\} \cup \{g \in E_j : \textit{there is a capacitated path } p = (s, e_{in}, \dots, g_{in}, s) \textit{ in } N^{f^*}_{|E_j}\}.$$

Proof. Since f^* is a maximum flow for $N_{|E_j}$ and $val(f^*) < |E_j|$, there exists an element $e \in E_j$ with $f^*((s, e_{in})) = 0$. By Lemma 3.1 applied to the dual, exactly one such element e exists.
Let $I = \{g \in E_j : \textit{there is a capacitated path } p = (s, e_{in}, \dots, g_{in}, s) \textit{ in } N^{f^*}_{|E_j}\}$.
Since $f^*((s, g_{in})) = 1$ for all $g \in I$, it holds that $val^*(N_{|I}) = |I|$. Therefore, $I \in \mathcal{I}(M)$.
Let $C = \{e\} \cup I$. Then, $f^*((s, g_{in})) = 1$ for all $g \in I$ and $f^*((s, e_{in})) = 0$. Suppose $C \in \mathcal{I}(M)$, so $val^*(N_{|C}) = |C|$. For $val^*(N_{|C}) = |C|$ to be true, it must be possible to change the flow f^* to a flow f' such that $f'((s, g_{in})) = 1$ for all $g \in C$ and $f'((s, h_{in})) = 0$ for another element $h \in E_j \setminus C$. Therefore, there must be an augmented path p_e in N^{f^*} beginning with (s, e_{in}) and an augmented path p_h in $N^{f'}$ starting with (s, h_{in}) such that p_e and p_h share an edge somewhere in N. Otherwise, p_e and p_h would not share the same bottleneck and the reversion of f^* along p_h would not allow for an augmentation along p_e. Then, however, there must be a capacitated path $p = (s, e_{in}, \dots, h_{in}, s)$ in $N^{f^*}_{|E_j}$. Consequently, $h \in C$, which is a contradiction. Therefore, $C \notin \mathcal{I}(M)$. Next, we show that C is an inclusion-wise minimal dependent set in M. Let $S = C \setminus \{g\}$ for $g \in C$.
Case 1: If $g = e$, then $C \setminus \{g\} = C \setminus \{e\} = I$. Since I is independent in M, $C \setminus \{g\}$ is also independent in M.
Case 2: Suppose $g \neq e$. Since $g \in I$, there is a capacitated path $p = (s, e_{in}, \dots, g_{in}, s)$ in $N^{f^*}_{|E_j}$. Increasing the flow f^* along p would mean that $f^*((s, e_{in})) = 1$ and $f^*((s, g_{in})) = 0$. Since $f^*((s, h_{in})) = 1$ for all $h \in I \setminus \{g\}$, it holds that $val^*(N_{|C\setminus\{g\}}) = |C \setminus \{g\}|$. Therefore, $C \setminus \{g\} \in \mathcal{I}(M)$. □

Example 4.21. Let M be the gammoid with starting vertices $E = \{1, 2, 3, 4\}$ and the associated gammoid flow network N in Figure 4.25.

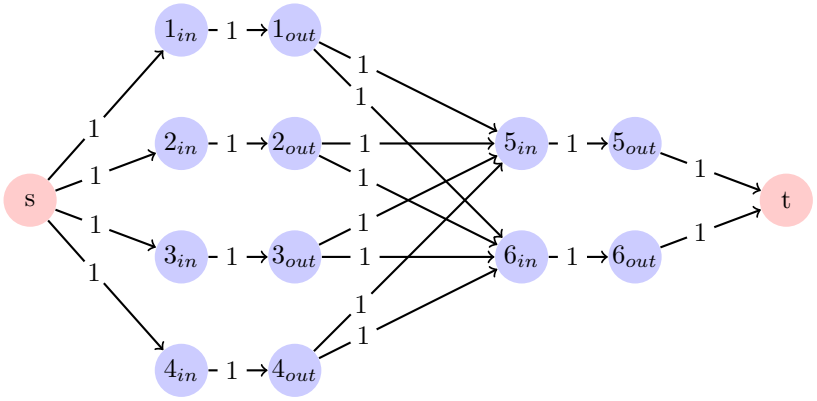


Figure 4.25: Flow network N

Suppose a bidder j is interested in the elements $E_j = \{1, 2, 3\}$. The restriction $N_{|E_j}$ is the flow network depicted in Figure 4.26.

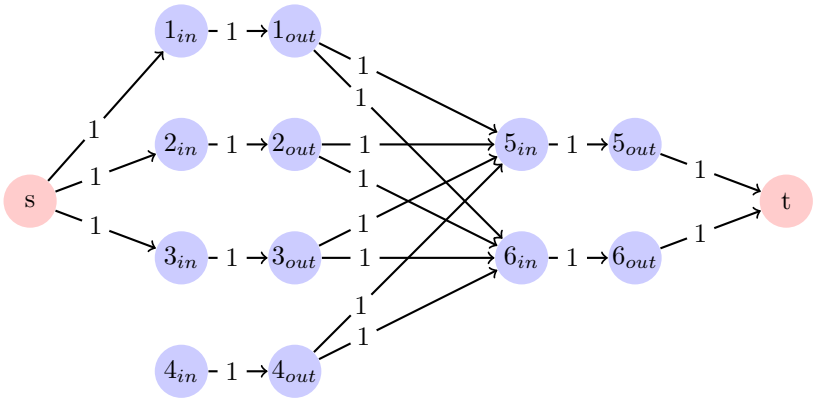


Figure 4.26: Flow network $N_{|E_j}$

To determine if bidder j controls a circuit, we identify a maximum flow f^* in $N_{|E_j}$ and determine if $val(f^*) < |E_j|$. For an initial flow f with $val(f) = 0$, the flow network and the residual network are identical. Therefore, Figure 4.26 also represents the residual network.
The identify the maximum flow, we apply the Edmonds-Karp algorithm. Therefore, we use Breath-First search to identify paths with available capacity in $N^f_{|E_j}$ and augment these paths. Depending on the exact paths that are found, the residual network may look like the network in Figure 4.27.

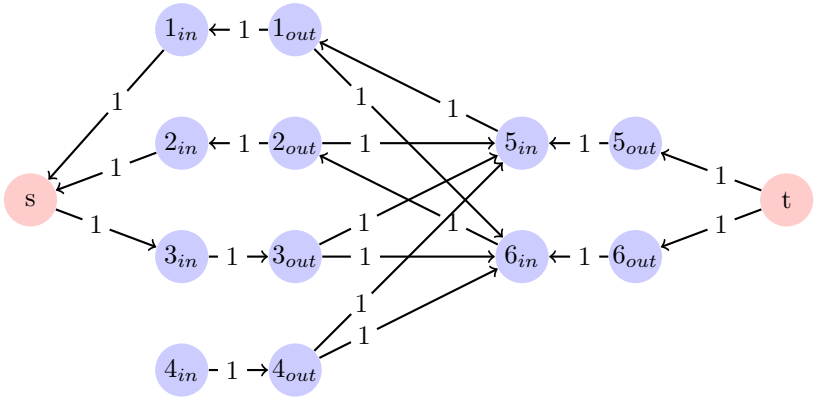


Figure 4.27: Residual network $N_{|E_j}$ after the execution of the Edmonds-Karp algorithm.

Since $val(f^*) = 2 < 3 = |E_j|$, the set E_j contains a circuit according to Lemma 4.67. Furthermore, $3 \in E_j$ is an element with $f^*((s, 3_{in})) = 0$. Since there exist paths $p_1 = (s, 3_{in}, 3_{out}, 5_{in}, 1_{out}, 1_{in}, s)$ and $p_2 = (s, 3_{in}, 3_{out}, 6_{in}, 2_{out}, 2_{in}, s)$, the circuit is $C = \{3\} \cup \{1, 2\} = \{1, 2, 3\}$ according to Lemma 4.68.

Theorem 4.69. *Let M be a gammoid with starting vertices E, destination vertices T, and the directed graph $D = (V, \mathcal{A})$. Determining whether a bidder j controls a cocircuit $C^* \in \mathcal{C}^*$ in M, and identifying this cocircuit, takes $O(|E_j| \cdot (|V| + |\mathcal{A}|))$ time.*

Proof. Let N be the associated gammoid flow network of the dual matroid M^* with $N = (D_n, c, s, t)$ and $D_n = (V_n, \mathcal{A}_n)$. By Lemma 4.66, $|V_n| \leq 4|V| + 2$ and $|\mathcal{A}_n| \leq 3|V| + |\mathcal{A}|$. As per Lemma 4.67, determining if a set E_j contains a circuit in M^* requires finding a maximum flow f^* for $N_{|E_j}$, which takes $O(val(f^*) \cdot (|V_n| + |\mathcal{A}_n|))$ time using the Edmonds-Karp algorithm [32] because all capacities in N are 1. Since $val(f^*) \leq |E_j|$, this can be expressed as $O(|E_j| \cdot (|V_n| + |\mathcal{A}_n|))$. Identifying the circuit involves a Depth-First search in $N^{f^*}_{|E_j}$, taking $O(|V_n| + |\mathcal{A}_n|)$ time. Thus, by applying the dual approach, determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$, and identifying this cocircuit is performed in $O(|E_j| \cdot (|V_n| + |\mathcal{A}_n|))$ time. In terms of V and $\mathcal{A}$, this corresponds to $O(|E_j| \cdot ((4|V| + 2) + (3|V| + |\mathcal{A}|)))$ time, which simplifies to $O(|E_j| \cdot (|V| + |\mathcal{A}|))$. □

Deletion

Since we use the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. However, this means that the deletion

operation in the auction corresponds to a contraction operation on our dual description. Lemma 4.70 shows how contraction can be performed on the dual matroid description.

Lemma 4.70. *Let M be a gammoid with starting vertices E and a gammoid flow network N. Let $e \in E$ and let f be a flow for N with $val(f) = 0$. Let p be a path with available capacity in N^f of the form $p = (s, e_{in}, \ldots, t)$.*
Then, M/e is the gammoid with the starting vertices $E \setminus \{e\}$ and the gammoid flow network $N_{|E\setminus\{e\}}$ where, if p exists, the edges of p are reversed in $N_{|E\setminus\{e\}}$.

Proof. Suppose e is a loop in M. This implies that the path p does not exist. Furthermore, by Lemma 2.14, $M/e = M \setminus e$. By Lemma 4.72, this means that M/e is the gammoid with the starting vertices $E \setminus \{e\}$ and the gammoid flow network $N_{|E\setminus\{e\}}$.
Otherwise, e is not a loop in M. By Lemma 4.57, a set $I \subseteq E$ is independent if and only if $val^*(N_{|I}) = |I|$. By Lemma 2.14, $\mathcal{I}(M/e) = \{I \subseteq E \setminus \{e\} : I \cup \{e\} \in \mathcal{I}(M)\}$. Let N^c be the same as the flow network $N_{|E\setminus\{e\}}$, but where the edges of p are reversed. We show that M/e is the matroid associated with the starting vertices $E \setminus \{e\}$ and the gammoid flow network N^c. Let $S \subseteq E \setminus \{e\}$.
Case 1: Suppose $S \cup \{e\} \in \mathcal{I}(M)$. Consequently, $val^*(N_{|S\cup\{e\}}) = |S \cup \{e\}|$. Let f^* be a maximum flow for $N_{|S\cup\{e\}}$ where p is one of the augmented paths in $N^{f^*}_{|S\cup\{e\}}$. Since all edges have capacity 1, reversing the edges along p is identical to augmenting the flow along p. Let f' be the flow f^* where the flow along p is reversed. Therefore, f' is a flow for N^c and $val(f') = |S|$. Consequently, $val^*(N^c_{|S}) = |S|$ and $S \in \mathcal{I}(M/e)$, where M/e is the matroid associated with the starting vertices $E \setminus \{e\}$ and the gammoid flow network N^c.
Case 2: Suppose $S \cup \{e\} \notin \mathcal{I}(M)$. Consequently, $val^*(N_{|S\cup\{e\}}) < |S \cup \{e\}|$. Since all edges have capacity 1, reversing the edges along p is identical to augmenting the flow along p. Therefore, $val^*(N^c_{|S}) < |S|$. Consequently, $S \notin \mathcal{I}(M/e)$, where M/e is the matroid associated with the starting vertices $E \setminus \{e\}$ and gammoid flow network N^c. □

Example 4.22. Let M be the gammoid from Example 4.18. Then, $M/2$ is the gammoid with starting vertices $E = \{1\}$ and the flow network shown in Figure 4.28, assuming that $p = (s, 2_{\text{in}}, 2_{\text{out}}, 3_{\text{in}}, 3_{\text{out}}, t)$ is the discovered augmenting path.

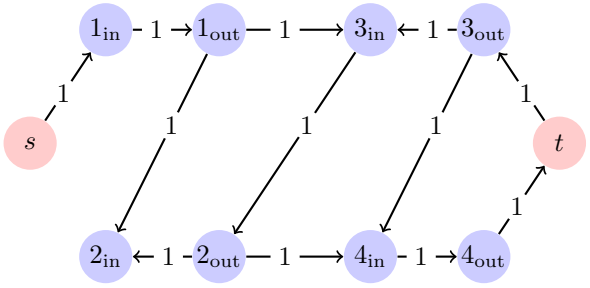


Figure 4.28: Gammoid flow network associated with $M/2$

Theorem 4.71. *Let M be a gammoid with starting vertices E, destination vertices T, and the directed graph $D = (V, \mathcal{A})$. The deletion operation in Auction 2 for M is performed in $O(|E| + |V| + |\mathcal{A}|)$ time.*

Proof. Since we use the dual approach, deleting an element e in a matroid M corresponds to contracting e in the dual matroid M^*. Let N be the associated gammoid flow network of M^* with $N = (D_n, c, s, t)$ and $D_n = (V_n, \mathcal{A}_n)$. By Lemma 4.66, $|V_n| \leq 4|V| + 2$ and $|\mathcal{A}_n| \leq 3|V| + |\mathcal{A}|$. Let f be a flow for N with $val(f) = 0$. According to Lemma 4.70, contraction in M^* involves finding a path of the form $(s, e_{in}, \ldots, t)$ in N^f and reversing the edges of this path. Identifying the path takes $O(|V_n| + |\mathcal{A}_n|)$ time using Breath-First search, and reversing reversing the edges of this path requires $O(|\mathcal{A}_n|)$ time. Furthermore, e is removed from the starting vertices E, which takes $O(|E|)$ time. Restricting N to $N_{|E\setminus\{e\}}$ takes constant time. Therefore, deletion for M in the auction takes $O(|E| + |V_n| + |\mathcal{A}_n|) = O(|E| + (4|V| + 2) + (3|V| + |\mathcal{A}|))$ time. This simplifies to $O(|E| + |V| + |\mathcal{A}|)$ time. □

Contraction

Using the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. Consequently, the contraction operation in the auction corresponds to a deletion operation in our dual description. Lemma 4.72 shows how deletion can be performed on the dual matroid description.

Lemma 4.72. *Let M be a gammoid with starting vertices E and a gammoid flow network N. Let $e \in E$. Then, $M \setminus e$ is the gammoid with the starting vertices $E \setminus \{e\}$ and the gammoid flow network $N_{|E\setminus\{e\}}$.*

Proof. By Lemma 4.57, a set $I \subseteq E$ is independent in M if and only if $val^*(N_{|I}) = |I|$. By the definition of deletion, $\mathcal{I}(M \setminus e) = \{I \in \mathcal{I}(M) : e \notin I\}$. Thus, $S \in \mathcal{I}(M \setminus e)$

if and only if $S \subseteq E \setminus \{e\}$ and $val^*(N_{|S}) = |S|$. Since e is deleted from the starting vertices E, the edges (s, e_{in}) and (e_{in}, s) are never used again. Consequently, we also restrict N to $N_{|E\setminus\{e\}}$. Therefore, $M \setminus e$ is the gammoid with the starting vertices $E \setminus \{e\}$ and the gammoid flow network $N_{|E\setminus\{e\}}$. □

Theorem 4.73. *The contraction operation in Auction 2 for a gammoid is performed in $O(|E|)$ time.*

Proof. Since we use the dual approach, contracting an element e in a matroid M corresponds to deleting e in the dual matroid M^*. Let N be the associated gammoid flow network of M^* with $N = (D_n, c, s, t)$ and $D_n = (V_n, \mathcal{A}_n)$. According to Lemma 4.72, deletion in M^* requires removing e from the set of starting vertices E, taking $O(|E|)$ time when E is stored as a hash set. Restricting N to $N_{|E\setminus\{e\}}$ takes constant time. Therefore, contraction overall takes $O(|E|)$ time. □

4.2.6 Linear Matroid

Description

A compact description of a linear matroid is a $m \times n$ matrix over some field $\mathbb{F}$. We do not use any auxiliary data structures for representing the linear matroid.

Dual Matroid

The class of linear matroids is closed under duality. To derive the dual, we follow the approach by Oxley [17]. First, we introduce the concept of the reduced row-echelon form.

Definition 4.74. The *reduced row-echelon form* is a specific form of a matrix that can be achieved through a series of elementary row operations. These operations include row swapping, multiplying a row by a non-zero scalar, and adding a multiple of one row to another [33].

A matrix is in reduced row-echelon form if it satisfies the following conditions:

1. The first non-zero element in each row, known as the *pivot element*, is 1. A column containing a pivot element is called a *pivot column*.

2. Each pivot element is the only non-zero entry in its column.

3. The pivot element in each row is to the right of the pivot element in the previous row.

4. Rows consisting entirely of zeros are positioned at the bottom of the matrix.

Example 4.23. Figure 4.29 shows four matrices in reduced row-echelon form.

Matrix 1
$$\begin{bmatrix} 1 & 0 & 0 & * \\ 0 & 1 & 0 & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix 2
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix 3
$$\begin{bmatrix} 1 & * & 0 & * & 0 \\ 0 & 0 & 1 & * & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix 4
$$\begin{bmatrix} 1 & 0 & 0 & * \\ 0 & 1 & 0 & * \\ 0 & 0 & 1 & * \end{bmatrix}$$

Figure 4.29: Matrices in reduced row-echelon form. The starred entries (*) may have any value, including zero.

Definition 4.75. The *duality-respecting description* of a linear matroid is a matrix of the following form. Here, I_r refers to the identity matrix with r rows and D denotes the matrix formed by the remaining columns.

$$\begin{array}{c} \begin{array}{cccccccc} 1 & 2 & \ldots & r & r{+}1 & r{+}2 & \ldots & n \end{array} \\ \left[\begin{array}{cccc|cccc} & & & & & & & \\ & & I_r & & & & D & \\ & & & & & & & \end{array}\right] \end{array}$$

Lemma 4.76 (Adapted from Theorem 2.2.8 in [17])**.** *If M is the linear matroid associated with the duality-respecting matrix $[I_r, D]$, then M^* is the linear matroid associated with $[-D^T, I_{n-r}]$.*

Theorem 4.77. *The dual matroid description of a linear matroid can be derived in $O(m \cdot |E|^2)$ time, where m refers to the number of rows in the matrix.*

Proof. The duality-respecting description is essentially the reduced row-echelon form with the columns rearranged such that the identity matrix appears on the left. This form can be achieved through a series of elementary row operations and row deletions in $O(m \cdot |E|^2)$ time using Gaussian elimination [33]. Once the matrix is in reduced row-echelon form, the necessary column swaps to position the identity matrix on the left side are performed in $O(|E|)$. Deriving the dual description by transposing and changing the sign of the rightmost columns D takes $O(m \cdot |E|)$ time. Creating and appending the new identity columns requires $O(m_d \cdot |E|)$ time, where m_d refers to

the number of rows in the dual description. Since $m_d \leq |E|$, the overall worst-case time complexity of deriving the dual description is $O(m \cdot |E|^2)$. □

Example 4.24. Consider the linear matroid M associated with the following 4×6 matrix:

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix}$$
$$\begin{bmatrix} 2 & 4 & 6 & 16 & 10 & 12 \\ 1 & 3 & 5 & 7 & 9 & 11 \\ 4 & 5 & 6 & 17 & 8 & 9 \\ 1 & 2 & 1 & 5 & 3 & 2 \end{bmatrix}$$

We ensure that the matrix is duality-respecting by applying elementary row operations. In this case, no column swaps are necessary.

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 & -2 & -2 \\ 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

We derive the description of the dual matroid M^* according to Lemma 4.76.

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix}$$
$$\begin{bmatrix} 2 & -2 & -1 & 0 & 1 & 0 \\ 2 & -1 & -2 & 0 & 0 & 1 \end{bmatrix}$$

Cocircuit

We determine if a bidder j controls a cocircuit in M by checking if E_j contains a circuit in the dual matroid M^*. Lemma 4.78 allows us to determine if a bidder j controls a circuit, while Lemma 4.79 enables us to identify this circuit.

Lemma 4.78. *Let M be a linear matroid. The set E_j contains a circuit C if the reduced row-echelon form of the matrix formed by the columns E_j has at least one non-pivot column.*

Proof. We show the contrapositive. Suppose the reduced row-echelon form of the matrix formed by the columns corresponding to E_j only has pivot columns. This implies that the columns E_j are linearly independent. Hence, $E_j \in \mathcal{I}(M)$. Therefore, E_j cannot contain a circuit. □

Lemma 4.79. *Suppose E_j contains a circuit C. Let A_j be the reduced row-echelon form of the matrix formed by the columns E_j. Let $v = [a_1, \ldots, a_m]^t$ be a non-pivot column in A_j and let e be the index of v. Then*

$$C = \{e\} \cup \{f \in E_j : \textit{column } f \textit{ has a pivot element in row } k, a_k \neq 0, 1 \leq k \leq m\}.$$

Proof. Note that there exists at most one non-pivot column v in the reduced row-echelon form of the matrix formed by the columns E_j. Suppose there exists another column that does not introduce a new pivot element. Then, both of these columns can be expressed by a linear combination of pivot columns. This would imply that bidder j would control two circuits in the dual matroid, and therefore two cocircuits in the original matroid, which contradicts Lemma 3.1 applied to the dual matroid. Let $v = [a_1, \ldots, a_m]^t$.
Then, v is a minimal linear combination of the columns $w \in I$ that contain the pivot elements corresponding to the non-zero entries of v. Therefore, the columns $I \cup \{v\}$ are dependent. Let C be the set containing the indices corresponding to $I \cup \{v\}$. We show that C is a circuit by showing that the columns corresponding to the indices $C \setminus \{f\}$ are linearly independent for all $f \in C$.
Case 1: Suppose f is the index of v. Then, $C \setminus \{f\}$ are the indices corresponding to the columns I. Since I contains only pivot columns, I is linearly independent.
Case 2: Suppose f is not the index of v. Therefore, f is the index corresponding to a pivot column $w \in I$. Suppose that w is the pivot column that contains the pivot element in the k-th row. Since v has a non-zero element in the k-th row and all columns in $I \setminus \{w\}$ have pivot elements in rows different from k, v cannot be expressed as a linear combination of the columns $I \setminus \{w\}$. Since all columns $I \setminus \{w\}$ are pivot columns, and v is not a linear combination of these columns, $I \setminus \{w\} \cup \{v\}$ is linearly independent. □

Example 4.25. Consider the linear matroid M with the following associated 3×6 matrix. Suppose a bidder j is interested in the columns $E_j = \{1, 2, 4, 6\}$.

$$\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 2 & 4 & 6 & 8 & 10 & 12 \\ 1 & 3 & 5 & 7 & 9 & 11 \\ 4 & 5 & 6 & 7 & 8 & 6 \end{bmatrix} \end{array}$$

Transforming the matrix formed by the columns E_j into reduced row-echelon form

yields the matrix:

$$\begin{matrix} 1 & 2 & 4 & 6 \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & -2 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since column 4 is a non-pivot column, the columns E_j are linearly dependent and contain a circuit. Moreover, the columns containing the pivot elements corresponding to the non-zero entries in non-pivot column 4 are the columns 1 and 2. Therefore, the circuit is $C = \{4\} \cup \{1, 2\} = \{1, 2, 4\}$.

Theorem 4.80. *Determining whether a bidder controls a cocircuit $C^* \in \mathcal{C}^*$ in a linear matroid and identifying this cocircuit, takes $O(m_d \cdot |E_j|^2)$ time, where m_d is the number of rows of the dual matrix.*

Proof. By Theorem 4.77, the dual description is constructed in $O(m \cdot |E|^2)$. However, since the dual description is reused throughout the auction, the time for deriving the dual is negligible. Furthermore, according to Lemmas 4.78 and 4.79, determining if the set E_j contains a circuit in the dual description and identifying this circuit involves transforming the submatrix of the dual matrix corresponding to the columns E_j into reduced row-echelon form, which takes $O(m_d \cdot |E_j|^2)$ time, where m_d is the number of rows of the dual matrix [33]. □

Deletion

Since we use the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. However, this means that the deletion operation in the auction corresponds to a contraction operation on our dual description. Lemma 4.82 shows how contraction can be performed on the dual matroid description.

Definition 4.81. Let A be an $m \times n$ matrix, and let e be a column index of A. Define A/e as the matrix obtained from A by the following series of operations:

1. Apply elementary row operations to transform the column with index e into a unit vector. A unit vector in this context is defined as a column vector that has exactly one entry with a value of 1, and all other entries are 0.

2. Remove the row containing the non-zero element in the unit vector from A.

3. Remove the column with index e from A.

Lemma 4.82 (Adapted from Proposition 3.2.6 in [17])**.** *Let M be a linear matroid associated with a matrix A. Let e be a column index of A. Then, M/e is the linear matroid associated with the matrix A/e.*

Example 4.26. Figure 4.30 shows the contraction of the column with index 3.

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 2 & 1 & 2 & 2 \\ 0 & -1 & 1 & 1 \\ 3 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

(a) Initial linear matroid description

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 3 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

(b) Transform column 3 to a unit vector

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

(c) Remove the third row

$$\begin{array}{c} \begin{matrix} 1 & 2 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 0 \\ 3 & 1 & 0 \end{bmatrix} \end{array}$$

(d) Remove column 3

Figure 4.30: Contraction of the column with index 3.

Theorem 4.83. *The deletion operation in Auction 2 for a linear matroid is performed in $O(m_d \cdot |E|)$ time, where m_d refers to the number of rows of the dual matrix.*

Proof. Since we use the dual approach, deleting an element e in a matroid M corresponds to contracting e in the dual matroid M^*. According to Lemma 4.82, contraction in M^* involves unitizing the column e, which takes $O(m_d)$ time, deleting the row, which takes $O(|E|)$ time since the matrix is stored as a list of lists, and deleting the column e, which takes $O(m_d \cdot |E|)$ time. Therefore, deletion in a linear matroid takes $O(m_d \cdot |E|)$ time. □

Contraction

Using the dual approach, we maintain the description of M^* throughout the auction instead of the description of M. Consequently, the contraction operation in the auction corresponds to a deletion operation in the dual description. Lemma 4.84 shows how deletion can be performed on the dual matroid description.

Lemma 4.84 (Adapted from Proposition 3.2.3 in [17])**.** *Let M be a linear matroid associated with the $m \times n$ matrix A. Let e be a column index of A. Then, $M \setminus e$ is the linear matroid associated with the submatrix of A where the column with index e is removed.*

Theorem 4.85. *The contraction operation in Auction 2 for a linear matroid is performed in $O(m_d \cdot |E|)$ time, where m_d refers to the number of rows of the dual matrix.*

Proof. Since we use the dual approach, contracting an element e in a matroid M corresponds to deleting e in the dual matroid M^*. According to Lemma 4.84, deletion in M^* involves removing the column corresponding to e, which takes $O(m_d \cdot |E|)$ time since the matrix is stored as a list of lists. □

4.3 Time Complexities

Table 4.1 compares the worst-case time complexities analyzed in Section 4.2.
In this table, E refers to the ground set, and E_j denotes the elements of interest to a bidder j.

Matroid	**Dual**	**Cocircuit**	**Deletion**	**Contraction**
Uniform	$O(1)$	$O(1)$	$O(\|E\|)$	$O(\|E\|)$
Partition[1]	$O(n_p)$	$O(n_p)$	$O(\|E\|)$	$O(\|E\|)$
Planar[2]	$O(\|E\| + \|V\|)$	$O(\|E_j\|)$	$O(\|E\|)$	$O(\|E\|)$
Graphic[2, 3]	undefined	$O(\|E\| \cdot \alpha(\|V\|))$	$O(\|E\|)$	$O(\|E\| + \alpha(\|V\|))$
Gammoid[2, 4]	$O(\|V\|^2 + \|\mathcal{A}\|^2)$	$O(\|E_j\| \cdot (\|V\| + \|\mathcal{A}\|))$	$O(\|E\| + \|V\| + \|\mathcal{A}\|)$	$O(\|E\|)$
Linear[5, 6]	$O(m \cdot \|E\|^2)$	$O(m_d \cdot \|E_j\|^2)$	$O(m_d \cdot \|E\|)$	$O(m_d \cdot \|E\|)$

[1] n_p: The number of parts in the partition.
[2] V: The set of vertices in the graph.
[3] α: The inverse Ackermann function.
[4] $\mathcal{A}$: The set of edges in the directed graph.
[5] m: The number of rows in the original matrix.
[6] m_d: The number of rows in the dual matrix, given by $m_d = |E| - r$, where r is the rank of the original matrix.

Table 4.1: Comparison of the worst-case time complexities for the matroid operations across different matroid classes.

5 Running Time Comparison

We implemented Auction 2 and our operations on compact matroid descriptions in `Python`. The code is available on `GitHub` [34]. In this chapter, we compare the running time across the different representable matroids based on this implementation.

5.1 Environment

All code is written in `Python 3.10.12`. While most of the code is written from scratch, we rely on some external libraries. `Pytest` is used for unit testing the individual matroid operations and integration testing the entire auction. `SymPy` and `NumPy` are used to work with the matrices of the linear matroid, and `NetworkX` is used to derive the planar embedding and graph faces for the planar matroid.

The running time analysis is conducted on a machine that uses `Microsoft Windows 10 Pro` as the operating system. The machine is equipped with an AMD Ryzen 5 3400G processor with Radeon Vega Graphics, operating at a clock speed of 3700 MHz with 4 Cores. The system has 16.0 GB of RAM.

5.2 Methodology

Figure 2.3 visualizes the hierarchy among the representable matroids discussed in this thesis. It illustrates that planar matroids form a subset of graphic matroids, which, in turn, are a subset of linear matroids. Additionally, uniform matroids are a subset of partition matroids, which are a subset of gammoids, themselves being a subset of linear matroids. Consequently, we conduct two main running time comparisons focusing on the two branches: planar-graphic-linear and uniform-partition-gammoid-linear.
We compare the methods by generating random instances of the least general matroids, namely random planar matroids and random uniform matroids. These in-

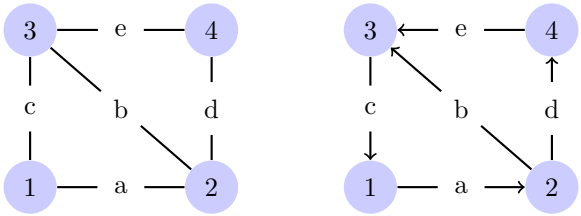
stances are then converted to descriptions of the more general classes. Subsequently, we run the auction on each of these descriptions using our implementation of the methods proposed in Chapter 4 and compare the running time. To prevent excessively long execution times, we limit the auction duration to 100 seconds.

5.2.1 Planar-Graphic-Linear

We conduct the analysis by generating random, 2-connected, planar graphs of varying sizes. Furthermore, we create bidders with random interests and valuations of the graph edges. The graph must be 2-connected as a bidder could otherwise control a bridge at the start of the auction and, therefore, a cocircuit. We run the auction on this graph using the planar matroid approach, the graphic matroid approach, and the linear matroid approach. The graph can directly be used as input in the planar and graphic matroid approaches. In the case of the linear matroid approach, the graph is first converted to a linear matroid description, which is the vertex-edge incidence matrix of the underlying directed graph of the associated undirected graph [17].

Definition 5.1. The *vertex-edge incidence matrix* of a directed graph represents the relationship between vertices and edges, with entries indicating whether each vertex is the tail (-1), head (1), or not incident (0) to each edge.

Example 5.1. Figure 5.1 shows the conversion of a graphic matroid description to the description of a linear matroid.



(a) Undirected graph

(b) Underlying directed graph

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

(c) Vertex-edge incidence matrix

Figure 5.1: Conversion of an undirected graph (a) to its underlying directed graph (b), and then to its vertex-edge incidence matrix (c).

Overall, we conduct three different running time comparisons based on the vertex-edge ratios of the input graphs. One comparison for the vertex-to-edge ratio $|V| =$

$|E|$, one for $|V| = \frac{|E|}{2}$, and one for $|V| = \frac{|E|}{3} + 6$. Note that by Euler's formula, $3|V| - 6$ is the maximum possible number of edges in a planar graph [29]. We then measure the time taken to derive the dual description and the time it takes for the auction to finish.

5.2.2 Uniform-Partition-Gammoid-Linear

We begin our analysis by generating uniform matroids with varying ground set sizes. Additionally, we create bidders with randomly assigned interests and valuations for the elements within the ground set. We run the auction on this graph using the uniform matroid approach, the partition matroid approach, the gammoid approach, and the linear matroid approach. In the case of the uniform matroid approach, the input can be used as it is, with groundset E and integer k. Since a uniform matroid is a special case of a partition matroid, the input for the partition matroid approach is simply the partition $P_1 = E$ and integer $k_1 = k$. In the case of the gammoid, we first convert the uniform matroid to a gammoid description according to the proof of Lemma 2.32. Therefore, we achieve this by creating a directed graph with vertices for all $e \in E$ and making E the set of starting vertices. We then add k different additional vertices and make these the destination vertices T. Then, we connect each starting vertex to each destination vertex.

Example 5.2. The gammoid description of a uniform matroid with ground set $E = \{1, 2, 3, 4\}$ and integer $k = 3$ has starting vertices $E = \{1, 2, 3, 4\}$, destination vertices $T = \{t_1, t_2, t_3\}$, and is associated with the directed graph in Figure 5.2.

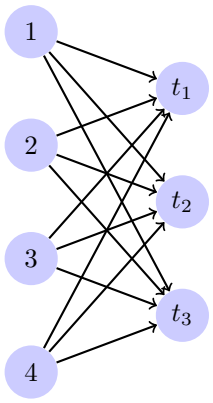


Figure 5.2: Directed graph associated with the gammoid description of a uniform matroid with starting vertices $E = \{1, 2, 3, 4\}$ and $k = 3$.

The linear matroid description corresponding to a uniform matroid with ground set size n and integer k is a $k \times |E|$ matrix where each entry is the column index raised to the power of $(row\ index - 1)$ [35].

Example 5.3. The linear matroid description corresponding to a uniform matroid with ground set size n and integer k is:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ 1 & 2^2 & 3^2 & \cdots & n^2 \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 2^{k-1} & 3^{k-1} & \cdots & n^{k-1} \end{bmatrix} \end{array}$$

Overall, we conduct three different running time comparisons based on the k-n ratios of the uniform matroids. One where $k = 0.1n$, one where $k = 0.5n$, and one where $k = 0.9n$. We then measure the time taken to derive the dual description and the time it takes for the auction to finish.

5.3 Limitations

The implementation and running time comparison face several limitations. Our implementation automatically reveals the true valuations of bidders. An alternative, more interactive system design could involve prompting bidders in each iteration to either withdraw interest from certain elements or select an element to be awarded if they control a cocircuit. While this interactive approach is more practical for real-world applications, it is unnecessary for a running time study.

Additionally, our work relies on external libraries that may not implement algorithms exactly as described in Chapter 4. This may lead to differences when comparing the theoretical time complexities with the running times. For example, `NetworkX` employs the Left-Right Planarity test for deriving the planar embedding, rather than using the Boyer-Myrvold algorithm [36].

Furthermore, our implementation is restricted to the unit-step auction and does not include the long-step version. Therefore, the running time depends on the high-

est item valuation since the price increases incrementally. We address this by limiting the valuations of the bidders to 1000. Consequently, the unit-step auction performs at most 1000 price increase steps. For a computational context, this is a relatively small number of steps, so this limitation does not significantly affect the running time.

Finally, a more comprehensive comparison between the gammoid and the linear matroid would be beneficial, particularly involving more complex gammoids. However, deriving the matrix descriptions of complex gammoids is a challenging task and beyond the scope of this thesis.

5.4 Results

5.4.1 Planar-Graphic-Linear

Tables 5.1, 5.2, and 5.3 display the running times for varying vertex-edge ratios and ground set sizes across our planar, graphic, and linear matroid implementations. Figures 5.3, 5.4, and 5.5 illustrate the corresponding running times, respectively.

Ground Set Size	Planar Matroid	Graphic Matroid	Linear Matroid
10	0.227	0.002	0.006
20	0.004	0.003	0.01
30	0.005	0.005	0.022
40	0.006	0.007	0.037
50	0.011	0.013	0.066
60	0.014	0.014	0.101
70	0.013	0.017	0.119
80	0.023	0.027	0.155
90	0.021	0.027	0.188
100	0.025	0.034	0.221
120	0.035	0.043	0.342
150	0.049	0.064	0.462
190	0.072	0.1	0.96
240	0.228	0.209	1.499

Table 5.1: Running times for a vertex-edge ratio of $|V| = |E|$ (in seconds).

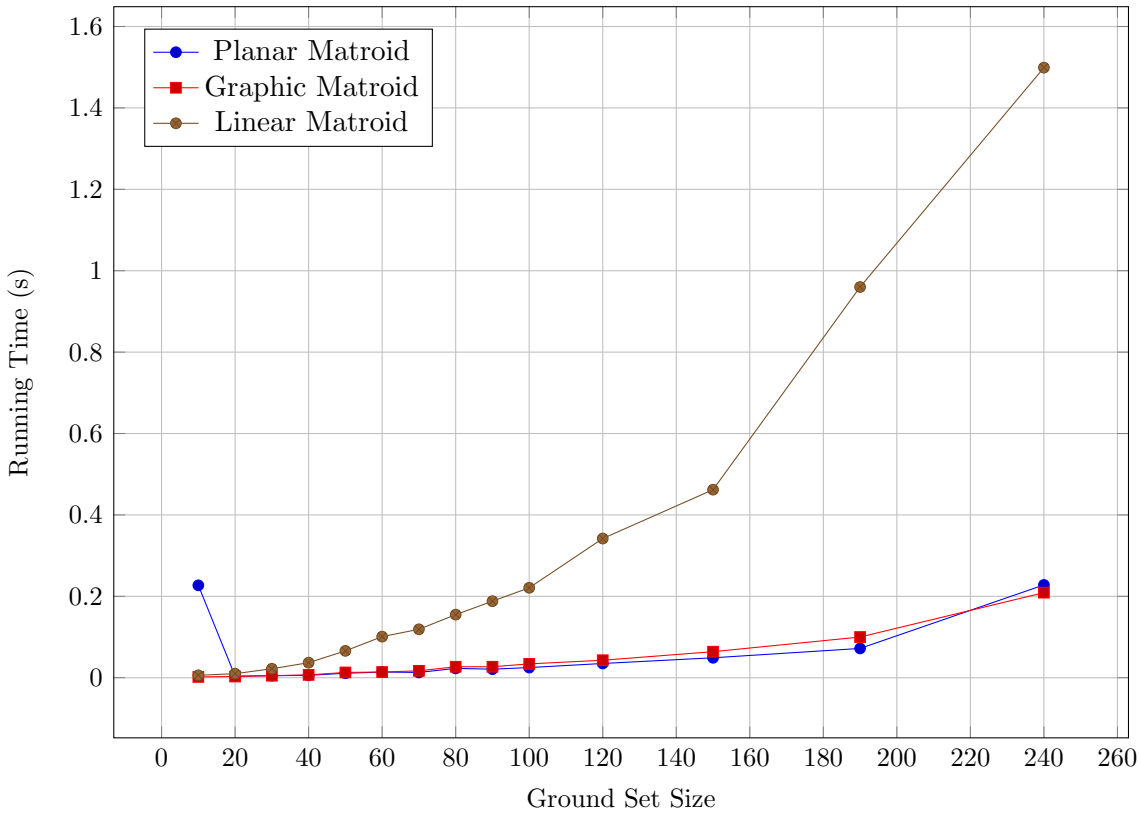


Figure 5.3: Plot illustrating the running times presented in Table 5.1.

Ground Set Size	Planar Matroid	Graphic Matroid	Linear Matroid
10	0.007	0.003	0.009
20	0.018	0.014	0.050
30	0.038	0.035	0.114
40	0.054	0.061	0.216
50	0.083	0.112	0.413
60	0.152	0.217	0.675
70	0.141	0.268	1.035
80	0.188	0.429	1.427
90	0.217	0.576	1.897
100	0.283	0.763	2.625
120	0.371	1.235	4.282
150	0.510	2.385	7.665
190	0.947	4.811	15.594
240	1.307	9.436	31.037

Table 5.2: Running times for a vertex-edge ratio of $|V| = \frac{|E|}{2}$ (in seconds).

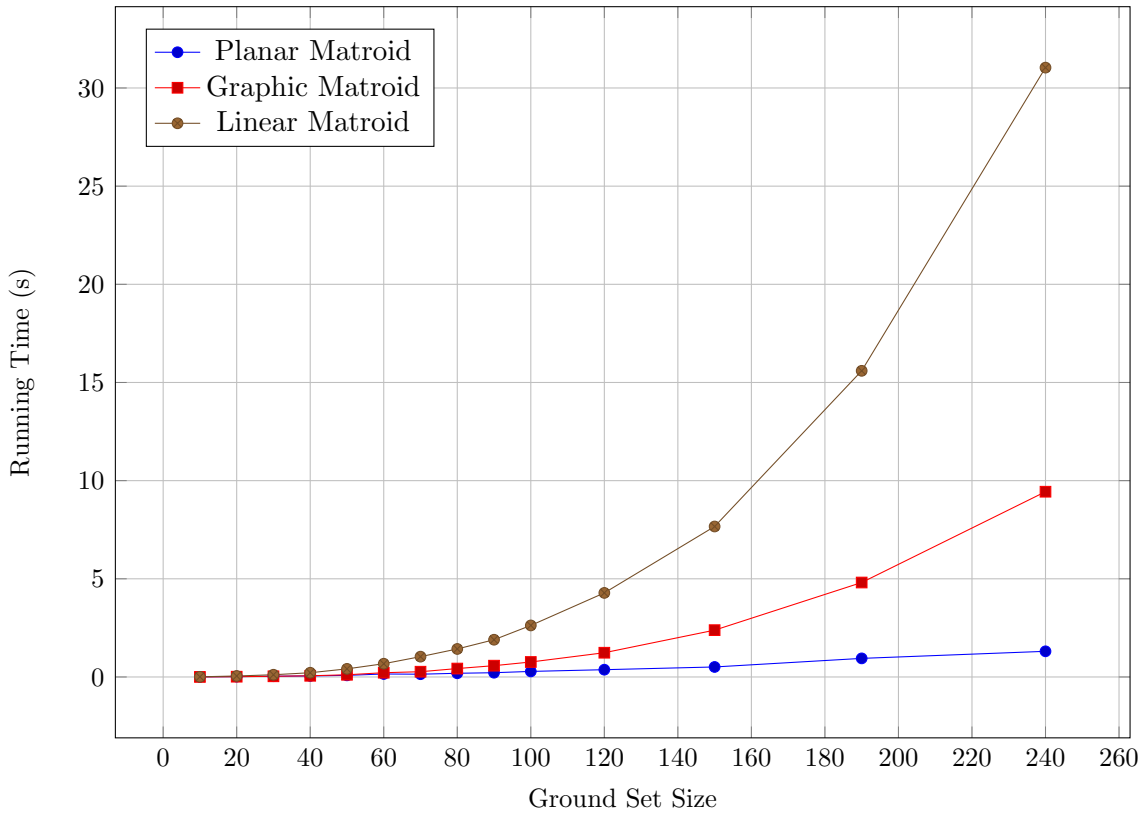


Figure 5.4: Plot illustrating the running times presented in Table 5.2.

Ground Set Size	Planar Matroid	Graphic Matroid	Linear Matroid
10	0.009	0.003	0.007
20	0.024	0.019	0.064
30	0.034	0.03	0.111
40	0.09	0.063	0.251
50	0.085	0.117	0.49
60	0.136	0.196	0.828
70	0.172	0.296	1.305
80	0.208	0.425	1.977
90	0.297	0.601	2.705
100	0.374	0.784	3.67
120	0.461	1.417	6.273
150	0.68	2.689	13.217
190	1.168	5.505	25.796
240	1.768	10.548	50.243

Table 5.3: Running Times for a vertex-edge ratio of $|V| = \frac{|E|}{3} + 6$ (in seconds).

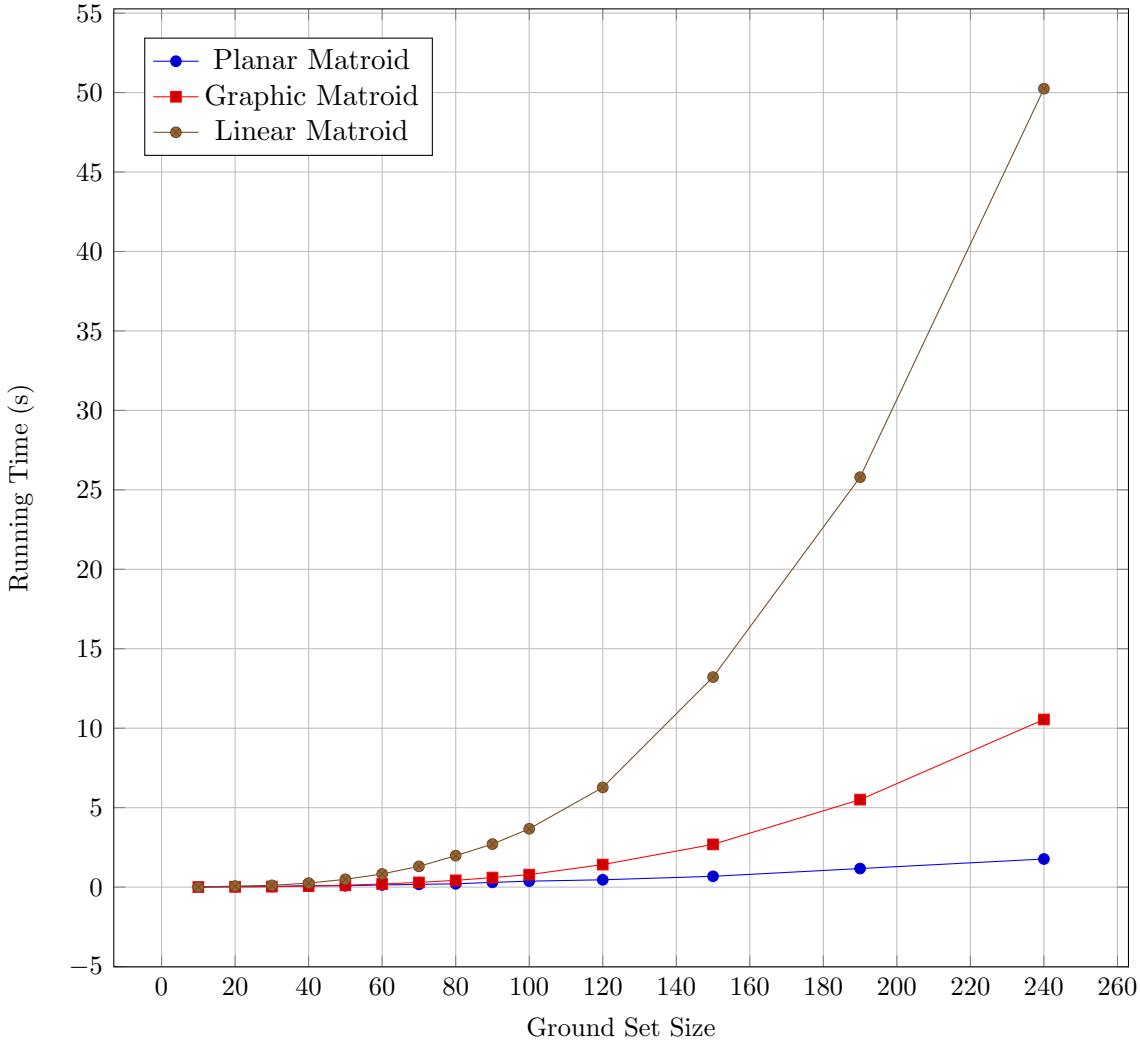


Figure 5.5: Plot illustrating the running times presented in Table 5.3.

5.4.2 Uniform-Partition-Gammoid-Linear

Tables 5.4, 5.5, 5.6 present the running times for varying k-n ratios and ground set sizes across our uniform matroid, partition matroid, gammoid, and linear matroid implementations. Figures 5.6, 5.7, and 5.8 illustrate the corresponding running times, respectively.

Ground Set Size	Uniform Matroid	Partition Matroid	Gammoid	Linear Matroid
10	0.004	0.003	0.018	0.025
14	0.006	0.005	0.043	0.046
18	0.007	0.006	0.085	0.084
22	0.007	0.009	0.166	0.143
26	0.009	0.009	0.288	0.235
30	0.015	0.013	0.431	0.294
34	0.012	0.014	0.736	0.481
38	0.012	0.013	0.895	0.63
42	0.018	0.015	1.416	0.81
46	0.018	0.017	1.939	1.106
50	0.019	0.02	2.403	1.494
60	0.027	0.025	4.427	2.323
70	0.025	0.03	7.447	3.273
80	0.034	0.041	12.121	5.152
90	0.065	0.055	18.063	7.55

Table 5.4: Running times for a k-n ratio of $k = 0.1n$ (in seconds).

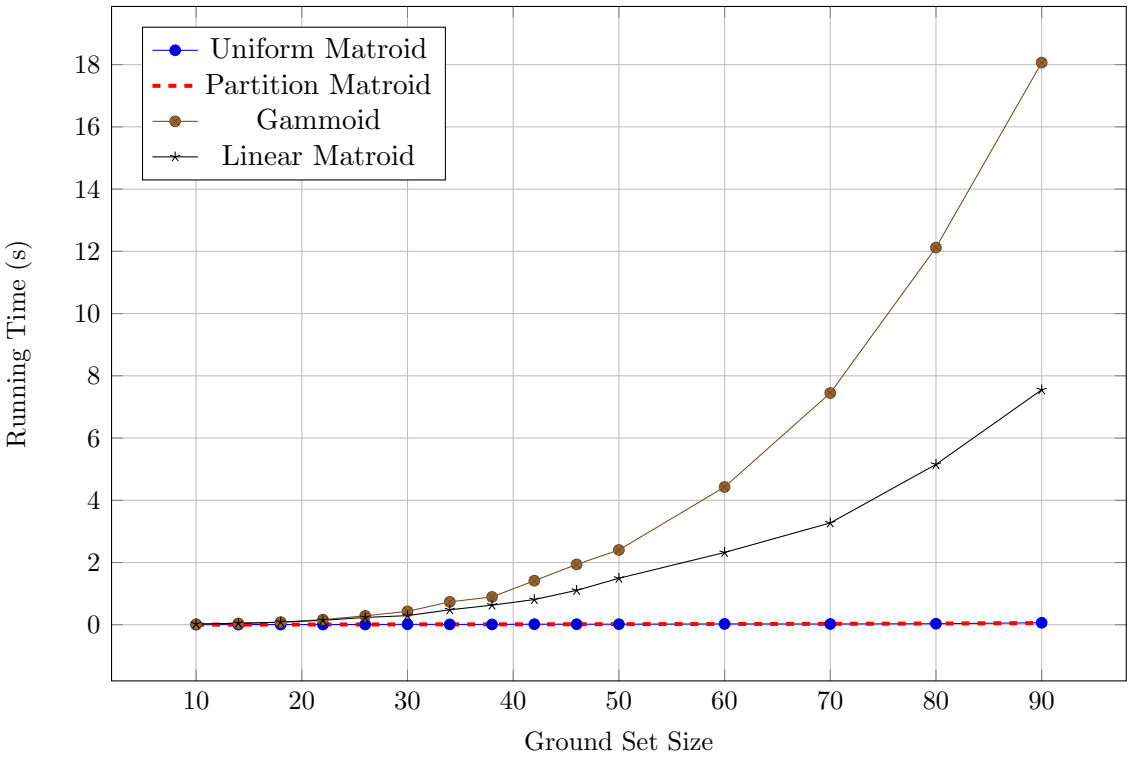


Figure 5.6: Plot illustrating the running times presented in Table 5.4.

Ground Set Size	Uniform Matroid	Partition Matroid	Gammoid	Linear Matroid
10	0.003	0.003	0.021	0.026
14	0.003	0.004	0.061	0.068
18	0.008	0.007	0.161	0.089
22	0.005	0.006	0.196	0.166
26	0.018	0.035	0.372	0.192
30	0.008	0.008	0.581	0.383
34	0.009	0.009	0.797	0.708
38	0.009	0.008	1.174	4.342
42	0.01	0.009	1.587	42.383
46	0.013	0.013	2.396	-
50	0.016	0.014	3.209	-
60	0.012	0.013	6.466	-
70	0.017	0.017	10.861	-
80	0.029	0.033	19.219	-
90	0.02	0.022	30.596	-

Table 5.5: Running times for a k-n ratio of $k = 0.5n$ (in seconds). A dash (-) indicates that the duration exceeded 100 seconds.

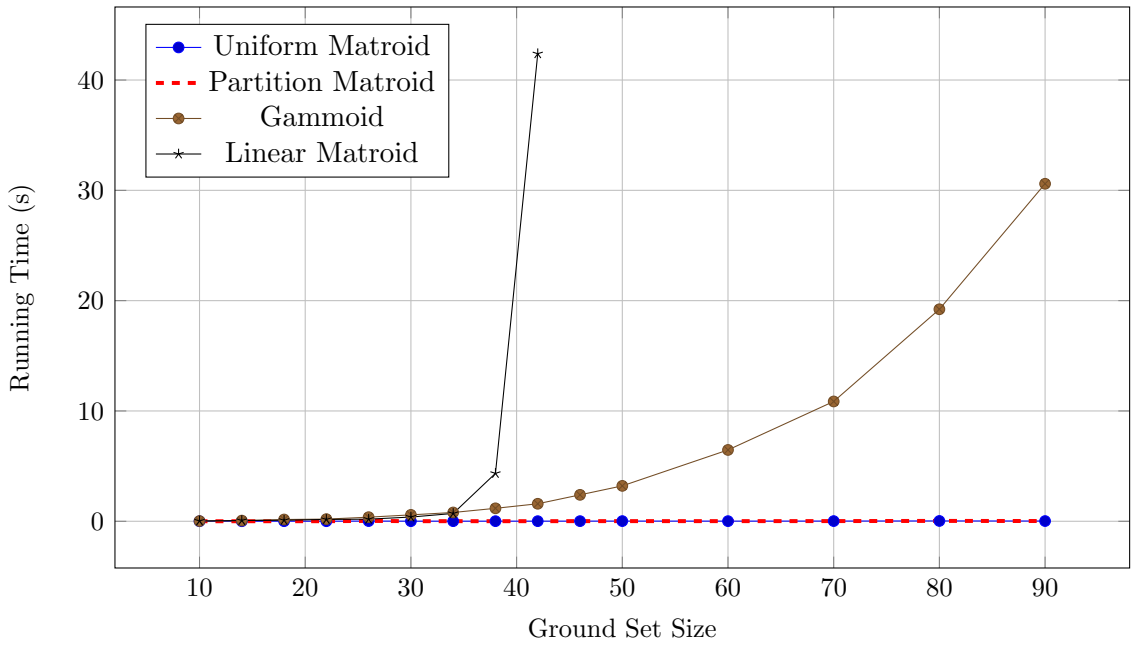


Figure 5.7: Plot illustrating the running times presented in Table 5.5. The discontinued line for the linear matroid indicates that the duration exceeded 100 seconds.

Ground Set Size	Uniform Matroid	Partition Matroid	Gammoid	Linear Matroid
10	0.001	0.001	0.006	0.015
14	0.001	0.001	0.026	0.034
18	0.002	0.004	0.037	0.095
22	0.002	0.002	0.075	1.228
26	0.003	0.004	0.111	87.449
30	0.003	0.003	0.174	-
34	0.003	0.003	0.3	-
38	0.006	0.004	0.411	-
42	0.004	0.005	0.666	-
46	0.004	0.004	0.9	-
50	0.005	0.008	1.575	-
60	0.005	0.005	2.618	-
70	0.007	0.006	3.893	-
80	0.013	0.014	7.476	-
90	0.008	0.007	12.157	-

Table 5.6: Running times for a k-n ratio of $k = 0.9n$ (in seconds). A dash (-) indicates that the duration exceeded 100 seconds.

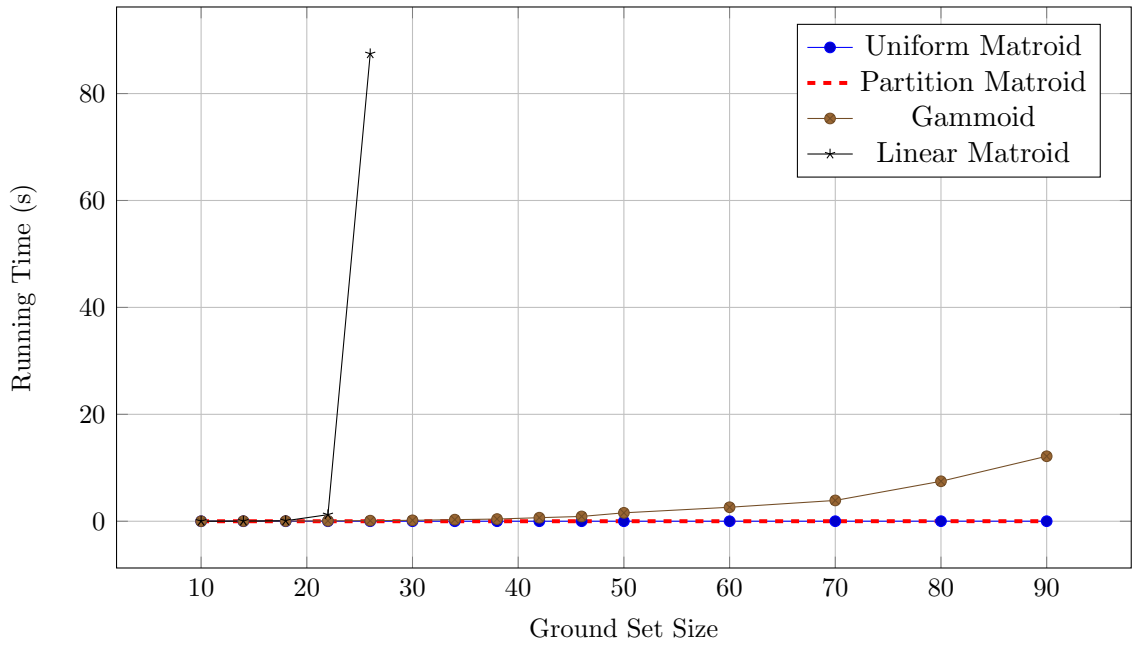


Figure 5.8: Plot illustrating the running times presented in Table 5.6. The discontinued line for the linear matroid indicates that the duration exceeded 100 seconds.

5.5 Discussion

The theoretical results presented in Section 4.3 are worst-case complexities for individual operations, whereas the implementation results consider the entire auction with empirical time operations. Comparing these is challenging due to this difference. Additionally, numerical operations like subtraction are inefficient for higher numeric values, whereas in the theoretical analysis, we assumed that numerical operations take constant time. Furthermore, the implementation is subject to the limitations outlined in Section 5.3. Despite these factors, the running times generally align with the theoretical analysis. Considering that finding cocircuits is the most frequently performed operation, both results indicate that larger ground set sizes increase the running time, and more general approaches tend to be less efficient than specialized ones.

5.5.1 Planar-Graphic-Linear

Planar Matroid

The planar matroid implementation is notably fast, even considering the initial time needed to derive the dual of the planar graph. Its efficiency surpasses the graphic and linear matroid implementations due to finding cocircuits taking $O(|E_j|)$. Additionally, the derivation of the dual matroid only takes $O(|E| + |V|)$, placing it on a similar scale as a single cocircuit lookup in the graphic implementation, which takes $O(|E|+\alpha(|V|))$. An exception is the comparison with $|V| = |E|$, making the running time of the graphic and planar implementations nearly identical. This is because, in a sparse graph, after initial deletions, most 1-element sets become cocircuits, resulting in most cocircuit lookups leading to contractions. As explained in Section 4.1, the time spent finding cocircuits is relatively low compared to other operations in the auction when this is the case. Nevertheless, the planar implementation remains superior in other comparisons. For example, with a ground set size of $|E| = 240$ and a vertex-edge ratio of $|V| = \frac{|E|}{2}$, the planar implementation is more than seven times faster than the graphic implementation.

Graphic Matroid

Although the graphic matroid implementation does not require deriving the dual, it is generally slower than the planar matroid implementation because it relies on the primal approach, where finding cocircuits takes $\Omega(|E|)$. Nevertheless, the im-

plementation remains efficient and faster than the linear matroid implementation due to nearly constant Union-Find operations. This enables nearly constant-time contractions and, on average, very efficient identification of cocircuits. Additionally, the number of vertices does not significantly affect the running time. This is because the graphic matroid approach mainly scales with the number of edges and, considering the vertices, only with $\alpha(|V|)$, where α is the inverse Ackermann function. The inverse Ackermann function scales so slowly that for all practical purposes, it can be considered nearly constant.

Linear Matroid

The linear matroid implementation is the least efficient of the three implementations. Its running time depends both on the ground set size and the number of rows in the dual description, given by $m_d = |E| - r$, where r is the matrix rank. A matrix with a low row-to-column ratio, such as when $|V| = \frac{|E|}{3} + 6$, results in a high number of rows in the dual description, leading to significantly longer running times compared to cases like $|V| = \frac{|E|}{2}$. As explained in Section 4.1.2, this could be improved by using the primal approach when the matrix has very few rows. However, the dual approach is still preferred in cases with a low or medium row-to-column ratio.

5.5.2 Uniform-Partition-Gammoid-Linear

Uniform Matroid & Partition Matroid:

The uniform and partition matroid implementations are highly efficient, with the uniform matroid being a special case of the partition matroid. In these cases, deriving the dual and finding cocircuits are $O(1)$ operations, explaining the consistently low running times of less than 0.1 seconds in all comparisons.

Gammoid

The gammoid implementation shows varying performance across different k-n ratios. It performs efficiently for high and low ratios but is less efficient for $k = 0.5n$. This behavior is only partially explained by theoretical worst-case complexities and mostly relates to the structure of the dual gammoid description.

For $k = 0.9n$, the dual gammoid description contains only $0.1n$ different destination vertices. Consequently, finding the maximum flow in the corresponding gammoid flow network is quick, as it requires identifying at most $0.1n$ augmenting paths. This allows the Edmonds-Karp algorithm to conclude faster.

For $k = 0.1n$, there are $0.9n$ destination vertices in the dual gammoid description. However, most of these vertices also serve as starting vertices, making the identification of vertex-disjoint paths for these vertices immediate, as each vertex corresponds to a path to itself. Only the $0.1n$ starting vertices that are not destination vertices require finding actual vertex-disjoint paths in the directed graph and thus augmenting paths in the corresponding gammoid flow network. This again allows the Edmonds-Karp algorithm to conclude quickly.
In contrast, the implementation takes more time in the case with a moderate number of paths. For $k = 0.5n$, there are $0.5n$ destination vertices in the dual description and $0.5n$ starting vertices that are not destination vertices. Consequently, it is necessary to find actual vertex-disjoint paths for these $0.5n$ starting vertices, which is more than the $0.1n$ needed in the comparison for $k = 0.1n$. Furthermore, finding a maximum flow in the corresponding gammoid flow network requires finding $0.5n$ different augmenting paths overall, which is more than the $0.1n$ needed in the comparison with $k = 0.9n$. This makes the case with $k = 0.5n$ significantly slower than the other two comparisons.

Linear Matroid

The linear matroid implementation is extremely slow in scenarios with larger values of k. Initially, it can be faster than the gammoid implementation, particularly for $k = 0.1n$. Higher values of k, however, dramatically reduce efficiency. The reason lies in the matrix description of uniform matroids, where values reach up to n^{k-1} for a ground set size n, leading to inefficient operations. Even for $k = 0.1n$, the linear matroid implementation would become slower than the gammoid implementation for substantial ground set sizes.

6 Conclusion

This thesis explored the intersection of matroid theory and auction theory, contributing to both fields. Initially, we presented the fundamentals of both fields, introduced several representable matroid classes, and examined their hierarchical relationships. We then presented the auction by Bikhchandani et al. for selling bundles of indivisible, unique goods to different bidders under matroid constraints. The main contribution of this thesis is the adaptation of this auction to several classes of representable matroids, showcasing how the auction operations, specifically finding cocircuits, deletion, and contraction, directly apply to compact matroid descriptions.

To achieve this, we outlined a schema for adapting the auction to a matroid class and introduced two approaches for identifying cocircuits: the primal and the dual approach. While the dual approach is generally more efficient than the primal approach, it requires the matroid class to be closed under duality for direct application, whereas the primal approach is universally applicable. Additionally, we analyzed and compared the worst-case time complexities of the adapted matroid operations. Our analysis revealed that although more general matroid classes can model a wider range of scenarios, they tend to be less computationally efficient.

Furthermore, we implemented the auction by Bikhchandani et al. and our custom adaptations in `Python`, conducting a running time analysis to evaluate these methods. Despite some limitations, the empirical results aligned with our theoretical analysis.

Future research could extend our approach to additional matroid classes or address the limitations identified in our `Python` implementation.

Bibliography

[1] S. Bikhchandani, S. Vries, J. Schummer, and R. Vohra, "An ascending vickrey auction for selling bases of a matroid," *Operations Research*, vol. 59, 07 2005.

[2] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

[3] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, pp. 17–33, 1971.

[4] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, pp. 617–631, 1973.

[5] L. M. Ausubel, "An efficient ascending-bid auction for multiple objects," *American Economic Review*, vol. 94, pp. 1452–1475, December 2004.

[6] P. Cramton, "Ascending auctions," *European Economic Review*, vol. 42, no. 3-5, pp. 745–756, 1998.

[7] A. Kelso and V. Crawford, "Job matching, coalition formation, and gross substitutes," *Econometrica*, vol. 50, pp. 1483–1504, 02 1982.

[8] F. Gul and E. Stacchetti, "The english auction with differentiated commodities," *Journal of Economic Theory*, vol. 92, no. 1, pp. 66–95, 2000.

[9] D. Parkes and L. Ungar, "Iterative combinatorial auctions: Theory and practice.," in *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pp. 74–81, 01 2000.

[10] L. M. Ausubel and P. R. Milgrom, "Ascending auctions with package bidding," *The B.E. Journal of Theoretical Economics*, vol. 1, 2002.

[11] S. de Vries, J. Schummer, and R. V. Vohra, "On ascending vickrey auctions for heterogeneous objects," *Journal of Economic Theory*, vol. 132, no. 1, pp. 95–118, 2007.

[12] D. Mishra and D. C. Parkes, "Ascending price vickrey auctions for general valuations," *Journal of Economic Theory*, vol. 132, no. 1, pp. 335–366, 2007.

[13] L. M. Ausubel, "An efficient dynamic auction for heterogeneous commodities," *American Economic Review*, vol. 96, pp. 602–629, June 2006.

[14] B. Peis and N. Rieken, "A simplified analysis of the ascending auction to sell a matroid base," 2024.

[15] T. Helgason, "Aspects of the theory of hypermatroids," in *Hypergraph Seminar: Ohio State University 1972*, pp. 191–213, Springer, 1972.

[16] S. Raach, *Implementing efficient outcomes in combinatorial allocation problems.* doctoralthesis, Universität Trier, 2023.

[17] J. Oxley, *Matroid Theory.* Oxford graduate texts in mathematics, Oxford University Press, 2006.

[18] M. Stallmann, "A gentle introduction to matroid algorithmics," Technical Report TR-2016-4, North Carolina State University. Dept. of Computer Science, March 2016. Available: `http://www.lib.ncsu.edu/resolver/1840.20/33534`.

[19] I. Albrecht, "Duality respecting representations and compatible complexity measures for gammoids," *Discrete Mathematics*, vol. 344, p. 112157, Jan. 2021.

[20] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 10.3)*, 2024. https://www.sagemath.org.

[21] N. Rieken, "Strategic decisions in networks, markets and politics: Lecture #10: Sealed-bid auctions and the VCG mechanism," January 22 2024. Lecture Notes.

[22] H. Whitney, "On the abstract properties of linear dependence," *American Journal of Mathematics*, vol. 57, no. 3, pp. 509–533, 1935.

[23] A. Ingleton and M. Piff, "Gammoids and transversal matroids," *Journal of Combinatorial Theory, Series B*, vol. 15, no. 1, pp. 51–68, 1973.

[24] D. Mayhew, "Matroid complexity and non-succinct descriptions," 2007.

[25] D. E. Knuth, "The asymptotic number of geometries," *Journal of Combinatorial Theory, Series A*, vol. 16, no. 3, pp. 398–400, 1974.

[26] J. E. Dawson, "Optimal matroid bases: An algorithm based on cocircuits," *The Quarterly Journal of Mathematics*, vol. 31, pp. 65–69, 03 1980.

[27] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction To Algorithms*. Mit Electrical Engineering and Computer Science, MIT Press, 2001.

[28] J. Boyer and W. Myrvold, "On the cutting edge: Simplified o(n) planarity by edge addition," *J. Graph Algorithms Appl.*, vol. 8, pp. 241–273, 01 2004.

[29] T. Nishizeki and N. Chiba, *Planar graphs: Theory and algorithms*. Elsevier, 1988.

[30] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, p. 399–404, 1956.

[31] J. H. Mason, "On a class of matroids arising from paths in graphs," *Proceedings of the London Mathematical Society*, vol. s3-25, no. 1, pp. 55–74, 1972.

[32] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, p. 248–264, apr 1972.

[33] D. Lay, S. Lay, and J. McDonald, *Linear Algebra and Its Applications Global Edition*. Pearson Deutschland, 2015.

[34] L. Kempkes, "Ascending vickrey auctions for selling bases of linear matroids repository." GitHub repository, 2024. Available: `https://github.com/lucak5s/thesis`.

[35] S. Kratsch, S. Saurabh, M. Wahlstrom, and T. Kociumaka, "Matroid theory and kernelization," 2013. Lecture series conducted on 10-11 April 2013.

[36] U. Brandes, "The left-right planarity test," *Manuscript submitted for publication*, vol. 3, 2009.