LLM-based Proxies for Preference Elicitation in Combinatorial Auctions

a thesis presented
by
David Z. Huang
to
The Department of Applied Mathematics

in partial fulfillment of the requirements
for the degree of
Bachelor of Arts
in the subject of
Applied Mathematics

Harvard University
Cambridge, Massachusetts
November 2024

Thesis advisor: Professor David Parkes, Dr. Francisco Marmolejo-Cossio David Z. Huang

LLM-based Proxies for Preference Elicitation in Combinatorial Auctions

Abstract

Elicitation in combinatorial auctions is challenging as bidder preferences may be inherently difficult to describe and consequently communicate to an auctioneer. Classical work in elicitation focuses on using query-based techniques inspired by proper learning—often via proxies that interface between bidders and an auction mechanism—to incrementally learn bidder preferences as needed to compute efficient allocations. Although such elicitation mechanisms enjoy theoretical query efficiency, the amount of communication required may still be too cognitively taxing for bidders in practical scenarios.

Significant recent advancements in natural language processing, particularly Large Language Models (LLMs), suggest the use of natural language for eliciting preferences. In this thesis, we propose an efficient LLM-based proxy design for eliciting preferences from bidders in a combinatorial auction setting where communication is limited. Our proposed mechanism combines LLM pipelines and DNF-proper-learning techniques to quickly approximate preferences with limited communication. To validate our LLM-based approach to proxy design, we create a testing sandbox for evaluating elicitation mechanisms that make use of natural language as a means of communication.

Additionally, we address the scalability of our approach by demonstrating how we can ensure complexity that is polynomial in the number of items, in both simulating bidder responses and the inference process of the proxy. By leveraging sparse preference representations and restricting inference to smaller bundle sizes, our simulation remains faithful to the ground-truth preferences and our proxy sustains high efficiency for the auction, respectively, while ensuring polynomial complexity.

Reaching approximately efficient outcomes five times faster than classical proper-learning-based elicitation mechanisms, our LLM-based approach demonstrates the potential of natural-language-based elicitation approaches. Moreover, the LLM-based proxies provide outcomes that converge, with sufficient communication, to those arising from DNF-proper-learning techniques.

Contents

List of Figures

Acknowledgments

Thank you, David, for your guidance and generosity. Thank you, Francisco, for your boundless support, kindness, and advice. Thank you, Edwin, for your insight and collaboration. To my Mom and Dad and Waverly, thank you for getting me to who I am. To Iris and Abigail, thank you for your love, strength, and patience, always.

1

Introduction

Auctions are mechanisms for the sale of items to the highest bidder, with the overall goal of ensuring that items are received by those who value them the most. Historically, auctions have been utilized to sell a wide array of goods, from art pieces to public resources.

> I require that [my things of art] shall all be dispersed under the hammer of an auctioneer, so that the pleasure which the acquiring of each one of them has given me shall be given again, in each case, to some inheritor of my own tastes.

— Will of Edmond de Goncourt, 1896[30]

An *allocation* in an auction refers to an assignment of items to bidders. Edmond de Goncourt wanted to allocate the items of his estate such that they reached "some inheritor of [his] own tastes." An auction facilitates the sale of items to recipients who value the items highly and who are willing to outbid everyone else. The *value* a person assigns to items is the worth they perceive in those items, and the *payment* is the amount they pay the auctioneer for their allocated items.

The *utility* a person derives from items they win is calculated as their value of the items minus the payment made. Bidders aim to achieve positive utility, where the value exceeds the payment, thereby maximizing their satisfaction. Conversely, negative utility implies that the payment outweighs the value, which bidders seek to avoid. Zero utility means that a bidder is indifferent towards winning the items.

Auctions are not limited to estate sales or bankruptcies; they play a crucial role in allocating and pricing public goods. This concept dates back to Eric Lindahl in 1919,[42] who discussed the allocation of public services through a financial process which is akin to auctions and taxes which are akin to payments in an auction.

> We have shown that, provided the taxpayers are all in an equal position to defend their economic interests when tax laws are passed, the financial process would result in each individual having to pay a tax amount corresponding to his valuation of public services.

— Positive Lösung by Eric Lindahl, 1919[42]

Lindahl did not precisely describe the mechanism, which he calls a "financial process", that would converge on this outcome. In modern applications, such as the Federal Communications Commission's (FCC) 2019 spectrum auctions, combinatorial auctions (CAs) have proven highly effective for efficiently allocating items. In 2019, the FCC successfully raised two billion dollars by auctioning licenses in a combinatorial format, where bidders can place bids on bundles of items rather than on only individual items[43]. The FCC chose to use CAs because CAs reach more efficient allocations. An allocation is *more efficient* than another if the social welfare of the first allocation is higher than the social welfare of the second allocation. The *social welfare* or the total value of an allocation is the sum of the values that each bidder assigns to their allocated bundles. The goal is to reach a *social-welfare-maximizing outcome*, where no other allocation yields a higher social welfare.

1.1 Combinatorial auctions (CAs)

Combinatorial auctions (CAs) are a type of auction designed to address the inefficiencies of traditional auction mechanisms, which often fail to achieve efficient outcomes in certain scenarios[47]. CAs mitigate these shortcomings[7] but introduce significant challenges in terms of cognitive and computational burdens on bidders[45]. We will explore how CA designs incorporate *proxies*[10] to assist bidders in formulating their preferences and strategically placing bids based on those preferences, which are traditionally communicated through queries in specialized languages. CAs have been successfully employed across many domains, from spectrum sales[3] to procurement[24] and beyond[39].

1.1.1 Motivating example

We start with an example that motivates why a CA achieves more efficient allocations.

When we have an auction of a single item, the bidder with the highest value for that item will offer the highest bid because they are willing to pay the most. In auctions with multiple items, however, the bidders with the highest values might not always win because of the *exposure problem*[7].

A situation where this happens is when someone values a bundle of items highly but has low value for each item individually. For example, in an estate sale, a bidder might want to buy a matching pair of nightstands but not either nightstand individually. A bidder in this situation would not be willing to bid much on each individual item because of the risk of having to pay for that item when they are unable to acquire the rest of the items in the bundle. Therefore, the auctioneer who auctions off each of the items individually would not realize the efficient outcome that the auctioneer who sells the bundle of items together for a higher value can achieve.

A general diagnosis of the exposure problem is that bidders might have *combinatorial preferences*, where bidders have preferences over bundles of items instead of *separable preferences*, where bidders have preferences over individual items, considering each item independently of the others. When bidders are unable to express their combinatorial preferences, inefficiencies can occur because auctioneers cannot take those preferences into account. *Combinatorial auctions* (CAs) are auctions that are, at their essence, designed to give bidders the ability to express their combinatorial preferences by bidding on bundles of items instead of individual items.

1.1.2 Exponential complexity in CAs

The essential fact about CAs—working with combinatorial preferences—also raises an essential challenge for the practical implementations of CAs[9]. Given N items for auction, a person with combinatorial preferences has 2^N possible bundles to express preferences over. This is exponentially more possibilities than a person with separable preferences who has only N possible individual items to express preferences over. This exponential complexity raises two practical considerations:

1. How can people intelligently formulate their preferences?
2. How can people efficiently calculate their best responses during a CA?

To address these practical considerations, CA designs often introduce *proxies* to represent a person in an auction[10]. To set up our discussion of proxies in CAs, we will first introduce bids and related concepts in Section 1.1.3. We will then introduce in Section 1.1.4 the queries traditionally used for communication between proxies and the person they represent before discussing the role of proxies in CAs in Section 1.1.5.

1.1.3 Bids

A bidder formulates their preferences in terms of a bid.

Bids submitted to an auction may sometimes strategically differ from a bidder's preferences to gain an advantage, e.g. in certain situations, one might want to pretend like one doesn't want an item in order to get it at a lower price.

When a bid does not differ from a bidder's preferences, the bid is *truthful*, so a bidder's *valuation function*—a function which maps bundles of items to the value a person has for them—can be described in terms of their truthful bid.

A bid also has a correspondence to prices. The prices during an auction can be described as the bid one would have had to make where the seller would be indifferent to selling to them any given item or bundle of items.

1.1.4 Queries

Queries are issued by proxies to request that the bidder respond with a certain piece of information about their preferences. There are many different query types, and we will focus specifically on three. These three queries will be used in our construction of the testing sandbox and of proxies. The three queries also provide a representative picture of the currently studied communication channels between a bidder and their proxy.

Value queries The proxy asks the bidder what value they would place on a specific bundle.

Demand queries The proxy asks the bidder whether they are happy with their allocated bundle at given prices, and if not, an example of a bundle they would like to switch to instead.

Equivalence queries The proxy asks the bidder whether or not the proxy's hypothesis of the bidder's valuation is correct, and if not, an example of a bundle which the hypothesis valuation values incorrectly.

In the context of computational learning theory, value queries may also be called *membership* queries. Given the ability to make membership and equivalence queries, *learning algorithms* from computational learning theory[23] can learn specific classes of preferences[26]. Learning algorithms are different from preference elicitation algorithms in that the former learns the complete preferences of a person while the latter only elicits the parts of the preferences that are necessary for representing the person in the auction and determining the final allocation.

1.1.5 Role of proxies in CAs

Proxies facilitate the interactions between the auction and the bidder in a way that helps the bidder be more efficient and effective[43]. Different proxy designs implement the facilitation of interactions differently. Proxies can help bidders by calculating the bidder's best responses given their preferences and implementing strategic bidding behavior on their behalf[41]. Proxies can also help bidders formulate their preferences via a process called *preference elicitation*[6].[*] The classical proxy has the property of only requiring a number of interactions with the person they represent with complexity polynomial in the number of atomic bundles. This demonstrates an avenue of attacking the exponential complexity of CAs. Therefore, in this work, we will focus on comparing our LLM-based

[*] Preference elicitation is often described as being performed by the auctioneer instead of explicitly by a proxy. We explicitly call out preference elicitation as being done by a proxy who then forwards to the auctioneer the elicited preferences in order to consider modifications of preference elicitation algorithms as modifications of a proxy instead of as modifications of the underlying auction mechanism.

proxies to *classical proxies*, which perform preference elicitation using methods based on learning algorithms. A formal description of a classical proxy will be given in Chapter 2.

However, currently, even with the use of proxies (whether classical or otherwise), participating in CAs can still be prohibitively expensive, because even a moderate number of interactions still takes a long time for bidders to respond to because of non-trivial response times [11,33]. This means that it is fairly expensive for bidders to participate in CAs because of the substantial time and effort required to formulate preferences and strategically bid.

1.2 Large language models (LLMs) to process natural language

Recent leaps in natural language processing using large language models (LLMs)—where LLMs can pass the bar exam [22], medal in the International Mathematics Olympiad [15], and fluently communicate in eight languages [31]—suggest that we should consider the benefits of using natural language in the communication and understanding of preferences.

1.2.1 Strengths of LLMs

Reasoning capability LLMs, evidenced by their ability to solve complex mathematical problems [19,49], are strong at reasoning over numbers and mathematical objects. This suggests the ability of LLMs to understand relatively complex pricing and valuation schemes.

Base of knowledge Being trained on an enormously large amount of data,[†] LLMs have the knowledge that allows them to ace a wide swath of standardized tests [37]. LLMs have demonstrated a command of these concepts and are able to understand their expression in natural language. This

[†]The exact data used for LLMs is not known for proprietary LLMs and is not known even for top LLMs with open-source weights and code. We do know that they are trained from 800 GB [14] to 15,000 GB [31] with data from sources like Wikipedia, textbooks, research articles, and forum posts.

provides some intuition that LLMs will have the facility to make decisions based on natural language descriptions of items and preferences.

Generalizing to social sciences tasks Studies have shown the ability of LLMs to function as models of cognitive behavior[4] and to exhibit similar intuitions as humans[18]. Specifically in economic tasks, LLM-based agents have been able to function fairly strategically and similar to humans[20,5]. These studies will be discussed in depth when discussing related works in Section 1.4.

1.2.2 Weaknesses of LLMs

Hallucinations LLMs have been shown to hallucinate and generate non-factual responses. Encouraging evidence shows that hallucination can be mitigated with techniques like prompt engineering[21].

Context length LLMs have a limited context length, meaning that one can only communicate to the LLM a capped amount of text. Engineering has been done to vastly increase the cap[32], but this still means that for large numbers of items we are restricted in the number of bundles we can communicate prices or valuations for.

Context length constraints highlight the role of bounded rationality theories[48] even when using LLMs and computational methods, and the importance of communication complexity considerations[36].

Costly calls LLM calls have a much higher associated cost versus traditional computations, because billions of arithmetic operations are required to compute the output of an LLM as LLMs are at a minimum size consisting of billions of parameters[31]. Reducing the amount of LLM calls and thus reducing the computational cost will be a continued theme in our design of simulations and of proxies.

1.3 Summary of contributions

We consider LLMs as a technology to improve the efficiency of allocating items in a combinatorial auction by using LLMs to improve proxies that represent simulated people in a combinatorial auction. LLMs enable computer systems to make decisions based on natural language, suggesting that LLM-based proxies can understand preferences expressed in natural language. Thus, we develop LLM-based proxy designs to utilize these strengths of LLMs to improve a proxy's ability to elicit the person's preferences, in the sense of reducing the amount of communication required from the person and the cognitive burden required to produce that communication.

To evaluate these LLM-based proxy designs, we develop a testing sandbox, which allows us to simulate runs of combinatorial auctions, each with varying setups of items for auction and participating simulated persons. We use a pipeline of LLM calls to create a simulated person with coherent, combinatorial preferences who can respond to natural language questions about their preferences, in addition to responding to value and demand queries. This is done by generating a seed, a natural language description, to characterize the simulated person's preferences. We run checks on the generated seeds to show that they give reasonable, robust, and precise responses to value queries by examining the shape and distribution of valuations given by responses to the queries. This method of generating seeds and the preferences they characterize is of broader interest because these seeds define a class of preferences that can be readily expressed in natural language. However, further investigations beyond the checks we run, including human comparisons, are left to future works and discussed further in the concluding chapter.

Having implemented the simulated persons in the testing sandbox, we next incorporate LLMs into proxy designs to reduce the number of interactions between a person and their proxy, while maintaining efficient outcomes. Using nine toy setups—each setup has a population of three simulated people under one of three unique scenarios we design with six items each—in our testing sandbox, we measure the social welfare of the allocation resulting from simulating an auction with LLM-based proxies representing the simulated persons. We compare our LLM-based proxies to classical proxies on the basis of the social welfare of the resulting allocation and the amount of communication between the LLM-based proxies and the people they represent, measured by the number of interactions between them. We demonstrate that compared to classical proxies—which already only require a number of interactions with the person they represent with polynomial complexity in the number of items—LLM-based proxies even more quickly reach approximately efficient outcomes, using five times fewer interactions. That is, when using an LLM-proxy, we reach 75% efficiency of the auction in two interactions between a proxy and the person they represent, while, when using a classical proxy, it takes ten interactions between a proxy and the person they represent. [‡]

LLM-based proxies achieve this level of performance through three features: competently using value and demand queries, asking questions in natural language, and inferring preferences that are not explicitly stated. Together, these features mean that the LLM-based proxies can communicate more effectively with the person they represent and submit bids that are comprehensive without requiring the person to explicitly spell out every single preference.

Having shown the ability of LLM-based proxies to quickly learn preferences and achieve approximately efficient outcomes in just a few interactions, we further develop a *hybrid* LLM-based proxy which incorporates key algorithms from classical proxies. We demonstrate that by merging the classical and LLM designs, in the long run, we can ensure that these hybrid LLM-based proxies converge to the same social welfare value as classical proxies.

[‡]The efficiency of an auction is calculated as the social welfare of the allocation of the auction divided by the maximum social welfare of any possible allocation, given ground-truth valuations. This calculation is described in detail in Chapter 2.

Given how costly it is to use LLMs, our implementations must also be efficient in the number of LLM calls made in service of both running the simulated persons and the LLM-based proxies. We show that we can apply strategies to restrict the number of LLM calls made and still sustain high efficiency for the auction. This is a crucial step in demonstrating we can scale up both the simulation and the proxy from handling our toy setup to handling large numbers of items. Using a strategy to restrict the number of LLM calls made in the construction of the simulated person to be polynomial in the number of items, we approximate the simulated person's preferences with a representation that is on average 25% of the original size. We show that we can restrict the number of LLM calls as such without any statistically significant impact on the validity of the simulated person—where we measure validity as the lack of change in the efficiency of the auction after using the approximation—in two of the three scenarios, and only a 5.6% decrease in efficiency in the third scenario. Using a strategy to restrict the number of LLM calls made by the LLM-based proxies to be polynomial in the number of items, we reduce the maximum number of inferences the LLM-based proxies makes to 23% of the original maximum. We show that this restriction has no statistically significant impact on the performance of the LLM-based proxies.

The limitation of our empirical studies in demonstrating that there is little performance hit to using the polynomial-complexity restricted strategy is that we only demonstrate the validity and results of the strategy on a limited set of items. This point will be discussed further in the concluding chapter.

The work presented here indicates the potential for using LLMs as proxies in combinatorial auctions. More broadly, the techniques presented in this work provide further evidence of the possibility of using LLMs for economic simulations and as economic agents. We might imagine a world where people and organizations all have AI proxies who represent them—where their proxies bid and negotiate on their behalf under economic mechanisms.

1.4 Related work

Related to our use of LLMs to simulate people participating in combinatorial auctions, there have been impressive studies demonstrating the ability of LLMs to mimic human-like responses. These studies will be highlighted in Section 1.4.1.

Though not on settings specific to combinatorial auctions, other studies describe LLMs as components of economic systems. These exciting studies, which we will describe in Section 1.4.2, imagine new roles for LLMs as actors in the economy and examine possible impacts. Similarly, in describing LLM-based proxies for participation in combinatorial auctions, we are imagining a new type of economic actor enabled by the unique strengths of LLMs.

1.4.1 LLMs to mimic human-like responses

LLMs have proven competent in simulating human responses to queries in many different applications. Horton [20] showed that LLM-pipelines can incorporate specified social preferences and political views in behavioral economics tasks, and give appropriate human-like responses. Park et al. [40] conducted 2-hour interviews with a thousand different people and then had each person complete the General Social Survey, the Big-Five Inventory, and participate in various games and tasks. They showed that an LLM-pipeline—given only the interview transcripts—could accurately predict corresponding human behavior over the surveys, game, and tasks. Argyle et al. [2] show that LLMs, conditioned on socio-demographic backstories and select survey responses, can generate additional survey responses that accurately match aggregate statistics as well as individual patterns within an individual's survey responses.

The breadth of results showing the ability of LLMs to simulate the response of humans to queries—here we take queries broadly to include survey responses, action selection in games and tasks, reports of utilities—demonstrate the general robustness of the LLMs' competency in mimicking human responses. These studies also give indication of techniques to get more human-like responses from LLMs. Studies find that LLMs give more human-like responses when they are prompted with vignettes—stories in natural language as opposed to plain numerical specifications—of a scenario[5], and when one uses the more advanced LLMs with larger parameter sizes and with additional refinement on response quality via a technique called reinforcement learning from human feedback (RLHF)[8].

Moreover, this ability of LLMs to simulate human response is enormously useful because we can use LLM-simulated people as subjects of simulations. Studies have shown such simulations to yield informative results.

Li et al.[27] showed that LLMs could be used as actors in macroeconomic simulations in a realistic manner—in the sense that they recovered from the simulations nominal macroeconomic indicators and regularities like the Phillips Curve and Okun's Law.

Zhao et al.[50] showed that LLMs could be used to simulate customers and restaurant owners in a manner that led to a "winner's take all phenomenon" among restaurants. They then modified various aspects of the LLM-simulated customers to show how those factors affected the phenomenon. Interestingly, due to the use of LLMs for the simulation, they were able to probe micro-level decisions made by the simulated customers.

Manning et al.[29] showed that LLMs can be used not only as simulated people within a economic system—systems as diverse as bargaining, hearings, hiring, and auctions—but also as an experimenter setting up the systems, endowing the simulated people with specific preferences and attributes. They go on to show that the results of running the system with LLM-simulated people lead to surprising results in the sense that the results would not have otherwise been predicted by an LLM before running the systems.

1.4.2 LLMs as components of economic systems

Work is also beginning to investigate the participation of LLMs as components of economic systems. These economic systems exhibit unique features due to their inclusion of LLMs. In our work, we will show that LLM-based proxies decrease the burden bidders face when participating in the combinatorial auctions. Related work focuses on many different features in many different domains, though not precisely on the topic we study. We organize the related work, roughly, into the following areas: collusion and collaboration in games, privacy in information markets, and augmentation of social choice.

Collusion and collaboration in games Fish et al. [12] showed that LLMs as participants in markets exhibited strategic behavior. They demonstrate that LLMs can collude to manipulate down the prices in two settings—LLMs setting prices as Bertrand oligopolists may collude to achieve higher prices, and LLMs bidding in first-price auctions may collude to achieve lower prices. Lin et al. [28] showed that LLMs as Cournot oligopolists in a complex multi-market setting collude to lower production quantities.

We see from these works the strategic capabilities of LLMs and that the effect of introducing LLMs as components of economic systems can potentially be detrimental to social welfare, in this case, via manipulated prices. Of course, it must be noted that collusion in one domain might translate to collaboration in another domain. One might imagine LLMs collaborating to stabilize energy consumption in brownout situations, or LLMs collaborating to help workers collectively bargain with an employer.

Privacy in information markets The use of LLMs can provide ingenious solutions to tricky problems.

In an information market, the buyer's inspection paradox is that the buyer needs to see the information that is for sale in order to calculate their value for the information. The paradox is that then, in a sense, the seller has already given the buyer the information. Rahaman et al.[46] describe a method where an LLM proxy—given the objective of a buyer—screens the information available for sale on behalf of the buyer by examining the content of the information, and then makes a purchasing decision on behalf of the buyer. In this manner, the bidder via their LLM proxy is able to examine and reason about their value for the information for sale without ever actually seeing the information themselves. They demonstrate that this method leads to better information purchases from the buyer.

Augmentation of social choice Social choice theory traditionally operates over a set of explicitly-defined alternatives and people's preferences over those alternatives. This raises a problem: when we consider all possible policies in the broadest sense, the socially-optimal policy may not exist in the set of explicitly-defined alternatives.

Fish et al.[13] showed that an LLM-based system—given human responses on a survey on their opinions on policy—is able to propose policy statements that lead to broad consensus. They do this by using an LLM to construct from each person's survey responses a model of their utility of any given policy. They then implement a system that consists of these LLMs which model utilities and another LLM which generates policy statements. They demonstrate how the system can be organized to effectively query these LLMs via an algorithm—with formal guarantees given a specific sense of fairness—in order to arrive at proposed policy statements, which they show to be empirically popular from human surveys.

The potential applicability of this kind of LLM-based system to more settings can be seen in the work of Gudiño et al.[17] who similarly showed the ability of LLMs to model voter responses to policy proposals in the Brazilian context. This suggests that a corresponding approach paired with a policy proposal generator LLM would be successful in proposing popular policy proposals.

We see from this example the theme of using LLMs to bridge human preferences defined in natural language with formal economic systems to achieve more efficient outcomes.

1.5 Roadmap

In this work, we describe a study on using LLM proxies in combinatorial auctions to reduce the time and effort required from bidders while maintaining auction efficiency. We evaluate the LLM proxies using simulated combinatorial auctions. We continue with Chapter 2 where we give a presentation of theory underlying the specific setting we are working in. Chapter 3 describes the testing sandbox the proxies are evaluated in and gives robustness checks for the sandbox. In Chapter 4, we formulate our LLM proxies and describe their performance in our simulation sandbox compared to our classical proxies. In Chapter 5, we describe and evaluate strategies to ensure both the simulation and the LLM-based proxies have complexity that is polynomial in the number of items. We end with concluding thoughts in Chapter 6.

2

Framework - Theory

This chapter describes the theory upon which we will implement the simulation framework and develop and evaluate LLM-based proxies. We begin with a brief survey of combinatorial auctions in Section 2.1. Section 2.2 gives our model of combinatorial auctions by describing the various components and how they interact. Section 2.3 gives a formal treatment of the XOR language. Section 2.4 gives a formal treatment of the Competitive Equilibrium Combinatorial Auction (CECA) and the classical proxies for it. Section 2.5 describes various metrics necessary for analyzing our simulations.

2.1 Brief survey of CA designs

We give a presentation of three standard CA designs [10,25] – the Vickrey-Clarke-Groves (VCG) auction, the Combinatorial Clock Auction (CCA) [3] and the Competitive Equilibrium Combinatorial Auction (CECA) [26] – to describe the considerations for the settings where our LLM-based proxies will act. A formal treatment of CECA, which is the specific auction mechanism that we will use for our simulations, will be given in Chapter 2.4. Each of these three auction mechanisms [10] has been shown to lead to efficient outcomes given *rational* bidder behavior, though an account of rational bidder behavior varies auction by auction.

The presentations of each CA design will focus on the communication required from the bidder during the auction and the difficulty for the bidder to produce such communication. Additional consideration will be given to the role of price discovery and privacy preservation, which are of broader interest in auction design. *Price discovery* is where the auction incrementally converges on prices to allow bidders to adjust to new market price information. *Privacy preservation* is where bidders only need to reveal some sub-part of their valuation.

Vickrey-Clarke-Groves (VCG) Auction

The VCG auction is a sealed-bid auction where bidders submit their valuations for all possible bundles of items. The auctioneer then allocates the items to maximize social welfare and charges each winner the opportunity cost of their winnings, which is the difference in the total welfare of all other bidders from the current allocation to the allocation where the current winner does not get allocated any items and the items are instead allocated according to their next best use among other bidders. This pricing scheme makes it a dominant strategy for bidders to report their values truthfully. Furthermore, using the truthful reports, the VCG auction reaches an efficient allocation of goods.

However, the VCG mechanism places a considerable burden on bidders. It requires them to determine and report their values all at once in a specified formula for all bundles they are interested in, without any price information to guide their valuation efforts. Additionally, privacy is compromised as bidders must reveal their comprehensive valuations.

Combinatorial Clock Auction (CCA)

The Combinatorial Clock Auction (CCA)[3] is an iterative auction that employs a clock phase for price discovery, followed by a sealed-bid proxy round to promote efficiency. During the clock phase, the auctioneer announces prices for individual items, and bidders indicate their desired quantities at those prices. Prices for items with excess demand are incrementally increased until the market clears. This provides valuable price information to bidders, allowing them to adjust their valuations and bidding strategies.

The proxy round takes place after the clock phase, using the market-clearing item prices from the clock phase as the starting point for package bidding. Bidders submit their valuations to proxy agents, who strategically bid on their behalf to maximize their profits. The inclusion of the clock phase reduces the burden on bidders by providing price discovery and allowing them to focus their valuation efforts on relevant packages. The use of the proxies in the subsequent proxy round reduces the time and cognitive demand for strategic bidding. The dominant strategy for the bidders using the proxies is simply to state their truthful preferences.

However, CCAs still require bidders to be able to give a comprehensive, truthful valuation for the sealed-bid proxy round.

Competitive Equilibrium Combinatorial Auction (CECA)

The Competitive Equilibrium Combinatorial Auction (CECA)[26] is an iterative auction format designed to achieve an efficient allocation of goods by converging to competitive equilibrium (CE) prices. In a CECA, the auctioneer announces prices over bundles of items and an allocation of items and asks bidders if they are happy with the allocation or if they would bid more; they continue doing this until every bidder indicates they are happy and thus a competitive equilibrium is reached.

Proxies help bidders determine how they should strategically respond to the auctioneer's announced prices and allocation and update their bid. Classical designs for proxies use a learning algorithm to decide what input from the bidder is needed to determine the relevant parts of their valuation and to decide how to update their bid.

By using the iterative format and the proxies to aid the bidders, CECA lowers the burden placed on bidders to participate in the auction and allows bidders to keep their valuations private until parts of their valuation become necessary.

We will be working with CECA, because its iterative format provides further opportunities to improve the classical proxies by preference elicitation using LLMs. As such, the CECA will be formally described in Section 2.4.

2.2 Model

In our model, we consider various *auction scenarios* over which auctions are run. A scenario $\mathcal{S}$ has N *items*: $\mathcal{T} = \{T_1, \ldots, T_N\}$. We have a population $\mathcal{P}$ of *people*, where $p \in \mathcal{P}$ is a specific person participating in the auction. A *bundle* b of items is a subset of $\mathcal{T}$, and $\mathcal{B}$ is the set of all possible bundles. A *price function* $\varphi : \mathcal{B} \to \mathbb{R}_+$ assigns a price to each bundle of items. Prices are considered to be personalized, where person p_i faces prices φ_i.

We assume that the objects of an auction scenario—the scenario itself, its items, bundles, prices,

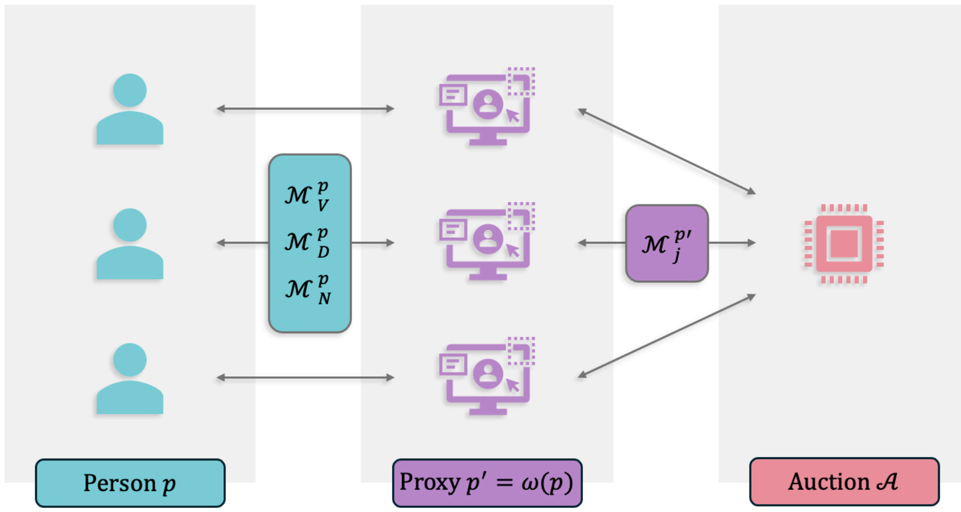

Figure 2.1: Diagram of the relevant components. There are three categories of actors: people, proxies, and the auction. Each of the actors are completely distinct from each other and can only communicate via messaging by calling $\mathcal{M}$. The proxies mediate the communication between the people and the auction. Proxies communicate with people by messaging them with value queries $\mathcal{M}_V^p$, demand queries $\mathcal{M}_D^p$, and natural language questions $\mathcal{M}_N^p$. Auctions communicate with proxies by messaging them with messages of type j by calling $\mathcal{M}_j^{p'}$ where the specific message type(s) depend on the auction design.

and people—have a natural language description given by the *description function* $\mathcal{D}$. For instance, we may have $\mathcal{D}(\mathcal{S})$ = "a farmer's market" and three types of items, $\mathcal{D}(T_1)$ = "apple", $\mathcal{D}(T_2)$ = "banana" and $\mathcal{D}(T_3)$ = "strawberry". In addition, we might have a person with a description given by $\mathcal{D}(p)$ = "Carlos; only likes red fruit" bidding in the auction. For expedience, we often directly refer to an object by its description.

People in a scenario participate in the auction via a proxy. We consider a family Ω of proxy designs. A person p is represented by a proxy $p' = \omega(p)$ with $\omega \in \Omega$. This proxy has an internal context, or 'memory', and so can refine its understanding of the person's preferences by repeatedly communicating with its person. Similarly, each person has a memory of the conversations between it and its proxy. A person communicates with its proxy by responding to natural language questions, value queries, and demand queries made by the proxy. We write $\mathcal{M}_N^p$ (Natural

Language Query) for the function that maps any natural language question to person p's answer. For example, we might have that for $q =$ Do you have specific colors of fruit you like or dislike? $\mathcal{M}_N^p(q) =$ Yes, I like red fruit.

Similarly, $\mathcal{M}_V^p$ (Value Query) maps any bundle to a (non-negative real) value that person p has for the bundle [*]. Finally, $\mathcal{M}_D^p$ (Demand Query) takes prices φ and a bundle b as input and returns either $(1, b)$ if the person is happy with b at φ or $(0, b')$ if the person would prefer some bundle $b' \neq b$ over b at φ. After each of these questions, the question and answer form part of the person and proxy's memories.

When evaluating our LLM proxy designs, we simulate people by secondary LLMs pipelines (distinct from the LLM proxy). These LLMs are seeded with the description $\mathcal{D}(p)$ of virtual people. In this simulated setting, we assume that the functions $\mathcal{M}_N^p$, $\mathcal{M}_V^p$, and $\mathcal{M}_D^p$ are implemented by the secondary LLM pipelines instead of by a person directly. [†]

Given a scenario $\mathcal{S}$ and K people $p_1, \ldots, p_K$ represented by their proxies $p_i' = \omega(p_i)$, for a given proxy design $\omega \in \Omega$, we define an auction $\mathcal{A}$ as an algorithm that takes as input $\mathcal{S}$ and $p_1', \ldots, p_K'$ and gives as output both an allocation $b_1, \ldots, b_K$ and payments $\phi_1, \ldots, \phi_K$, where the person p_i is given bundle b_i and must pay the auctioneer ϕ_i. The auction algorithm only communicates with proxies via messages $\mathcal{M}_j^{p'}$ where j is the type of message the auction algorithm is communicating with.

2.3 Bidding Language

Currently implicit in our discussion of combinatorial preferences and bidding in combinatorial auctions is how these preferences are communicated and what kind of *language* they are described

[*] The implementation of these message functions may not be deterministic.

[†] This can be formally captured by a context c_i which contains a transcript of the first i (natural language) communications. This is how the context is implemented in our LLM proxy design.

in. We use the term language to encompass different kinds of formula that may be used. In this section, we will introduce general considerations for such languages and introduce the commonly used bidding language for combinatorial auctions, XOR.

2.3.1 Considerations for bidding languages

Bidding languages are assessed on two criteria[34].

Succinctness

A bidding language's succinctness measures how efficiently it can represent a bidder's preferences. High succinctness enables bidders to convey complex preferences with minimal communication, which is valuable for reducing both time and computational demands, especially as the number of items grows. Succinctness is considered with respect to a specific preference class and it is not possible to succinctly express arbitrary combinatorial preference[‡].

Expressiveness

A bidding language's expressiveness reflects its capacity to capture a wide range of preferences. Two kinds of preferences to consider are preferences where items are *complements* and where items are *substitutes*. Items are complements when a bundle of items has a higher value than the individual items; an extreme case of which is given in the exposure problem. Items are substitutes when a bundle of items has lower value than the individual items; intuitively, this can happen when someone

[‡]Consider the construction of combinatorial preferences as follows. Start with a list of all $2^N - 1$ non-empty bundles with valuation equal to the number of items they contain. Independently for each bundle, add noise ε, where ε is a uniform random variable from $(-0.5, 0.5)$, to construct a new combinatorial preference. Because of the addition of the noise which makes it so that there is no pattern in valuations, we need to define the valuation in terms of at minimum $2^N - 1$ values so this preference cannot be succinctly expressed.

wants an item of a specific type but there are multiple varieties of that type available and they have the same value for one item of that type as they would for multiple.

2.3.2 XOR Language

A commonly-used language, which we will use for our simulations in the subsequent chapters, is the XOR language because it is fully expressive[35]—capable of expressing all possible preferences. An XOR bid consists of *atomic* bundles and the bids for each of the individual atomic bundles. An XOR bid has for any given bundle an implied bid equal to the highest bid of any atomic bundle the given bundle contains.

We can see how XOR bids are fully expressive: we can add atomic bundles for all 2^N possible bundles, such that we can independently control the valuation for any bundle. However, note that because expressing arbitrary preferences requires exponential complexity in the number of atomic bundles, the XOR language is not succinct for arbitrary preferences. The exponential complexity of the size of the XOR bid in terms of the number of items places prohibitive burdens on the bidder and the auctioneer[36]. Addressing this exponential complexity is a central challenge for combinatorial auction design.

Formally, an XOR bid $\theta = (B, v)$ consists of a set of atomic bundles $B \subseteq \mathcal{B}$ and valuation function $v_\theta : B \to \mathbb{R}_+$. The empty set $\emptyset$ is always an atomic bundle in B and $v(\emptyset) = 0$ because a person can always choose to bid on nothing for free.

For any bundle $b \in \mathcal{B}$, a bid θ induces a valuation function v^* whose domain is all bundles instead of just the atomic bundles. v^* values a bundle b as the highest value of any atomic bundle contained in b and is given by the following equation.

$$v^*_\theta(b) = \max_{\{b' \in B | b' \subseteq b\}} v(b') \tag{2.1}$$

This formulation captures the idea that the value of a bundle is determined by the highest-valued atomic bundle it contains. Recalling that a bid can represent a valuation or prices, the induced v^* captures the notion of *free disposal*, i.e. that the bid, price, or valuation of a bundle does not increase when an item in the bundle is removed. We will use φ_θ^* to refer to the prices that the induced valuation v^* of an XOR bid θ.

XOR-valuation query response

We model a simulated person with an XOR valuation $\theta = (B, v)$ as responding to queries in the following manner.

Value queries   When they are issued a value query, they are asked what value they would place on a specific bundle b. They would return in response $v_\theta^*(b)$ given by Equation 2.1.

Demand queries   When they are issued a demand query, they are asked whether they are happy with their allocated bundle b at given prices φ, and if not, which bundle they would like to switch to instead. Their response can be computed by first calculating the set of bundles B' that have higher utility than the allocated bundle.

$$B' = \{b' | v_\theta^*(b') - \varphi(b') > v_\theta(b) - \varphi(b); b' \in \mathcal{B}\} \tag{2.2}$$

If B' is non-empty, they give, as a response to the demand query the tuple $(0, b')$, where b' is a random element from B', because b' has higher utility than b.[§] Otherwise, they give, as a response to the demand query, the tuple $(1, b)$, indicating that they are happy with their allocated bundle.

[§] We choose a random element from B' to match common definitions of a demand query and the related equivalence query[26,1]. This can be thought of as the simulated person responding to a demand query by giving an example of a bundle they would prefer more at given prices than the allocated bundle.

Equivalence query When they are issued an equivalence query, they are asked whether their valuation is equivalent to a hypothesis valuation $\hat{\theta} = (\hat{B}, \hat{v})$. If $\hat{\theta} = \theta$, they return the tuple $(1, \emptyset)$, indicating that the hypothesis valuation is their valuation. If not, then they are not equivalent and they find a counterexample bundle b', by randomly selecting from the set where $v^*_{\hat{\theta}}(b') \neq v^*_{\theta}(b')$ in a similar fashion to the calculation of the response to demand queries, and they return the tuple $(0, b')$.

XOR preference elicitation

The XOR preference elicitation algorithm makes queries to a person to determine their XOR preferences $\theta = (B, v)$. We denote this algorithm as $\mathcal{L}_{xor}$ and formulate it from Lahaie & Parkes [26], who were inspired by the connection between XOR bids and monotone-DNFs identified by Blum et al. [6].

$\mathcal{L}_{xor}$, described in Algorithm 1, begins with an empty hypothesis valuation $\hat{v}$, containing an empty set of atomic bundles $\hat{B}$ with corresponding empty mapping $\hat{v}$. $\mathcal{L}_{xor}$ iteratively discovers new atomic bundles by checking if $\hat{B}$ is complete using either an equivalence query or a demand query with $\hat{\theta}$ as prices and the empty bundle b_0 as the allocated bundle.¶ If the response to the equivalence or demand query indicates that the current valuation correctly represents the person's preferences, the algorithm concludes that all atomic bundles have been identified and terminates. Otherwise, the algorithm invokes a subroutine, denoted $\mathcal{L}_{xor,step}$, to extract a new atomic bundle from the bundle b that is returned as a part of the response to an equivalence or demand query.

In the $\mathcal{L}_{xor,step}$ subroutine, given in Algorithm 2, the algorithm queries the bidder for the value of

¶Equivalence queries are used in the original learning algorithms from computational learning theory that the XOR preference elicitation algorithms are based on. Demand queries can also be used as given by Lahaie & Parkes [26], where they show that $\hat{B}$ is missing an atomic bundle b', then the bidder would indicate in the response to the demand query that they would desire the bundle b' as b' would give them positive utility while all $b \in B$ would give them zero utility when facing $\hat{\theta}$ as the prices.

Algorithm 1 Learn XOR Preferences ($\mathcal{L}_{xor}(p)$)

Input: Person p
Initialize $B \leftarrow \emptyset$, $v \leftarrow \emptyset$, equivalent $\leftarrow$ **False**
while not equivalent **do**
  $(r, b) \leftarrow \mathcal{M}_D^p(\varphi^*_{(B,v)}, b)$ {Obtain response and bundle from demand query}
  if $r = 1$ **then**
    equivalent $\leftarrow$ **True**
  end if
  if $r = 0$ **then**
    $(b', \text{value}) \leftarrow \mathcal{L}_{xor,step}(p, b))$ {Identify atomic bundle and its value}
    $B \leftarrow B \cup \{b'\}$
    $v(b') \leftarrow$ value
  end if
end while
return (B, v)

the bundle b using $\mathcal{M}_V^p(b)$ and then attempts to identify which items are essential by checking for each item in b whether removing it to create smaller bundle b'' changes the valuation, i.e. whether $\mathcal{M}_V^p(b) = \mathcal{M}_V^p(b'')$. If removing an item does not change the valuation, it implies that the item is not essential to the bundle's value and can be excluded from the atomic bundle. This process is repeated for all items in the bundle to identify the minimal atomic bundle b', which is then added to the set B along with updating the valuation function to the queried value of b.

This iterative process continues until all atomic bundles and their corresponding valuations are discovered, ensuring that the algorithm accurately learns the person's XOR preferences with a bounded number of queries.

2.4 Competitive Equilibrium Combinatorial Auction (CECA)

The competitive equilibrium combinatorial auction (CECA), described by Lahaie and Parkes[26] and first introduced here in Section 2.1 is characterized by its auction algorithm denoted $\mathcal{L}_{CECA}$,

Algorithm 2 Learn XOR Preferences Step ($\mathcal{L}_{xor,step}(p, b)$)

Input: Person p, Bundle b {Current bundle from demand query}
$v \leftarrow \mathcal{M}_V^p(b)$ {Obtain original bundle value}
$b' \leftarrow b$ {Initialize refined bundle}
for $i = 1$ to N **do**
  if $T_i \in b'$ **then**
    $b'' \leftarrow b' \setminus \{T_i\}$ {Remove item i from bundle}
    $v' \leftarrow \mathcal{M}_V^p(b'')$ {Query new bundle value}
    if $v' = v$ **then**
      $b' \leftarrow b''$ {Item i is not essential}
    end if
  end if
end for
return (b', v) {Return identified atomic bundle and its value}

which is given in Algorithm 3. The auction algorithm progresses through iterations—repeated steps where the auctioneer announces an allocation of items and prices, and the proxies adjust bids on behalf of the bidders they represent. This process continues until all bidders are *happy* and thus in a *competitive equilibrium*; that is, they indicate via their proxies that they no longer wish to change or increase their bids.

We call the bid that the proxy $\omega(p_i)$ submits at each iteration the manifest valuation $\tilde{\theta}_i$. The general form of the CECA considers different languages for representing $\tilde{\theta}_i$, and we will consider the specific case where $\tilde{\theta}_i$ is represented in the XOR language. The problem of computing the optimal allocation from the manifest valuation is also called the *winner determination problem* because in computing the allocation of items we are determining who wins what items. We describe in Section 2.4.1 how the auctioneer uses the manifest valuations to compute the optimal allocation of items by using a technique called *integer programming*. We describe in Section 2.4.2 how the auctioneer uses the manifest valuations to compute the prices that the bidders face, and the prices will be constructed in a way that makes them *Lindahl prices* which means that a competitive equilibrium will be reached. We describe in Section 2.4.3 the classic design for a proxy that participates in the competitive equilibrium combinatorial auction.

Algorithm 3 Competitive Equilibrium Combinatorial Auction ($\mathcal{L}_{CECA}$)

Input:
  $\mathcal{S}$: Scenario
  $p'_1, p'_2, \ldots, p'_K$: Proxies for each person p_i
Initialize:
  Set manifest valuations $\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_K \leftarrow (\emptyset, \emptyset \to 0)$ {Initial guess, e.g., all zeros}
  Set counterexample_flag $\leftarrow$ False
while not counterexample_flag **do**
    $b_1, \ldots, b_K \leftarrow$ allocate$(\mathcal{S}, \tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_K)$ {Allocate bundles to maximize social welfare via integer programming}
    counterexample_flag $\leftarrow$ False
  for each proxy $i = 1$ to K **do**
      $\varphi_i(b_i) \leftarrow$ Compute maximal Lindahl prices(φ, b_i) {Per Equation 2.5}
      $(r, \theta') \leftarrow \mathcal{M}_C^{p'_i}(\varphi_i, b_i)$ {Check if happy}
    if $r = 0$ **then**
        $\theta_i \leftarrow \theta'$
        counterexample_flag $\leftarrow$ True {Need to reallocate}
    end if
  end for
end while
return $\{(b_i, \varphi_i(b_i)) \mid i = 1, 2, \ldots, K\}$ {Allocated bundles and corresponding payment}

2.4.1 Winner determination with integer programming

Solving the winner determination problem involves finding the optimal allocation of bundles $b^* = \{b_1, \ldots, b_K\}$ that maximizes social welfare, given the submitted manifest valuations $\tilde{\theta}_1, \ldots, \tilde{\theta}_K$. Formally, this can be expressed as:

$$b^* = \arg\max_b \sum_{i=1}^{K} v^*_{\tilde{\theta}_i}(b_i), \tag{2.3}$$

where $v^*_{\tilde{\theta}_i}(b_i)$ denotes the induced XOR valuation function for one with XOR valuation $\tilde{\theta}_i$, and where b^* represents the optimal allocation of items. By choosing the allocation b^* according to Equation 2.3, we maximize the social welfare under truthful manifest valuation $\tilde{\theta}_i$.

We can solve for b^* by using integer programming, which is a technique where we write a *program* that consists of variables that we are optimizing the values of, a linear expression that we are trying to maximize, and linear inequalities that are conditions that must be satisfied. Then, an integer program solver is used to calculate the values of the maximizing variables. Efficient solvers are readily available, such as the OR-TOOLS package[44], which we'll use.

We translate the winner determination problem into an integer program which is given in Equations 2.4.

The program is given N items, $\{T_1, \ldots, T_N\}$, to allocate among K people who each have XOR valuations $\theta_i = (B_i, v_i)$ where their valuation has $m_i = |B_i|$ atomic bundles, $B_i = \{b_{i,1}, \ldots, b_{i,m_i}\}$. The variables we are optimizing are $x_{i,j} \in \{0, 1\}$, where if $x_{i,j}$ is 1, person i is allocated their atomic bundle $b_{i,j}$ and if $x_{i,j}$ is 0, person i is not allocated that bundle (Equation 2.4d). Defined as such, $x_{i,j}$ characterizes the allocation and we want to choose $x_{i,j}$ in a way that maximizes social welfare (Equation 2.4a). We impose two conditions on the allocation. First, as XOR valuations are defined, only one atomic bid can be accepted per person so we write that as a condition in Equation 2.4b. Second, each item can be allocated only once, so we write that as a condition in Equation 2.4c, where $a_{i,j,k}$ is 1 if item T_k is in atomic bundle $b_{i,j}$ in person i's XOR valuation, and as a result, $a_{i,j,k} \cdot x_{i,j}$ is 1 if and only if item k is allocated to person i via their atomic bundle $b_{i,j}$.

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{i=1}^{K} \sum_{j=1}^{m_i} v_i(b_j) \cdot x_{i,j} && && \text{(2.4a)} \\
\text{such that} \quad & \sum_{j=1}^{m_i} x_{i,j} \le 1 && \forall i \in \{1, \ldots, K\} && \text{(2.4b)} \\
& \sum_{i=1}^{K} \sum_{j=1}^{m_i} a_{i,j,k} \cdot x_{i,j} \le 1 && \forall k \in \{1, \ldots, N\} && \text{(2.4c)} \\
\text{where} \quad & x_{i,j} \in \{0, 1\} && \forall i \in 1, \ldots, K; \forall j \in 1, \ldots, m_i && \text{(2.4d)}
\end{aligned}
$$

We compute b^* by feeding the program given by Equations 2.4 into the SCIP integer program solver provided in the OR-TOOLS package[44].

The winner determination problem is solved for the manifest valuations at the beginning of each iteration at the auction. The winner determination problem's solution b^*, which the auctioneer announces as the allocation, is called the *intermediate allocation* for that iteration.

2.4.2 Competitive equilibrium with Lindahl prices

Having described the computation of the optimal allocation $b^* = \{b_1, \ldots, b_K\}$ in the previous Section 2.4.1, in this section, we describe computation of prices $\varphi^* = \{\varphi_1, \ldots, \varphi_K\}$ such that the allocation and corresponding prices has the property that neither the person or the auctioneer will want to modify the allocation by changing which items are sold to which person. This property, called *competitive equilibrium* is desired because we want the population of people participating in the auction and the auctioneer to agree on the allocation b^*—we do not want a participant to look at the prices and decided that they would like to switch to buying some other bundle and we do not want the auctioneer to look at the prices and decide they would like to instead allocate items based on some alternate allocation—because b^* is the allocation that maximizes social-welfare and so we want to realize that outcome b^*.

When prices φ^* and allocation b^* together form a competitive equilibrium, prices φ^* are said to be *Lindahl prices* that *support* the allocation b^*.

We compute prices $\varphi^* = \{\varphi_1, \ldots, \varphi_K\}$ that support the optimal allocation b^* with Equation 2.5, where the price of bundle b for person i is $\varphi_i(b)$ and is given by their manifest valuation implied value $v^*_{\tilde{\theta}_i}$ for that bundle minus a constant d_i.

$$\varphi_i(b) = v^*_{\tilde{\theta}_i}(b) - d_i \tag{2.5}$$

We now give brief justification for the fact that neither the auctioneer nor people participating in the auction would want to change the overall allocation b^* or their allocated bundle b_i, respectively.

For the auctioneer, we can calculate their revenue as $\sum_{i=1}^{K} \varphi(b_i) = \sum_{i=1}^{K} v^*_{\tilde{\theta}_i}(b_i) - \sum_{i=1}^{K} d_i$. We see that $\sum_{i=1}^{K} d_i$ is constant and we know from the integer program given by Equations 2.4 that b^* is the allocation that maximizes $\sum_{i=1}^{K} v^*_{\tilde{\theta}_i}(b_i)$. Thus b^* is the allocation that maximizes the auctioneer's revenue, so the auctioneer will not want to choose some alternate allocation.

For the people participating, with truthful manifest valuations $\tilde{\theta}_i$, we see that each person derives constant utility d_i from any bundle b as $u_i(b) = v^*_{\tilde{\theta}_i}(b) - \varphi_i(b) = d_i$, so they will be indifferent towards switching to any bundle. In this chapter, in considering the properties of the CECA auction we will assume honest responses both by the proxies and by the participants of the auction in keeping with convention with computational learning theory and prior work on CECA. When this assumption is not possible, as will be the case in subsequent chapters with LLM-based proxies and participants whose response are not deterministic and who may hallucinate, we focus instead on the empirically realized social-welfare and auction efficiency.

In our implementation of the CECA auction, for simplicity, we will be using *maximal* Lindahl prices, where $d_i = 0$. Though other valid Lindahl prices exist, we are primarily interested in understanding the effect of LLM-proxies on the relationship between auction efficiency and the number of interactions between the proxy and the person they represent. Therefore, under truthful bidding, though the choice of Lindahl prices effects the relative utility between the auctioneer and the participants, it does not affect the overall social welfare and auction efficiency.

2.4.3 Classical proxy

For the CECA auction, we have classical proxies who elicit preferences from the people they represent using algorithms inspired by computational learning theory. Specifically, when manifest valuations are reported in the XOR bidding language, the classical proxy uses the XOR preference elicitation algorithm given in Section 2.3.2 to decide how to elicit preferences from the person they represent. This algorithm is inspired by monotone-DNF learning algorithms.

The *XOR classical proxy*, ω_{xor}, implements for $p' = \omega_{xor}(p)$, $\mathcal{M}_C^{p'}$ similar to Algorithm 1, $\mathcal{L}_{xor}$. p' maintains in its memory a hypothesis valuation $\hat{\theta} = (\hat{B}, \hat{v})$, with $\hat{B}$ initialized to the set containing only the empty set and $\hat{v}$ maps the $\emptyset$ to 0. The proxy implements $\mathcal{M}_C^{p'}(\varphi_i, b_i)$ as the function that returns the result of the following algorithm: call a demand query $\mathcal{M}_D^{p}(\varphi_i, b_i)$, and depending on the result, if the person is happy, return $(1, \hat{\theta})$, and if the person is not happy and prefers some bundle b', update the hypothesis valuation $\hat{\theta}$ with a new atomic bundle and its corresponding valuation, which is computed by running the algorithm $\mathcal{L}_{xor,step}(p, b')$, and returns the update hypothesis valuation in the tuple, $(0, \hat{\theta})$.

2.5 Metrics

From the simulations, we will be primarily interested in how—for different proxy designs—the overall auction efficiency corresponds with the number of interactions the person has with their proxy. For robustness, we will be aggregating over multiple runs of the testing sandbox over many auction scenarios with different scenarios and different populations of people. In defining our metrics, we will also need to consider two facts: different people may have different amounts of interactions with their proxy both overall and within each iteration of the auction, and value queries $\mathcal{M}_V^p$ may not be deterministic.

Our metrics will be calculated from *records* of runs of the auction. Each sequence of records R consists of a sequence of tuples $r_j = (n_j, v_j)$ where r_j is the record for the jth iteration of the auction, n_j is the average total number of interactions between a proxy and the person they represent from the 1st to the jth iteration of the auction, inclusive, and v_j is the sum of the valuations each person p_i has for the bundle $b_{i,j}$ they are assigned in the intermediate allocation of the jth iteration, which is calculated as $v_j = \sum_i v^*_{\theta_i}(b_{i,j})$. For our simulations, each person p_i will have a defined ground-truth XOR valuation θ_i.

2.5.1 Auction Value at Number of Interactions

For a given run of an auction with records R, the *Auction Value at n interactions* (AV_n) represents the total auction value at the iteration of the auction where the average number of interactions per person is closest to n without exceeding it. Formally, we define it as follows.

Given a record $R = \{r_1, r_2, \ldots, r_J\}$ where each $r_j = (n_j, v_j)$,

$$AV_n = v_{j^*} \tag{2.6}$$

where j^* is the highest iteration that has an average number of interactions that is less than or equal to n, satisfying the following.

$$n_{j^*} = \max\{n_j \mid n_j \leq n, \forall j \in \{1, \ldots, J\}\} \tag{2.7}$$

2.5.2 Auction Efficiency

The *Auction Efficiency* (AE_n) at n interactions measures how effectively the auction converts interactions into total auction value relative to the maximum possible auction value achievable in the scenario.

Given AV_n and the optimal auction value ν,

$$AE_n = \frac{AV_n}{\nu} \tag{2.8}$$

ν represents the maximum possible total valuation that could be achieved if the auction allocated bundles optimally based on true valuations of all participants.

For our simulations each person p_i will have a defined ground-truth XOR valuation θ_i. Therefore, the social-welfare maximizing total value (ν) for a scenario $\mathcal{S}$ with population $\mathcal{P} = \{p_1, p_2, \ldots, p_K\}$ can be calculated by first solving for the social-welfare maximizing allocation by using the integer program for solving the winner determination problem (Section 2.4.1) using the ground-truth valuation $\{\theta_1, \theta_2, \ldots, \theta_K\}$ and then summing the individual values each person holds for their allocated bundle according to their ground-truth valuations.

2.5.3 Evaluating proxies

We define a benchmark B as a set of *setups* where a setup is a tuple of scenarios and populations, $(\mathcal{S}_i, \mathcal{P}_i)$. Thus, a proxy's efficiency at n interactions, PE_n, is the average auction efficiency across all setups in the benchmark.

3
Framework - Simulation

Here we describe the simulation sandbox and various checks performed on the sandbox to provide support for its further use.

3.1 Making LLM calls

When we describe making a call to an LLM, the following three steps are performed:

1. *Collate* the relevant information into a prompt

2. *Call* an LLM with the prompt. The prompts are then passed to OpenAI's GPT-4o-mini model[38], versioned July 18, 2024. Unless otherwise specified, all LLM calls in this work use this model.

3. *Compile* the response from the output of the LLM. The responses are either extracted using regular expressions from the LLM output or using the structured outputs feature provided for GPT-4o-mini model[38] which allows for the specification of the response format. If the responses cannot be identified as such or the LLM otherwise encounters errors in some way, we repeatedly call the LLM with the same prompt until it succeeds.

3.2 Scenarios

We create three scenarios in which we test the LLM pipelines for modeling bidders and test the LLM pipelines for acting as proxies in CECAs. We design the scenarios to be accessible so that the reader can readily assess the realism of the interactions and so that various LLMs can be tested without domain-specific knowledge or fine-tuning. Each scenario has $N = 6$ items for auction, where some of the items are clear substitutes for each other and others are clear complements to each other, and many are somewhere in between. These scenarios and their items are presented in Table 3.1.

3.3 Bidders

3.3.1 Seed generation pipeline

Recall that a seed of person p, denoted as $\mathcal{D}(p)$, is the natural language description that defines p's preferences.

Table 3.1: Scenarios and Available Items

Scenario	Items	Description
$\mathcal{S}_1$: **Electronics**	1. Apple AirPods (2nd Generation) 2. Apple AirPods Max 3. Apple iPad (9th Generation) 4. Apple iPad Air (M2) 5. Apple Pencil (2nd Generation) 6. Apple Pencil Pro	A collection of like-new electronics donated to a local library, ranging from AirPods to iPads and accessories.
$\mathcal{S}_2$: **Preserves**	1. Organic Strawberry Jam 2. Wild Blueberry Preserves 3. Apricot and Lavender Conserve 4. Sugar-Free Raspberry Spread 5. Spiced Plum Chutney 6. Tropical Mango and Passionfruit Jam	Gourmet food items are available, including exotic jams and chutneys with unique flavors, perfect for cooking or gourmet meals.
$\mathcal{S}_3$: **Transportation**	1. Electric Scooter S2 Pro (2024) 2. Electric Scooter Elettrica (2023) 3. Voltron SP03 Electric Scooter (2024) 4. Troik Verve+ 2 (2023) 5. Titan Escape 3 (2023) 6. Schwin Suburban (2021)	A variety of electric scooters and bikes are up for auction, catering to urban commuters and fitness enthusiasts.

The process for generating the seeds is carried out in a four-step LLM pipeline.

The first step of the pipeline generates the germ of the seed. Each subsequent step revises the seed to be more specific and more coherent. We take the final revised preferences at the end of the fourth step to be the true seed, $\mathcal{D}(p)$.

Step 1: Initial Preference Generation. The LLM generates a person's specific purchase occasion and taste over the items, which forms an initial description of a person's preferences based on the given auction scenario. To encourage diversity in responses, we ask the LLM to generate three possible versions and then randomly sample one.

Step 2: Revision for Clarity and Consistency. The LLM revises the initial preferences to enhance clarity and internal consistency in describing the person's preferences.

Step 3: Anchor with Specific Valuations. The LLM anchors the person's preferences by assigning explicit dollar values to specific bundles of interest.

Step 4: Refinement for Precision in Complex Bundles. The LLM refines the person's preferences to outline a process that defines how they evaluate other unspecified bundles, considering items as substitutes and complements.

3.3.2 Message response pipeline

Using the seeds for the simulated person, $\mathcal{D}(p)$, we create our *ground-truth simulated person*, which we will use to evaluate and characterize various LLM-based proxies in Chapter 4. We model each simulated person as having ground-truth XOR preferences θ.

To construct θ, we implement an LLM bundle-valuation subroutine that gives the value a simulated person has for a bundle. We use this to get explicit valuations v of all possible bundles $\mathcal{B}$, so that the ground-truth simulated person has *dense preferences* $\theta = (\mathcal{B}, v)$ over all possible bundles. Construction of the ground-truth simulated person will thus use a number of LLM calls with complexity exponential in the number of items due to calling LLM valuation subroutine for all 2^N possible bundles.

θ is used to directly implement to respond to value queries and demand queries, $\mathcal{M}_V^p$ and $\mathcal{M}_D^p$ respectively, in accordance with the equations given in Section 2.3.2. We define an additional LLM subroutine to respond to questions to implement the message response function for natural language questions, $\mathcal{M}_N^p$. For these three queries, the simulated person does not maintain a memory of previous responses. They will always give consistent responses to value and demand queries as $\mathcal{M}_D^p$ and $\mathcal{M}_V^p$ are deterministic given θ. Our proxy designs will only ask a maximum of one question at the start of the auction, so memory for $\mathcal{M}_N^p$ is not relevant.

In Chapter 5, we will remove the exponential complexity associated with the construction of the simulated person. Instead, we will create a simulated person with *sparse preferences*, i.e. preferences over a non-exponential-complexity number of bundles, that approximates the ground-truth simulated person. Because equivalence queries will only be used in the construction of the simulated person with sparse preferences, a description of the message response function to equivalence queries will be deferred to Chapter 5.

LLM subroutines

Bundle-valuation subroutine This subroutine takes as input the description of a person's preferences, $\mathcal{D}(p)$, and the description of a bundle b, $\mathcal{D}(b)$, and gives the value the person has for the bundle, by issuing a LLM call that reasons over the description of the person's preferences and the description of bundle's contents, according to the following prompt.

```
Value query prompt:

Description of scenario: D(S)
Description of the person: D(p)
*****
The person has the option to receive bundle: D(b)
*****
Please determine the value the person assigns to this bundle.
Provide the maximum dollar amount they are willing to pay
in the format.
```

Question-answering subroutine This subroutine takes as input the description of a person's preferences, $\mathcal{D}(p)$, and a question q, and gives the answer the person would give for the question, by issuing a LLM call that reasons over the descriptions of the person's preferences and the question asked, according to the following prompt.

```
Question prompt:

Description of scenario: 𝒟(𝒮)
Description of person: 𝒟(p)
*****
The person is asked the following question: q
*****
Please reason through and craft how you think the
person would respond to the question.
```

3.3.3 Pipeline checks

We now describe various checks on the pipelines used to construct the ground-truth simulated person for the purposes of demonstrating the validity of using them to evaluate proxies.

Worked Examples Below are examples of runs of the LLM pipelines. For brevity, we truncate the seeds and the bundle-valuation subroutine calls to the crucial text. The full versions are available in Appendix A.

Example Seed Below is an example of a generated seed.

Cecilia is focused on acquiring the Apple Pencil Pro (code: APPLEPENCILPRO) for her digital art. She places a high value on tools that enhance her creative capabilities, specifically seeking industry-leading performance and precision for her artistic projects. The dollar value for the Apple Pencil Pro she is willing to pay is estimated at $120, as this aligns with retail market prices for similar tools.

Cecilia is open to considering additional items that can complement the Apple Pencil Pro and significantly improve her workflow. Among the other available items, she may find the following combinations valuable, each associated with an estimated dollar value based on market rates:

1. **Single Item Bundle:**
   - Apple Pencil Pro (code: APPLEPENCILPRO) - $120

2. **Complementary Bundles:**
   - iPad Air 11-inch (M2) with Apple Pencil Pro:
     - iPad Air (code: IPAD12) - $600
     - Apple Pencil Pro (code: APPLEPENCILPRO) - $120
     - Total Value: $720
   - iPad (9th Generation) with Apple Pencil Pro:
     - iPad (code: IPAD9) - $350
     - Apple Pencil Pro (code: APPLEPENCILPRO) - $120

```
     - Total Value: $470

Cecilia considers these other items because they can enhance her digital art
    experience, allowing her to utilize the Apple Pencil Pro more effectively.
    The iPad models provide her the necessary platform for creative applications
    , which is a key factor in her evaluation.

However, Cecilia prioritizes the Apple Pencil Pro as her primary target and
    feels that any additional item should not just be an upgrade but must
    genuinely elevate her creative process. She is likely to discount bundles
    that include items she believes do not contribute directly to her artistic
    workflow, such as the Apple AirPods or AirPods Max, as they do not enhance
    her art-making activities.
```

Example Bundle-Valuation Subroutine Call Below is an example of the LLM's response in a bundle-valuation subroutine call issued for the ELECTRONICS scenario, for the simulated bidder Cecilia described in the example seed, and for the bundle consisting of an iPad Air (M2) and an Apple Pencil (2nd Generation). The LLM reasons through the situation and then gives what the bidder values the bundle at.

```
Reasoning:
   - The iPad Air (IPAD12) has a baseline value of $600.
   - The Apple Pencil (2nd Generation) generally might not add as much value as
       the Pro version, which Cecilia values at $120; however, since it is still
```

```
      a creative tool, she won’t discount it too heavily but will assign it a
     lower value (let’s estimate around $80 instead of the Pro’s $120).
   - The value of the proposed bundle can be assessed as follows:
     - iPad Air: $600
     - Apple Pencil (2nd Generation): $80
     - Combined Value: $600 + $80 = $680

Since the Apple Pencil (2nd Generation) is not as substantial an upgrade in her
    creative workflow compared to the Pro version, we may apply a slight
    adjustment downward, let’s say a 10\% discount for it not being the Pro
    model:
Final Estimated Value = $680 - ($680 * 0.10) = $680 - $68 = $612.

In conclusion, the estimated value of the proposed bundle is:

Bundle value: $612
```

Example Answers to Questions Below are a question and answer for each of the three scenarios. The answer is given by the question-answering subroutine for a specific seed. The first seed is the example seed under the ELECTRONICS scenario.

```
ELECTRONICS scenario (example seed)
Q: What specific types of electronics are you interested in purchasing, and what
     is your maximum bid for those items?
```

```
A: I am primarily interested in purchasing the Apple Pencil Pro, with a maximum
    bid of $120, and I would also consider bundles that include an iPad if they
    enhance my digital art experience.

PRESERVES scenario
Q: What types of gourmet preserves do you prefer (e.g., fruit, savory, unique
    flavors), and what is your maximum bid amount for a single jar or a bundle
    of jars?
A: I prefer health-conscious fruit preserves with low sugar options,
    specifically the Sugar-Free Raspberry Spread and Organic Strawberry Jam, and
     my maximum bid for a single jar is $10, while for a bundle including both,
    I would consider $17.

TRANSPORTATION scenario
Q: How much would you be willing to spend on all three scooters and all three
    bikes, and do you prefer one over the other?
A: I am only interested in the Schwin Suburban and the Titan Escape 3, so I
    would be willing to spend around $850 total, but I do not consider the
    scooters at all.
```

Accuracy of pipelines We found that the seeds generated by the seed-generation pipeline robustly describe a specific valuation. To see this, we examined the inter-model robustness of the seeds. We compared the valuations generated by the message response LLM pipeline using the GPT-4o mini model[38] to those generated with the Gemini-1.5-flash model[16]. We found that the valuations were fairly consistent across models as shown in Figure 3.1. Because the models are provided by separate teams from OpenAI and Google, respectively, this shows that the seeds and the valuations interpreted from it via the message-response pipeline are not specific to a specific family of models and thus to some extent, generally interpretable and robust.

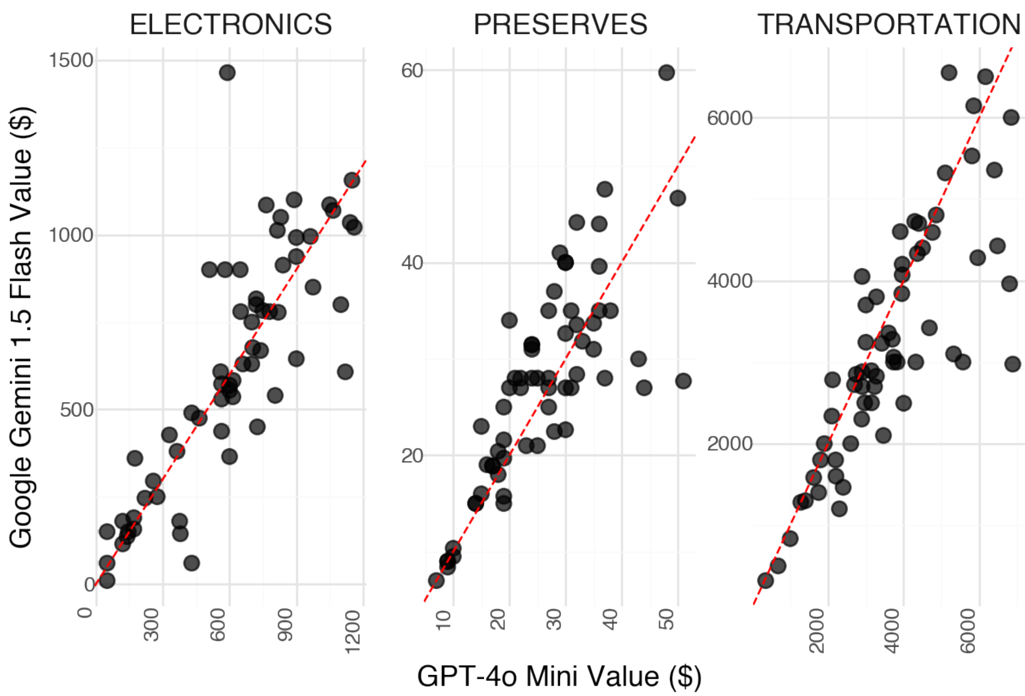

Figure 3.1: Comparison between GPT-based and Gemini-based models in bidder robustness tests. GPT-4o-mini and Gemini-1.5-flash give similar values in response to value queries evidenced by the dots, each of which representing a single bundle, falling near the identity GPT-4o-mini value ($) = Gemini-1.5-flash value line ($).

Precision of pipelines We also ran a set of experiments to understand the shape of valuations implied by the seed.

For a specific seed in each scenario, we ran a sequence of steps. At each step, we ran the message response LLM pipeline for value queries ten times, varying the bundle for each step. For the first step, we used a bundle of one item and we continued for each subsequent step by incrementally adding items to the bundle, and then upon reaching a bundle of all the items, we incrementally removing items from the bundle. The results are shown in Figure 3.2.

The results display sensible characteristics.

First, the preferences are monotonic or, equivalently put, they satisfy free disposal. We can see in Figure 3.2 that, as we add items to the bundle, the valuation of that bundle consistently goes up and, similarly, as we remove items from the bundle the valuation of the bundle consistently goes down. For smaller bundles—those that have zero, one, two or three bundles, numbered in the figure as Bundle 1, 2, 3, 9, 10, 11, 12—we can see that the boxes, indicating 25th and 75th percentile of the valuations of the bundles, do not even overlap.

Second, valuations are more precise for smaller bundles—as seen by the narrower boxes in the figure—which are less complex than other larger bundles. This is sensible as it seems easier to define the value of a pair of AirPods or an Apple Pencil plus an iPad versus the value of two iPads and two Apple Pencils. Reasonably, people (including our simulated people) might have less precise valuations over more convoluted combinations. We should note that in our scenarios, bundle size directly corresponds to how complicated the bundles are. Generally though, this may not be the case. For example, reserving time slots for a contiguous four hours consisting of eight half-hour chunks would be much less complicated than three random half-hour chunks throughout the day.

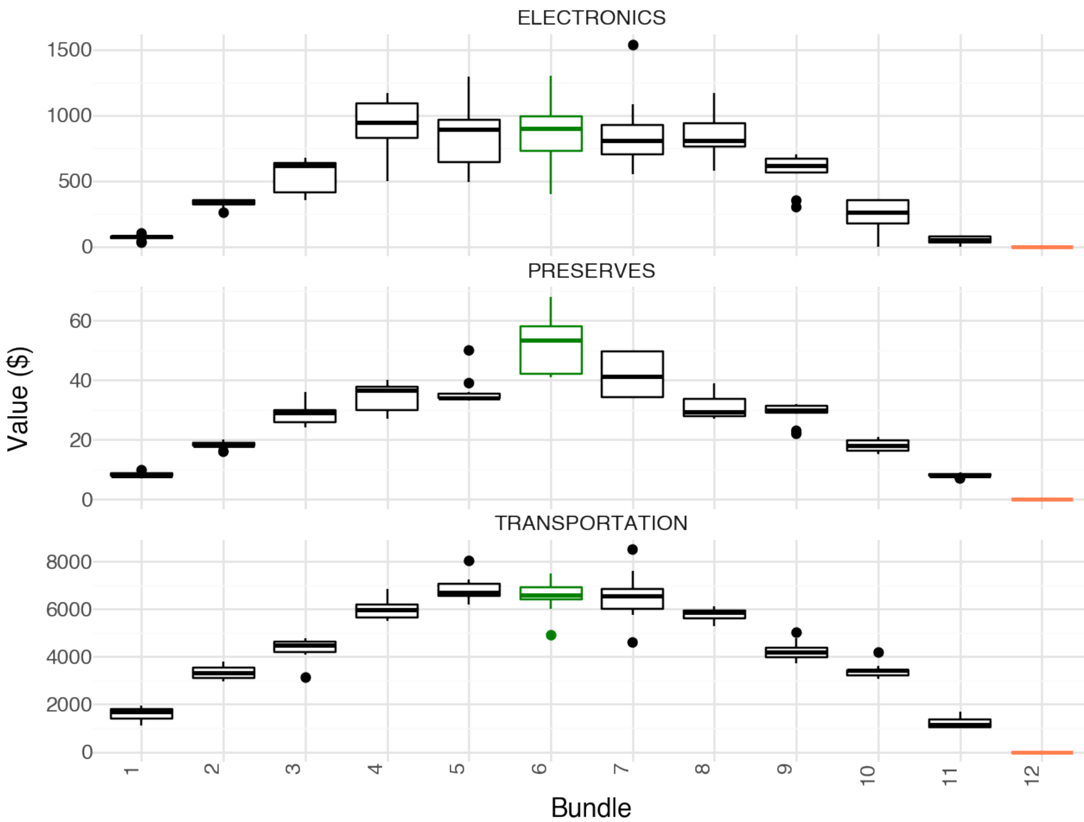

Figure 3.2: Preference variability and shape across repeated value queries to a single person. Each boxplot displays the distribution of values for a single bundle ($n = 10$). Bundle 6, in green, has all 6 items. Bundle 12, in orange, has no items. Bundle $i \leq 6$ has items $T_1, \ldots, T_i$. Bundle $6 < i \leq 12$ has items $T_i, \ldots, T_6$.

4

Constrained scenarios

We implement various proxy designs that incorporate LLMs into proxies by implementing for each proxy design ω the CECA-step function, denoted by the subscript C as $\mathcal{M}_C^{\omega(p)}$, which is required for the proxy to participate in a CECA in our testing sandbox. A CECA-step message response function takes as input the announced prices φ and allocates bundle b for person p and gives the proxy $\omega(p)$'s response to the announced prices and allocation on behalf of the person.

For each of our LLM-based proxy designs or our classical proxy design, ω_{XOR}, we simulate the results of the auctions when we use one of the proxy designs to represent each person in the population for each of the setups in a benchmark B_1, the creation of which will be described in the following section.

4.1 Benchmark creation (B_1)

We use the LLM-based seed generation and message response pipelines as defined in Section 3.3 to construct a benchmark B_1 encompassing the three scenarios: ELECTRONICS, PRESERVES, and TRANSPORTATION. B_1 consists of nine setups, three setups each for the three scenarios described in Table 3.1, where $B_1 = \{\beta_{i,j} | i \in \{1, 2, 3\}, j \in \{1, 2, 3\}\}$, $\beta_{i,j}$ is the setup defined by the tuple $(\mathcal{S}_i, \mathcal{P}_{i,j})$, and $\mathcal{P}_{i,j}$ is a population of three simulated people generated to participate in $\mathcal{S}_i$, and each simulated person has ground-truth XOR valuation $\theta_{i,j,k}$.

B_1 will be used to evaluate and compare different proxy designs.

4.2 Pure-LLM proxies

We first consider *pure-LLM proxies* which exclusively use LLMs to make decisions on submitting manifest valuations to the auction (via $\mathcal{M}_C^{\omega(p)}$) and on communicating with the bidder.

4.2.1 Design of pure-LLM proxies

We consider two kinds of pure-LLM proxies. *Drop-in LLM proxies* are those that only use value and demand queries when messaging their person. They are drop-in replacements for proper-learning proxies like ω_{XOR} because they message their person with the same message types as ω_{XOR}: value queries and demand queries. *Plus-questions LLM proxies* use natural language questions in addition to value queries and demand queries when messaging the person they represent.

4.2.2 Drop-in LLM proxy

We give two drop-in LLM proxy designs.

Design 1 $\omega_{VD,1}$ maintains in its memory a transcript of the conversation, c_i (with i being the given iteration of the auction), and a hypothesis XOR valuation $\theta = (B, v)$. For person p and its proxy $p' = \omega_{VD,1}(p)$, the algorithm implements $\mathcal{M}_C^{p'}$ as the algorithm, which when called with prices φ and allocated bundle b, prompts an LLM with the scenario $\mathcal{S}$, φ, b, and c_i. The LLM is asked to choose one of the following actions:

1. to message the person a value query;
2. to message the person a demand query; or
3. to reply to the auction that the person is satisfied.

With (1), the LLM also specifies a bundle b. We call a value query $\mathcal{M}_V^p(b)$ to message the person to receive their value ν for the bundle. This is used to create an updated hypothesis valuation $\theta' = (B', v')$, where $B' = B \cup \{b\}$ and v' is equal to v when given any item from B, and is equal to ν when given b.

With (2), we call a demand query, $\mathcal{M}_D^p(\varphi, b)$ to get the person's response and demanded bundle, (r, b'). If $r = 1$, the proxy forwards the information that the person is happy. If it is not, then we call $\mathcal{M}_V^p(b')$, receiving the person's value ν of b', and θ is correspondingly updated.

With (3), the bidder simply returns that it's happy, i.e. $(1, \theta)$.

Design 2 $\omega_{VD,2}$ is similar to $\omega_{VD,1}$ except that instead of directly returning its updated hypothesis valuation $\theta' = (B', v')$ it computes an inferred valuation $\theta^* = (B^*, v^*)$, which gives the same value $v^*(b_1) = v'(b_1)$ for all $b_1 \in B'$ and additionally infers values $v^*(b_2)$ for all $b_2 \in B'^c$, where B'^c is the set of bundles without explicitly specified values.

To construct θ^*, we first construct an inference function, $\gamma : \mathcal{B} \to \mathbb{R}_+$. γ is constructed at iterations of the CECA auction, including the first iteration, using an LLM pipeline. The LLM pipeline, given the proxy's memory, c_i, of their conversation with p, the scenario $\mathcal{S}$, and a bundle β, infers a value for β.

The pipeline first collates a prompt of the form

Here is the scenario description : $\mathcal{D}(\mathcal{S})$
Here is a transcript of the conversation with the person: c_i
What do you think the person would value the following
bundle at? $\mathcal{D}(b^*)$

and then calls an LLM with this prompt. The value the LLM responds with is denoted ν.

Thus, γ can be constructed by calling the pipeline with all bundles $\beta \in B'^c$. We set $\gamma(\beta)$ to $\nu \cdot \varepsilon$, where we discount the value ν given by the LLM by a constant factor of $\varepsilon = 0.75$ as it is better for the person to under-represent their valuation because otherwise their Lindahl prices under the auction will be set artificially high.*

Having constructed γ, we now construct θ^*. To do this, we follow the following procedure: maintaining a working $\theta^* = (B^*, v^*)$ initialized to θ', we iterate over an enumeration of all bundles in B'^c (the bundles where there is no explicit valuation) in ascending order of the number of items they contain and, for each bundle β, if $\beta \notin B'$ and $v^*(\beta) < \gamma(\beta)$, we add β to B^* and define $v^*(\beta) = \gamma(\beta)$.

The operation of the proxy ω_{NVD} has complexity exponential in the number of LLM calls because the calculation of γ requires an enumeration of all bundles, which has complexity exponential in the number of items. We will describe a strategy to restrict the number of bundles we need in the

*This value was chosen for ε by visual examination of the variability plots in Figure 3.2 and noting that 75% of the 75th percentile line indicated by the top of the boxplots was always below the mean value for all bundles and scenarios.

domain of γ in Chapter 5.

4.2.3 Plus-questions LLM proxy

We give one design for a plus-questions LLM-proxy, ω_{NVD}, that is based on the design of $\omega_{VD,2}$.

$\mathcal{M}_C^{\omega_{NVD}(p)}$ is the algorithm that is identical to $\mathcal{M}_C^{\omega_{VD,2}(p)}$ except that at the very first iteration of the auction, the proxy asks a natural language question q to the person they represent by calling $\mathcal{M}_N^p(q)$, where q is given by the LLM response to a prompt asking, "What should the proxy ask the person next to better understand their preferences?" given $\mathcal{D}(\mathcal{S})$.

Note that ω_{NVD}, like $\omega_{VD,2}$, also gives inferred θ^*. The additional information p provides in answering q is recorded in c_i and thus gives more information to the LLM pipelines for deciding what action to take and for inferring the valuations p holds for bundles without explicitly specified values.

4.2.4 Simulation results for pure-LLM proxies

We present in Figure 4.1 the results of the simulations with ω_{XOR}, $\omega_{VD,1}$, $\omega_{VD,2}$, and ω_{NVD} on our benchmark B by graphing PE_n, each proxy's average auction efficiency, versus n, the average number of interactions between a proxy and the person they represent. We plot up to $n = 16$ to illustrate the short-run behavior of these proxies; the long-run behavior will be examined in the following section. We find through simulation that we can reach more efficient outcomes more quickly using LLM-based proxies. We can reach 75% efficiency in ten interactions between the person and their proxy with ω_{XOR}, in four interactions with $\omega_{VD,2}$, and in two interactions with ω_{NVD}.

We also see from examination of Figure 4.1 that in the short-run the LLM-based proxy ω_{NVD} rapidly climbs in efficiency relative to the classical proxy ω_{XOR}, but the relative advantage of the LLM-based proxy diminishes with more iterations.

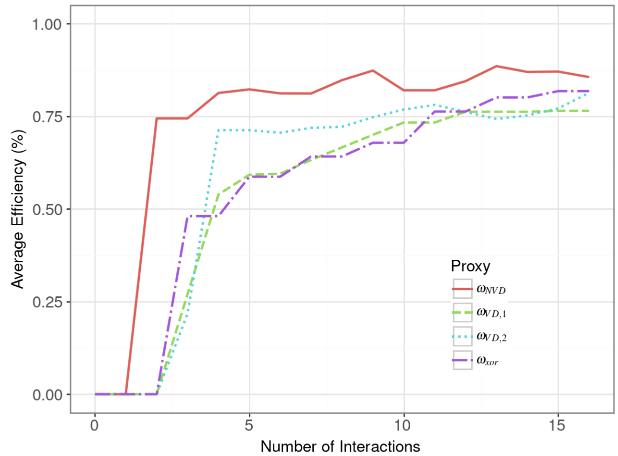

Figure 4.1: Average percent efficiency of the auction run under the three scenarios with three people participating in each. Anchoring on our XOR elicitation proxy, we see that the drop-in proxy $\omega_{VD,1}$ using an LLM performs similarly, both reaching 75% efficiency in around 10 interactions between person and proxy. The plus-questions proxy $\omega_{VD,2}$ performs markedly better, reaching 75% efficiency in four interactions that otherwise would take ten interactions. We see that the proxy ω_{NVD}, which is also allowed to use natural language questions, reaches the same 75% efficiency in two interactions.

4.3 Hybrid XOR-LLM proxies

We now consider the *hybrid XOR-LLM proxy* which is a hybrid of the classical proxy ω_{XOR} and the pure-LLM based proxy ω_{NVD}. The goal is to combine the short-run rapid increase in efficiency of the pure-LLM proxies, while maintaining the long-run convergence of classical proxies to efficient outcomes.

4.3.1 Design of hybrid XOR-LLM proxies

The hybrid XOR-LLM proxy ω_H combines ω_{NVD} and the ω_{XOR}. For person p and proxy $p' = \omega_H(p)$, we define $\mathcal{M}_C^{p'}$ as follows.

At the start when $\mathcal{M}_C^{p'}$ is called, we run $\mathcal{M}_C^{\omega_{NVD}(p)}$ such that the proxy is asking natural language questions and using the LLMs to infer valuations.

After a set number of iterations, α, we switch the algorithm such that when $\mathcal{M}_C^{p'}$ is called we switch to running $\mathcal{M}_C^{\omega_{XOR}(p)}$ where the proxy now uses the proper-learning algorithm. Throughout both the first α iterations and the period onwards, we continuously maintain and update the hypothesis valuation θ'.

Furthermore, before the switchover, recall that with $\mathcal{M}_C^{\omega_{NVD}(p)}$ we construct the inference function γ and at each iteration, construct and return the inferred valuation θ^*. After the switchover, we no longer construct γ in the same manner, but we continue constructing and returning θ^* based on the updated hypothesis valuation θ' given by $\mathcal{M}_C^{\omega_{XOR}(p)}$. We specify that at every iteration after the switchover, we update γ such that we have new $\gamma'(b) = \delta \cdot \gamma(b)$ for all $b \in \mathcal{B}$ and for $0 \leq \delta < 1$. This has the purpose of maintaining, in the short run, the benefits of using an inferred valuation, but in the long run, removing the inferred valuation by decaying γ to 0.

4.3.2 Simulation results for hybrid XOR-LLM proxies

The hybrid XOR-LLM proxy ω_H has the short-term rapid increase in efficiency without sacrificing long-term ability of the proxy to exactly learn the person's valuation and of the auction using the proxy to converge to the social-welfare-maximizing outcome. For the following two experiments, we set $\alpha = 10$, $\delta = 0.95$.[†]

First, we ran ω_H and ω_{XOR} as preference elicitation algorithms, by running the auction with only one person, using either ω_H or ω_{XOR} as a proxy, and constantly giving prices equal to the manifest valuation and allocating the empty bundle, and measured at each step n, the number of interactions the proxy had with the person they represent, and PE_n, the average auction efficiency for each proxy.

The results, plotted in Figure 4.2(a), show that ω_H approximates the person's valuation significantly quicker as the line for ω_H is significantly and consistently lower than that for ω_{XOR}. To put

[†]We set α to 10 to switch over to the classical proxy after an initial burst of LLM calls in the short-run. Discerning the most efficient setting of α would require more consideration, and is likely dependent on the number of items. We set δ to 0.95 to avoid abrupt shifts in manifest valuations, but also relatively quickly decay γ.

numbers to it, in all three scenarios, ω_H gets to a better approximation of the person's valuation in 5 interactions than ω_{XOR} does in 50 interactions. We note, however, that at the very end, ω_{XOR} gets to 0 error, e.g. exactly learns the person's valuation, around 10 interactions quicker than ω_H. This is because the LLM is sometimes inefficient with its queries. Additionally, though helpful for approximations, natural language questions don't aid as much in exact learning.

Second, we ran the auction simulations identical to those in Section 4.2.4. We aggregate and plot the data according to the same procedure as in Section 4.2.4 and extend it to include long-run data all the way until both proxies reach 100% efficiency. This is given in Figure 4.2(b).

We see that in the initial phase, ω_H rapidly reaches fairly efficient outcomes. This matches the results shown for ω_{NVD} which is expected as the algorithms $\mathcal{M}_C^{\omega_H(p)}$ and $\mathcal{M}_C^{\omega_{NVD}(p)}$ are identical for the first α iterations. We see that the efficiency of the outcome steadily increases as we increase the number of interactions—in line with ω_{XOR}—such that the social-welfare-maximizing outcome is also reached.

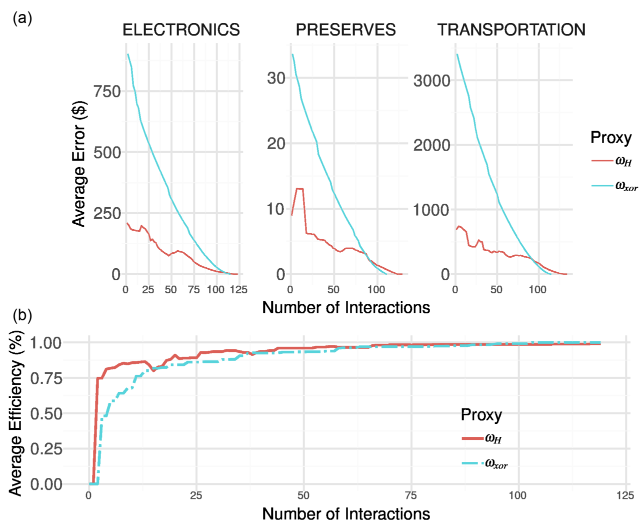

Figure 4.2: Performance of the hybrid XOR-LLM proxy ω_H (in red) and the XOR elicitation proxy ω_{XOR} (in teal) under varying numbers of interactions with the person they represent. Panel (a) shows that ω_H gets to a good approximation of the person's valuation significantly quicker than ω_{XOR} while still converging to the person's exact valuation. Panel (b) shows that when running the auction with ω_H, similar to Figure 4.1 we get a sharp increase in efficiency in the initial stages, while still converging on the welfare-maximizing outcome in the long run.

5

Scaling-up scenarios

In this chapter, we demonstrate the removal of the fact that the number of LLM calls that we use has complexity exponential in the number of items. This exponential complexity is incurred both on the simulation end (for the ground-truth simulated bidders to craft responses to demand query) and on the inference end (for the proxies $\omega_{VD,2}$, ω_{NVD}, and ω_H to infer a hypothesis valuation).

To evaluate the effectiveness of this approach to removing the exponential complexity, we compare the performance of this approach, which uses a partial enumeration of bundles, to that of the original approaches presented in Chapter 3 and Chapter 4, which use a full enumeration of bundles. We evaluate these approaches on an expanded benchmark B_2.

5.1 Benchmark creation (B_2)

B_2 is similar to B_1: for both benchmarks, we use the LLM-based seed generation described in Section 3.3.1 and message-response pipelines described in Section 3.3.2 to construct a benchmark encompassing the three scenarios. B_2 is a larger benchmark than B_1. B_2 consisting of thirty-six setups, with six setups per each of the three scenarios, where $B_2 = \{\beta_{i,j} | i \in \{1, 2, 3\}, 1 \leq j \leq 12\}$, $\beta_{i,j}$ is the setup $(\mathcal{S}_i, \mathcal{P}_{i,j})$, $\mathcal{P}_{i,j}$ is a population of three simulated people generated to participate in $\mathcal{S}_i$, and each simulated person has ground-truth XOR valuation $\theta_{i,j,k}$.

Our use of B_1 differs from B_2 in a crucial respect. In the previous chapter, the ground-truth XOR valuation $\theta_{i,j,k}$ was directly used to respond to queries from the proxies. For the purposes of this chapter, we cannot directly use $\theta_{i,j,k}$ to respond to queries from the proxy, because, in this chapter, we wish to compare the performance of a partial enumeration of bundles (with non-exponential complexity) versus the performance of a full enumeration of bundles (with exponential complexity). It would not be appropriate to compare a partial enumeration with a full enumeration queried from $\theta_{i,j,k}$ using the same $\theta_{i,j,k}$ to evaluate its performance in the sense that the full enumeration would be overfitted to $\theta_{i,j,k}$. So instead we construct a $\theta'_{i,j,k}$ by asking the bidder for an additional time what they would value every bundle at in the same manner we constructed $\theta_{i,j,k}$.

We use $\theta'_{i,j,k}$ during the course of the simulated auctions to implement the message response functions $\mathcal{M}_V^{p_{i,j,k}}$ and $\mathcal{M}_D^{p_{i,j,k}}$ to give the simulated person's response to value and demand queries. We use $\theta_{i,j,k}$ as the ground-truth valuation to evaluate the social-welfare of a given allocation. Equivalently put, we run the auction the same way as we did in the previous chapter, except that after the allocation is given we ask the bidder again to confirm for a second time what they would value that bundle at.

5.2 Removing exponential complexity on the simulation end

In this section, we work towards the removal of the exponential complexity in the simulation framework caused by the construction of a simulated person's dense-preferences XOR valuation θ, which as previously described in Chapter 3, requires the full enumeration of valuations over all bundles $b \in \mathcal{B}$. We will remove the exponential complexity by replacing the simulated persons' dense-preferences θ with sparse-preferences π; π will be constructed with a number of LLM calls with complexity polynomial in the number of items.

5.2.1 Construction of sparse-preferences

The construction of sparse-preferences π for a simulated person will be done by running a preference elicitation procedure directly on the simulated bidders. Preference elicitation is known to have polynomial-complexity in terms of the number of atomic bundles[26], so we must also ensure that the number of atomic bundles has polynomial-complexity. This point will be discussed after the preference elicitation procedure is described.

We will use the XOR preference elicitation algorithm described in Section 2.3.2 to construct π. When we previously used the XOR preference elicitation algorithm to construct preferences in Section 2.4.3 for ω_{XOR} and in Section 4.3.1 for ω_H, we could call demand queries $\mathcal{M}_D^p$ in the preference elicitation algorithm. However, because our implementation of the demand query response function for simulated person p given in Section 3.3.2, $\mathcal{M}_D^p$, requires first defining the XOR preferences of p, we cannot rely on calling demand queries to run our preference elicitation algorithm.* Thus,

*Suppose for the sake of argument that we did use demand queries for preference elicitation to construct π. Using demand queries means that we would have needed to first have defined the XOR preferences of p. We cannot use sparse-preferences π to circularly construct π so we would have to use the dense-preferences θ. Thus, we would have to construct θ to construct π and thus the construction of π would have exponential we implement the equivalence query response function to allow the simulated bidder to directly respond to equivalence queries without first defining the XOR preferences of p. This function is denoted $\mathcal{M}_E^p$.

To ensure polynomial-complexity in the number of atomic bundles, we implement it such that a simulated person only responds to an equivalence query saying that a hypothesis valuation is "Not equivalent" when the deviation from the simulated person's valuation is above a certain threshold ε. We can raise the threshold ε as appropriate to limit the number of atomic bundles.[†]

Implementation of equivalence query The equivalence query response function, $\mathcal{M}_E^p$, is implemented by an LLM call. We prompt the LLM to give the simulated person's response to the equivalence query given a description of the scenario $\mathcal{D}(\mathcal{S})$ and the person's preference $\mathcal{D}(p)$, the hypothesis XOR valuation defined in terms of the atomic bundles b_i and their corresponding valuations v_i, and the threshold ε. When using the equivalence query, we maintain a memory of prior value query responses during the preference elicitation process and prompt the LLM with it to simplify its task of responding to equivalence queries. Below is the prompt for the equivalence query LLM call.

```
Equivalence Query Prompt:

Description of scenario: D(S)
Description of the person's preferences: D(p)
*****
```

complexity due to the exponential complexity of constructing θ. So, we cannot rely on demand queries for the construction of π.

[†]In fact, when $\varepsilon = 0$, empirically, the simulated person almost never reports that the hypothesis valuation was equivalent. The LLM would always find some trivial reason to be upset with the hypothesis valuation.

```
Hypothesis XOR valuation function as a list of atomic bids:
- Atomic bid 1: Bundle: D(b1); Valued at v1
- Atomic bid 2: Bundle: D(b2); Valued at v2
- ...
*****
Explicit valuations:
- Bundle: D(b'1); Valued at v'1
- Bundle: D(b'2); Valued at v'2
- ...
*****
Please identify which bundles, if any, the hypothesis XOR valuation
function is most incorrect compared to the person's explicit valuations
and preferences. Ignore discrepancies less than ε.
```

We use the XOR preference elicitation algorithm using the equivalence query described above to obtain sparse preferences $\pi_{i,j,k}$ for each person $p_{i,j,k}$ in the benchmark B_2.

5.2.2 Sparse preferences performance

For the people $p_{i,j,k}$ in the populations $\mathcal{P}_{i,j}$ in B_2, their sparse preferences—the XOR valuations $\pi_{i,j,k}$—have an average of 16 atomic bids while their dense preferences have the full 64 atomic bids. Thus, the effective size of their preferences, required for responding to value and demand queries, is 25% of that under dense preferences.

For each setup $(\mathcal{S}_i, \mathcal{P}_{i,j}) \in B_2$, using the integer programming method described in Section 2.4.1, we solve for the social-welfare maximizing allocation when using the sparse preferences, $\{\pi_{i,j,1}, \pi_{i,j,2}, \pi_{i,j,3}\}$, for the three people in the population $\mathcal{P}_{i,j}$ and we also solve for the social-welfare maximizing allocation when using dense preferences, $\{\theta'_{i,j,1}, \theta'_{i,j,3}, \theta'_{i,j,2}\}$, for the three people in the population $\mathcal{P}_{i,j}$. We then solve for the auction efficiencies for both allocations, yielding $AE_{i,j,\pi}$ and $AE_{i,j,\theta}$, respectively, by using the ground-truth valuations $\{\theta_{i,j,1}, \theta_{i,j,2}, \theta_{i,j,3}\}$ for the people.

We calculate for each setup, $AE_{i,j,\theta'} - AE_{i,j,\pi}$, the difference in percent auction efficiency between using dense preferences and sparse preferences. We plot these differences in Figure 5.1 as well as the average difference. Statistical significance of the difference between using sparse and dense preferences is computed using a one-sample t-test on whether the percent differences are significantly different from 0, where 0 would indicate no difference between sparse and dense preferences.

We see that there is no statistically significant difference between using sparse and dense preferences in the TRANSPORTATION and PRESERVES scenario. There is a slight advantage in terms of auction efficiency for using dense preferences over sparse preferences in the ELECTRONICS scenario.

The use of sparse preferences over dense preferences is justified because the sparse preferences have polynomial-complexity, while the dense preferences have exponential-complexity, and using sparse preferences results in no statistically significant loss in efficiency or minimal loss in efficiency.

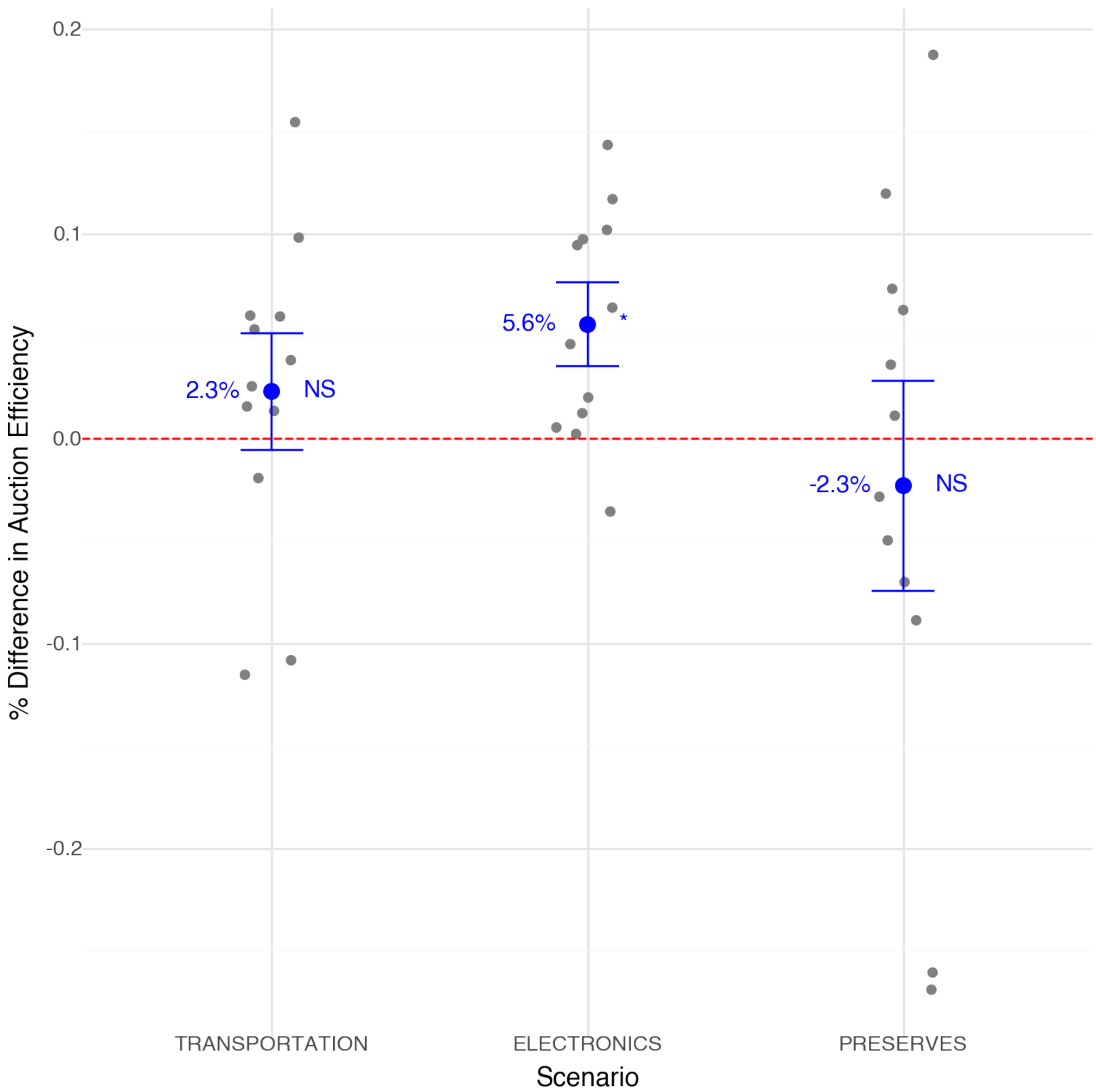

Figure 5.1: Percent difference in auction efficiency between using the dense-preferences simulated person and a sparse-preferences simulated person. We run the auction with the dense or sparse preferences simulated person and then evaluate the auction efficiency using the ground-truth simulated person. We see that there is no significant difference between the dense and the sparse preferences simulated person under the TRANSPORTATION and the PRESERVES scenario.

5.3 Removing exponential complexity on the inference end

In this section, we work towards the removal of the exponential complexity in the LLM proxies, specifically, the fact that the number of LLM calls used in the construction of the inferred valuations for $\omega_{VD,2}$, ω_{NVD}, and ω_H has exponential complexity in the number of items. Recall that in our design for these proxies in Chapter 4, for the proxy to construct the updated hypothesis valuation, which is submitted to the auction as the manifest valuation, the proxy needed to enumerate all bundles and use an LLM call to infer the person's valuation for each bundle. This means that those proxies use LLM calls with exponential complexity.

To remove this exponential complexity, instead of inferring the person's valuation for all possible bundles, we infer the person's valuation only for bundles with K items or fewer. We use this *restricted-inference* strategy to modify ω_H to create the proxy design, $\omega_{H,K}$. We simulate auctions using the setups of B_2 when using $\omega_{H,K}$ where $K = 1, 2, 3, 6$. Each scenario contains exactly 6 items so $\omega_{H,6}$ is equivalent to the ω_H and reflects the hybrid proxy with unrestricted inference.

We report in Table 5.1 for $n = 5, 7, 9$ the mean and standard deviation of PE_n, the auction efficiency at n interactions across the thirty-six setups of B_2, for $\omega_{H,K}$ and the ω_{XOR} classical proxy. We choose $n = 5, 7, 9$ because the hybrid proxy switches over from using LLM-based methods at $\alpha = 10$ iterations. Statistical significance is computed using a one-sample t-test on whether the differences in auction efficiency for each setup are significantly different from 0, where 0 would indicate no difference between proxy designs.

We see from Table 5.1 that we can significantly restrict proxies' use of LLM-inference to update the hypothesis valuation from full inference with $K = 6$ to with $K = 2$ or $K = 3$ bundles or fewer without a loss in auction efficiency. There is no statistically significant difference between $K = 6$ and $K = 2$ or $K = 3$ at any of $n = 5, 7, 9$. Thus, in these toy scenarios, we can go from inferring over 2^N possible bundles to inferring over the $\binom{6}{2} = 15$ bundles, which have 2 items or fewer.

Restricting inference as such results in a number of LLM calls with complexity polynomial in the number of items, because $\binom{n}{2}$ is polynomial in n.

Note, though, that we cannot restrict to only bundles of 1 item or fewer without efficiency loss. This is likely due to the design of the scenarios where there are complementary pairs of items, as well as the limited number of items ($N = 6$) which means that as the number of iterations quickly increases past 6 the LLM-inference on the 6 bundles that have only one item is less impactful towards increasing auction efficiency.

(a) $n = 5$

Proxy	Mean	Std Dev
$\omega^{*\dagger}_{H,6}$	0.73	0.11
$\omega^{*\dagger}_{H,3}$	0.76	0.09
$\omega^{*\dagger}_{H,2}$	0.74	0.12
$\omega_{H,1}$	0.46	0.11
ω_{XOR}	0.53	0.13

(b) $n = 7$

Proxy	Mean	Std Dev
$\omega^{*\dagger}_{H,6}$	0.75	0.09
$\omega^{*\dagger}_{H,3}$	0.76	0.10
$\omega^{*\dagger}_{H,2}$	0.77	0.14
$\omega_{H,1}$	0.60	0.18
ω_{XOR}	0.61	0.11

(c) $n = 9$

Proxy	Mean	Std Dev
$\omega^{*\dagger}_{H,6}$	0.77	0.09
$\omega^{*}_{H,3}$	0.74	0.12
$\omega^{*}_{H,2}$	0.76	0.14
$\omega_{H,1}$	0.69	0.14
ω_{XOR}	0.65	0.13

Significance Indicators:
$*$ Significantly different from ω_{XOR} ($p < 0.05$)
$\dagger$ Significantly different from $\omega_{H,1}$ ($p < 0.05$)

Table 5.1: Performance of K-hybrid LLM-XOR proxies in terms of aggregated auction efficiency under limited numbers of interactions with the bidders. Performing inference only on bundles of size $K = 2, 3$ or smaller does not have substantial impacts on performance versus performing full inference. Performing inference on bundles of only one item does have a significant impact on performance, in part due to not being able to capture complementary pairs of items.

6

Conclusion

In this work, we presented an innovative approach to preference elicitation in combinatorial auctions by integrating Large Language Models (LLMs) into proxies. Combinatorial auctions (CAs) play a crucial role in efficiently allocating multiple items, especially in complex scenarios such as spectrum sales and public goods distribution. However, the inherent exponential complexity of CAs poses significant challenges for both bidders and auctioneers. Traditional query-elicitation-based methods, while theoretically efficient, often impose substantial cognitive burdens on bidders due to the extensive communication and contemplation required. Our LLM-based proxy design mitigates this issue by leveraging natural language interactions, thereby streamlining the elicitation process and significantly reducing the number of necessary queries.

We designed a simulated testing sandbox to empirically evaluate our LLM-based proxy designs. We used LLM pipelines to create simulated bidders who mimicked human-like responses to queries. The LLM pipelines were used both to create natural language descriptions of preferences—seeds—under various scenarios and to craft responses to natural language questions as well as to value, demand, and equivalence queries. We showed that these pipelines created seeds that resulted in consistent interpretations across LLMs from different providers. We further showed that the valuations the pipelines characterized had properties of free disposal and reasonable precision.

However, we did not undertake any comparisons with human data. Thus, we must be wary of the degree to which our testing sandbox mimics human behavior. Specifically, future work is necessary to understand how representative the natural language descriptions of preferences we generate in the form of seeds are of human preferences, as well as how reasonable our simulated persons' responses to queries are. We make efforts to ensure that the queries issued by the proxies to the person they represent—value, demand, and natural language questions—are reasonably simple for people to respond to, and a more precise measure of the burden on bidders would tease apart the relative burden of various query types. We are encouraged, though, by the breadth of literature showing the ability of LLMs to mimic human responses in many diverse domains, especially the literature showing the ability to prompt LLMs with human preferences and receive well-calibrated preferences over unseen alternatives [13,40].

Empirical evaluations demonstrate that LLM-based proxies, particularly those that use LLMs to infer the bidder's valuation on unspecified bundles, achieve approximately efficient auction outcomes up to five times faster than classical XOR elicitation proxies. As illustrated in Figure 4.1, proxies such as $\omega_{VD,2}$ and ω_{NVD} reach 75% of the social welfare-maximizing outcome with markedly fewer interactions compared to the ω_{xor} proxy. Furthermore, the hybrid XOR-LLM proxy ω_H not only accelerates the initial elicitation phase but also converges to the social-welfare-maximizing outcome, by combining the rapid approximation capabilities of LLMs with the guarantees of classical preference elicitation methods.

Additionally, the robustness of LLM-based bidder modeling was validated across different models and repeated trials, ensuring consistent and reliable preference representations. This robustness underscores the feasibility of employing natural language processing in complex auction settings. By facilitating more efficient and accurate preference elicitation, LLMs contribute to the overarching goal of allocating resources to those who value them most.

A significant feature presented in this work pertains to the scalability of the approach. In Chapter 5, we address the exponential complexity inherent in both the simulation and inference processes present in Chapter 3 and Chapter 4, and demonstrate that our methodology should remain efficient even as the size and complexity of the auction environment grow. Specifically, we introduced sparse-preference models that replace dense XOR valuations, reducing the computational burden from exponential to polynomial complexity without sacrificing auction efficiency. Our experiments on the expanded benchmark B_2 revealed that using sparse preferences maintained auction efficiency in the TRANSPORTATION and PRESERVES scenarios, with only a minimal impact in the ELECTRONICS scenario. This outcome validates the effectiveness of sparse preference elicitation in large-scale settings, ensuring that our approach remains practical for real-world applications where the number of bundles can be prohibitively large.

On the inference side, we further enhanced scalability by limiting the scope of bundle valuations to those containing up to K items. This restricted-inference strategy significantly reduces the computational demands of the LLM proxies while retaining high levels of auction efficiency. Our findings indicate that inferring valuations for bundles with $K = 2$ or $K = 3$ items suffices to achieve performance comparable to full enumeration ($K = 6$) across various interaction levels. This selective inference approach not only streamlines the inference process but also ensures that the system remains responsive and efficient as the complexity of auction scenarios increases. Notably, restricting inference to single-item bundles ($K = 1$) was insufficient, highlighting the necessity of capturing multi-item interactions to maintain auction performance.

However, our studies on the polynomial-complexity strategies for simulation and proxy design are limited to the three scenarios with $n = 6$ items each. Thus, what we have shown is that we can use algorithms with polynomial complexity without significant effects on social welfare to replace algorithms that had exponential complexity in these scenarios. By using the polynomial-complexity algorithms, we reduced the size of the simulated person's preferences from 64 atomic bids to an average of 16. We also reduced the LLM calls used by the LLM-based proxies for inferring valuations from considering all bundles without explicit valuations out of 64 possible bundles to only 15 possible bundles with $K = 2$ items or fewer. We leave the complete demonstration of running the simulated combinatorial auctions with large-scale numbers of items to future work. We predict that as the ecosystem around training and deploying LLMs becomes more developed, many other practical concerns will diminish—the dollar costs associated with making LLM calls will decrease, the reasoning powers of LLMs will increase, and the context length of LLMs will increase—meaning that we will be able to leverage the polynomial-complexity algorithms in an effective manner at a large scale.

The removal of exponential complexity on both the simulation and inference ends marks a pivotal step toward deploying LLM-based proxies in large-scale, real-world auction environments. By ensuring that computational resources are managed effectively, our approach can scale to accommodate auctions with a vast number of items and participants without sacrificing efficiency or outcome quality. This scalability is crucial for practical adoption, as it enables the application of sophisticated preference elicitation techniques in diverse and dynamic market settings.

Moreover, this research exemplifies an exciting frontier in the use of LLMs to simulate economic settings and test auction and proxy designs. By harnessing the advanced natural language understanding and reasoning capabilities of LLMs, we can create more realistic and adaptable models of human behavior in auction settings, enabling the testing and refinement of mechanisms in a controlled yet dynamic environment. This synergy between artificial intelligence and economic theory opens up new possibilities for designing auctions that are not only theoretically sound but also practically viable and user-friendly. The ability to simulate complex interactions and preferences using LLMs may accelerate the innovation cycle by allowing for rapid prototyping and evaluation of novel auction formats and proxy strategies.

In summary, our findings highlight the potential of LLMs to revolutionize preference elicitation in multiagent systems by making the process more intuitive via natural language questions and less burdensome through significantly fewer queries for participants. By addressing the exponential complexity challenges inherent in combinatorial auctions, our approach not only enhances auction efficiency but also might promote broader participation by decreasing the burden on bidders. Our construction of the non-exponential version of the simulated persons and proxies, without meaningful loss in auction efficiency, underscores the viability of this approach for scalable auction simulation and design. Future research could explore the scalability of this approach to even larger auction environments and investigate the integration of the developments in increasingly advanced LLM capabilities to further enhance elicitation efficiency and accuracy at lower computational costs.

A

Example LLM outputs

A.1 Example Seed

```
Cecilia is focused on acquiring the Apple Pencil Pro (code: APPLEPENCILPRO) for
    her digital art. She places a high value on tools that enhance her creative
    capabilities, specifically seeking industry-leading performance and
```

precision for her artistic projects. The dollar value for the Apple Pencil Pro she is willing to pay is estimated at $120, as this aligns with retail market prices for similar tools.

Cecilia is open to considering additional items that can complement the Apple Pencil Pro and significantly improve her workflow. Among the other available items, she may find the following combinations valuable, each associated with an estimated dollar value based on market rates:

1. **Single Item Bundle:**
   - Apple Pencil Pro (code: APPLEPENCILPRO) - $120

2. **Complementary Bundles:**
   - iPad Air 11-inch (M2) with Apple Pencil Pro:
     - iPad Air (code: IPAD12) - $600
     - Apple Pencil Pro (code: APPLEPENCILPRO) - $120
     - Total Value: $720
   - iPad (9th Generation) with Apple Pencil Pro:
     - iPad (code: IPAD9) - $350
     - Apple Pencil Pro (code: APPLEPENCILPRO) - $120
     - Total Value: $470

Cecilia considers these other items because they can enhance her digital art experience, allowing her to utilize the Apple Pencil Pro more effectively. The iPad models provide her the necessary platform for creative applications , which is a key factor in her evaluation.

However, Cecilia prioritizes the Apple Pencil Pro as her primary target and feels that any additional item should not just be an upgrade but must genuinely elevate her creative process. She is likely to discount bundles that include items she believes do not contribute directly to her artistic workflow, such as the Apple AirPods or AirPods Max, as they do not enhance her art-making activities.

In conclusion, while Cecilia's main interest is the Apple Pencil Pro, she values potential bundles that combine it with compatible iPads, estimating these combinations could be worth between $470 and $720 depending on the iPad model chosen, thus weighing her bids according to the increased functionality they provide for her creative work.

Cecilia's evaluation process for complex bundles involves a systematic assessment of how each item's inclusion enhances her creative workflow. She prioritizes the Apple Pencil Pro as her core target, and her perception of any additional items is predominantly based on their ability to provide meaningful improvements to her digital art experience.

When considering large bundles, Cecilia does not view them merely as additive collections of items; she assesses the synergy between constituents. Each additional item must deliver substantial value to her artistic capabilities; otherwise, she perceives them as superfluous and potentially detracting from her primary objective. Therefore, while she appreciates bundles that combine compatible products (like the iPads with the Apple Pencil Pro) for their holistic benefits, she remains cautious with larger bundles that include irrelevant components, which she would discount heavily or avoid.

Regarding the necessity of large bundles, Cecilia is not averse to accepting items individually; rather, she prefers a tailored approach that allows her to select only those products that meet her artistic needs. However, if a bundle includes items she considers essential or significantly enhancing (like the iPad models), she is more likely to accept it despite its size.

Cecilia values multiples of similar items additively, but she applies a discount for redundancy. For example, if two artistic tools serve similar functions, she might estimate their combined value at 80% of the total individual retail prices, reflecting a 20% discount on the perceived benefit of repeated functionality. Her evaluation criteria incorporate percentage discounts based on utility:

1. **Complementary Value:** Items that enhance functionality (like the iPads with the Apple Pencil Pro) are valued at a premium, with a direct uplift of

about 20% to 30% in her willingness to pay.

2. **Substitution Effects:** If an item is viewed solely as a substitute rather than an enhancement (e.g., she may consider less advanced versions or unnecessary accessories), she would apply a significant discount, potentially around 50% or more on her perceived value.
3. **Bulk Effects:** For larger bundles, Cecilia's assessment would involve a progressive discount model; the first two items may maintain nearly full value, but each subsequent item may see a decline in perceived value of about 10% to 15%, reflecting the diminishing returns of utility in a larger collection.

In conclusion, Cecilia's decision-making process revolves around a clear framework where she rigorously evaluates the functional contributions of each item based on how effectively they can support and amplify her artistic endeavors. This meticulous approach ensures that her bids are strategically aligned with her artistic goals, emphasizing quality and relevant enhancements over quantity.

A.2 Example Bundle-Valuation Subroutine Call

```
To estimate Cecilia's value for the proposed bundle of items (which includes the
     iPad Air 11-inch (IPAD12) and the Apple Pencil (2nd Generation) (
    APPLEPENCIL2)), let's follow the five-step process outlined:

1. **Check if the person has explicitly stated the value for that bundle**:
    Cecilia has not explicitly stated a value for the bundle consisting of the
    iPad Air and the Apple Pencil (2nd Generation).

2. **Find the closest bundle(s) from the given bundle that the person has
    explicitly valued**:
   - She has explicitly valued the iPad Air (IPAD12) bundled with the Apple
      Pencil Pro (APPLEPENCILPRO) at $720.
   - She has also valued the iPad (IPAD9) bundled with the Apple Pencil Pro (
      APPLEPENCILPRO) at $470.

3. **Identify the process by which the person has specified they will calculate
    their value**:
   - Cecilia applies a premium value to combinations of compatible items that
      enhance her functionality.
   - Since she highly values the Apple Pencil Pro for her creative work, the
      inclusion of an Apple Pencil (2nd Generation) will likely have a
      different perceived value compared to the Pro variant.
```

- However, she has not specifically stated any value for the Apple Pencil (2nd Generation) or how it compares to the Pro version in this type of bundle.

4. **Identify any other relevant criteria**:
   - Cecilia would likely apply a discount for the inclusion of the Apple Pencil (2nd Generation) rather than the Apple Pencil Pro.
   - The iPad Air (M2) bundle was valued higher when combined with the Pro version of the Pencil, indicating she sees higher value in advanced tools that enrich her creative workflow.

5. **Factor (2), (3), and (4) in to estimate the person’s value**:
   - The iPad Air (IPAD12) has a baseline value of $600.
   - The Apple Pencil (2nd Generation) generally might not add as much value as the Pro version, which Cecilia values at $120; however, since it is still a creative tool, she won’t discount it too heavily but will assign it a lower value (let’s estimate around $80 instead of the Pro’s $120).
   - The value of the proposed bundle can be assessed as follows:
     - iPad Air: $600
     - Apple Pencil (2nd Generation): $80
     - Combined Value: $600 + $80 = $680

Since the Apple Pencil (2nd Generation) is not as substantial an upgrade in her

    creative workflow compared to the Pro version, we may apply a slight
    adjustment downward, let’s say a 10% discount for it not being the Pro model
    :

- Final estimated value: $680 - ($680 * 0.10) = $680 - $68 = $612.

In conclusion, the estimated value of the proposed bundle is:

‘‘‘Bundle value: $612‘‘‘

Bibliography

[1] Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2(4), 319–342.

[2] Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J., Rytting, C., & Wingate, D. (2023). Out of One, Many: Using Language Models to Simulate Human Samples. *Political Analysis*, 31(3), 337–351. arXiv:2209.06899 [cs].

[3] Ausubel, L. M. & Baranov, O. (2017). A Practical Guide to the Combinatorial Clock Auction. *The Economic Journal*, 127(605), F334–F350.

[4] Binz, M., Akata, E., Bethge, M., Brändle, F., Callaway, F., Coda-Forno, J., Dayan, P., Demircan, C., Eckstein, M. K., Éltető, N., Griffiths, T. L., Haridi, S., Jagadish, A. K., Ji-An, L., Kipnis, A., Kumar, S., Ludwig, T., Mathony, M., Mattar, M., Modirshanechi, A., Nath, S. S., Peterson, J. C., Rmus, M., Russek, E. M., Saanum, T., Scharfenberg, N., Schubert, J. A., Buschoff, L. M. S., Singhi, N., Sui, X., Thalmann, M., Theis, F., Truong, V., Udandarao, V., Voudouris, K., Wilson, R., Witte, K., Wu, S., Wulff, D., Xiong, H., & Schulz, E. (2024). Centaur: a foundation model of human cognition. arXiv:2410.20268 [cs.LG].

[5] Binz, M. & Schulz, E. (2023). Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6).

[6] Blum, A., Jackson, J., Sandholm, T., & Zinkevich, M. (2004). Preference elicitation and query learning. *J. Mach. Learn. Res.*, 5, 649–667.

[7] Chakravorti, B., Sharkey, W. W., Spiegel, Y., & Wilkie, S. (1995). Auctioning the airwaves: The contest for broadband spectrum. *Journal of Economics amp; Management Strategy*, 4(2), 345–373.

[8] Coda-Forno, J., Binz, M., Wang, J. X., & Schulz, E. (2024). CogBench: a large language model walks into a psychology lab. arXiv:2402.18225 [cs].

[9] Conen, W. & Sandholm, T. (2001). Preference elicitation in combinatorial auctions. *Proceedings of the 3rd ACM conference on Electronic Commerce*, (pp. 256–259).

[10] Cramton, P. C., Ed. (2010). *Combinatorial auctions*. Cambridge, Mass.: MIT Press, 1. paperback ed edition.

[11] Ergin, H. & Sarver, T. (2010). A unique costly contemplation representation. *Econometrica*, 78(4), 1285–1339.

[12] Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv:2404.00806 [econ].

[13] Fish, S., Gölz, P., Parkes, D. C., Procaccia, A. D., Rusak, G., Shapira, I., & Wüthrich, M. (2023). Generative Social Choice. arXiv:2309.01291.

[14] Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., & Leahy, C. (2021). The pile: An 800gb dataset of diverse text for language modeling. arXiv:2101.00027 [cs.CL].

[15] Google DeepMind (2024a). AI achieves silver-medal standard solving International Mathematical Olympiad problems. url:https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/.

[16] Google DeepMind (2024b). Gemini breaks new ground with a faster model, longer context, AI agents and more. *Google*. url:https://blog.google/technology/ai/google-gemini-update-flash-ai-assistant-io-2024/.

[17] Gudiño, J. F., Grandi, U., & Hidalgo, C. (2024). Large language models (LLMs) as agents for augmented democracy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2285), 20240100.

[18] Hagendorff, T., Fabi, S., & Kosinski, M. (2023). Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nature Computational Science*, 3(10), 833–838.

[19] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., & Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *NeurIPS*.

[20] Horton, J. J. (2023). *Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?* Working Paper 31122, National Bureau of Economic Research.

[21] Hu, S., Lu, C., & Clune, J. (2024). Automated Design of Agentic Systems. arXiv:2408.08435 [cs].

[22] Katz, D. M., Bommarito, M. J., Gao, S., & Arredondo, P. (2023). GPT-4 Passes the Bar Exam. *SSRN Electronic Journal*.

[23] Kearns, M. J. & Vazirani, U. V. (1994). *An introduction to computational learning theory*. The MIT Press. London, England: MIT Press.

[24] Klemperer, P. (2004). *Auctions: theory and practice*. Toulouse lectures in economics. Princeton: Princeton University Press. OCLC: ocm53001276.

[25] Krishna, V. (2009). *Auction theory*. Academic press.

[26] Lahaie, S. M. & Parkes, D. C. (2004). Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce* (pp. 180–188).

[27] Li, N., Gao, C., Li, M., Li, Y., & Liao, Q. (2024). EconAgent: Large language model-empowered agents for simulating macroeconomic activities. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 15523–15536). Bangkok, Thailand: Association for Computational Linguistics.

[28] Lin, R. Y., Ojha, S., Cai, K., & Chen, M. F. (2024). Strategic Collusion of LLM Agents: Market Division in Multi-Commodity Competitions. arXiv:2410.00031 [cs].

[29] Manning, B., Zhu, K., & Horton, J. (2024). *Automated Social Science: Language Models as Scientist and Subjects*. Technical Report w32381, National Bureau of Economic Research, Cambridge, MA.

[30] Mencken, H. L. (1942). *New dictionary of quotations*. New York, NY: Alfred A. Knopf.

[31] Meta (2024). The Llama 3 Herd of Models. arXiv:2407.21783 [cs].

[32] Munkhdalai, T., Faruqui, M., & Gopal, S. (2024). Leave no context behind: Efficient infinite context transformers with infini-attention. arXiv:2404.07143 [cs.CL].

[33] Nielsen, K. & Rehbeck, J. (2022). When Choices Are Mistakes. *American Economic Review*, 112(7), 2237–2268.

[34] Nisan, N. (2000). Bidding and allocation in combinatorial auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce* (pp. 1–12).

[35] Nisan, N. (2005). Bidding Languages for Combinatorial Auctions. In P. Cramton, Y. Shoham, & R. Steinberg (Eds.), *Combinatorial Auctions* (pp. 215–232). The MIT Press.

[36] Nisan, N. & Segal, I. (2006). The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1), 192–224.

[37] OpenAI (2023). Gpt-4 technical report. arXiv:2303.08774 [cs.CL].

[38] OpenAI (2024). GPT-4o mini: advancing cost-efficient intelligence. url:https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

[39] Palacios-Huerta, I., Parkes, D. C., & Steinberg, R. (2024). Combinatorial Auctions in Practice. *Journal of Economic Literature*, 62(2), 517–553.

[40] Park, J. S., Zou, C. Q., Shaw, A., Hill, B. M., Cai, C., Morris, M. R., Willer, R., Liang, P., & Bernstein, M. S. (2024). Generative Agent Simulations of 1,000 People.

[41] Parkes, D. C. & Ungar, L. H. (2000). Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (pp. 82–89).: AAAI Press.

[42] Peacock, A. T. & Musgrave, R. A., Eds. (1994). *Classics in the theory of public finance*. International Economic Association Series. Gordonsville, VA: Palgrave Macmillan, 1958 edition.

[43] Pekeč, A. & Rothkopf, M. H. (2003). Combinatorial Auction Design. *Management Science*, 49(11), 1485–1503.

[44] Perron, L. & Furnon, V. (2024). Or-tools. url:https://developers.google.com/optimization/.

[45] Porter, D., Rassenti, S., Roopnarine, A., & Smith, V. (2003). Combinatorial auction design. *Proceedings of the National Academy of Sciences*, 100(19), 11153–11157.

[46] Rahaman, N., Weiss, M., Wüthrich, M., Bengio, Y., Li, L. E., Pal, C., & Schölkopf, B. (2024). Language Models Can Reduce Asymmetry in Information Markets. arXiv:2403.14443 [cs].

[47] Rassenti, S. J., Smith, V. L., & Bulfin, R. L. (1982). A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, 13(2), 402.

[48] Simon, H. A. (1990). *Bounded Rationality*, (pp. 15–18). Palgrave Macmillan UK: London.

[49] Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., & Zhang, Z. (2024). Qwen2.5-math technical report: Toward mathematical expert model via self-improvement.

[50] Zhao, Q., Wang, J., Zhang, Y., Jin, Y., Zhu, K., Chen, H., & Xie, X. (2024). Competeai: Understanding the competition dynamics of large language model-based agents. In *Forty-first International Conference on Machine Learning*.