Algorithms for Bayesian Persuasion and Delegated Search

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich 12
der Johann Wolfgang Goethe - Universität
in Frankfurt am Main

von
Niklas Hahn
aus Frankfurt am Main

Frankfurt 2021
(D 30)

vom Fachbereich 12 der

Johann Wolfgang Goethe - Universität als Dissertation angenommen.

Dekan: Prof. Dr. Martin Möller

Gutachter: Prof. Dr. Martin Hoefer
Prof. Dr. Ulrich Meyer
Prof. Dr. Inbal Talgam-Cohen

Datum der Disputation: 27.04.2022

Deutsche Zusammenfassung

In dieser Dissertation betrachten wir Algorithmen zur Beeinflussung von Handlungen durch strategisches Teilen sowie Vorenthalten von Informationen in verschiedenen Varianten. Eine wichtige Gemeinsamkeit aller betrachteten Szenarien ist die Existenz von *Commitment Power*, der „Fähigkeit, sich glaubhaft festlegen zu können“. Gemeint ist hiermit, sich im Vorhinein für eine Handlungsstrategie basierend auf allgemein bekannten Informationen zu entscheiden, diese öffentlich bekannt zu machen und sich zu verpflichten, nach dieser Strategie zu handeln.

Das grundlegende Szenario ist das Folgende:
Es gibt zwei rational agierende Parteien mit eigenen Interessen, die jedoch in gewissem Maße aufeinander eingehen müssen. Eine der Parteien besitzt mehr Informationen über den Zustand der Welt als die andere, benötigt jedoch die zweite Partei, um davon zu profitieren. Eine gewisse Grundinformation über den Zustand der Welt ist jedoch beiden Parteien bekannt, zum Beispiel durch eine bekannte Verteilung über mögliche Zustände. Erstere Partei versucht durch das Senden bestimmter Signale die Handlungen der zweiten zu beeinflussen. Wir bezeichnen die sendende Partei kurz als $\mathcal{S}$, die zweite Partei, die die Signale empfängt, entsprechend als $\mathcal{R}$ (nach dem englischen Begriff *receiver*). Letztendlich muss $\mathcal{R}$ nur anhand der Signale von $\mathcal{S}$ und der im Vorhinein bekannten Information eine von n verschiedenen Aktionen auswählen, ohne den genauen Typen dieser Aktion zu kennen. Abhängig vom Typ der gewählten Aktion ziehen die beiden Parteien unterschiedlichen Nutzen aus dieser Wahl. Allerdings gehen wir davon aus, dass der Nutzen, den eine Aktion mit sich bringt, niemals negativ sein kann. Da es keinen Nutzen bringt, wenn keine Aktion gewählt wird, ist es nicht von Vorteil für $\mathcal{S}$ oder für $\mathcal{R}$, wenn $\mathcal{R}$ letztendlich keine Aktion auswählt. Wir betrachten zwei verschiedene Grundszenarien. Im ersten Teil der Arbeit behandeln wir *Bayesian Persuasion* mit unterschiedlichen Modellen, der zweite Teil handelt von *Delegated Search*.

Bayesian Persuasion

In Bayesian Persuasion – grob übersetzt mit „Bayessche[1] Überzeugungskunst“ – besitzt $\mathcal{S}$ Commitment Power. Das bedeutet, dass $\mathcal{S}$ sich auf ein Signalschema φ festlegen und dieses Schema $\mathcal{R}$ kommunizieren muss. Das Schema φ beschreibt, in welcher Situation welche Signale gesendet werden. Erst danach erhält $\mathcal{S}$ Informationen über den wahren Zustand der Welt. Daraufhin empfängt $\mathcal{R}$ die entsprechenden Signale, bestimmt durch φ, und trifft eine Entscheidung für eine der Aktionen, ohne genau zu wissen, welchen

[1] Benannt nach dem englischen Mathematiker Thomas Bayes, der als erster den nach ihm benannten Satz von Bayes über bedingte Wahrscheinlichkeiten beschrieben hat [18].

Nutzen die gewählte Aktion für beide Parteien hat. Da die Signale jedoch auf dem bekannten Schema φ basieren, kann $\mathcal{R}$ durch Bedingung auf die empfangenen Signale eine Aktualisierung der Annahmen über den Zustand der Welt durchführen und diese zusätzliche Information in die Entscheidung für eine der Aktionen einfließen lassen. Dies muss $\mathcal{S}$ folglich für das Design von φ berücksichtigen. $\mathcal{R}$ wird keinen Empfehlungen für bestimmte Aktionen folgen, wenn diese nur vorteilhaft für $\mathcal{S}$, nicht jedoch für $\mathcal{R}$ selbst sind. Wir betrachten das Problem aus der Sicht von $\mathcal{S}$ und versuchen also, Signalschemata zu entwickeln, die einen möglichst großen Nutzen für $\mathcal{S}$ garantieren.

Offline-Fall

Wir behandeln zuerst den *Offline*-Fall von Bayesian Persuasion in Kapitel 3. Im Offline-Fall erfährt $\mathcal{S}$ den kompletten Zustand der Welt auf einmal und schickt daraufhin ein Signal an $\mathcal{R}$. Zu diesem Szenario gibt es bereits einige Quellen, die vor allem Härteresultate zeigen [36, 34]. Während in [36] gezeigt wird, dass es $\#\mathcal{P}$-hart ist, den optimalen Wert für $\mathcal{S}$ zu bestimmen, wenn die Typen der Aktionen unabhängig voneinander aus bekannten Verteilungen gezogen werden, behandeln die Autoren in [34] den Fall, dass $\mathcal{S}$ maximal $k < n$ mögliche Signale verschicken kann. Das bedeutet, dass von vornherein nicht alle verschiedenen Aktionen eine Empfehlung erhalten können. Für diesen Fall wird gezeigt, dass es $\mathcal{NP}$-schwer ist, ein Signalschema zu konstruieren, das einen konstanten Approximationsfaktor garantiert. Wir setzen daran an und identifizieren Bedingungen, unter denen wir effiziente Algorithmen beschreiben können, die das Problem für $k \leq n$ mögliche Signale optimal oder wenigstens mit einem konstanten Approximationsfaktor lösen können. Die Hauptresultate sind die folgenden:

1. Wenn die zugrunde liegende Verteilung über die möglichen Zustände der Welt bestimmte Symmetriebedingungen erfüllt, können wir in Polynomialzeit ein optimales Signalschema für $\mathcal{S}$ berechnen, selbst, wenn es maximal k Signale gibt. Unser Algorithmus verwendet geometrische Eigenschaften der Instanzen, um für jeden möglichen Zustand der Welt ein optimales Signal zu bestimmen.

2. Wenn die Aktionen unabhängig voneinander aus bekannten Verteilungen gezogen werden und eine weitere Bedingung erfüllen, die wir ϱ_E-Optimalität nennen, so existieren Polynomialzeitalgorithmen, die einen konstanten Approximationsfaktor garantieren. Dies gilt auch, wenn die Anzahl Signale durch k beschränkt ist. Unsere Algorithmen bestehen aus zwei Schritten. Im ersten Schritt bestimmen sie eine möglichst gute Menge S von k Aktionen, die potenziell eine Empfehlung erhalten werden. Im zweiten Schritt bestimmen sie, basierend auf der Menge S, ein Signalschema.

Online-Fall

In Kapitel 4 betrachten wir Bayesian Persuasion unter einer *Online*-Annahme. Das bedeutet, dass der Zustand der Welt Aktion für Aktion von $\mathcal{S}$ beobachtet wird und nur die jeweils aktuelle Aktion empfohlen werden kann. Wenn $\mathcal{R}$ sich dazu entscheidet, sie nicht auszuwählen, kann sie zukünftig nicht mehr gewählt werden. Zuerst behandeln wir eine Online-Variante, in der die Typen der Aktionen unabhängig voneinander aus

bekannten Verteilungen gezogen werden. Zunächst zeigen wir, dass ein Polynomialzeitalgorithmus existiert, der unter Nutzung von n aufeinander aufbauenden linearen Programmen ein optimales Signalschema für $\mathcal{S}$ berechnen kann. Wir messen den Erfolg für $\mathcal{S}$ mit dem Wert, den $\mathcal{S}$ im Offline-Fall erreichen könnte, also mit dem Wissen über die Typen aller Aktionen. Hierbei gibt es Fälle, in denen selbst das optimale Schema keinen positiven Approximationsfaktor erzielen kann. Darauffolgend identifizieren wir eine Klasse von Instanzen, in der $\mathcal{S}$ einen besseren Faktor im Vergleich zum Offline-Fall erreichen kann. Auch hier spielt die ϱ_E-Optimalität eine Rolle, die wir bereits im Offline-Fall kennen gelernt haben.
Im zweiten Abschnitt zu Online-Persuasion betrachten wir Aktionen, über deren Typen zuvorderst bekannt ist, dass sie in einer zufälligen Reihenfolge aufgedeckt werden. Wir behandeln dieses Szenario mit unterschiedlichem Wissen über die möglichen Werte für $\mathcal{S}$ oder $\mathcal{R}$ sowie mit unterschiedlichen Zielfunktionen für die beiden Parteien. In Abschnitt 4.2.1 betrachten wir den Fall, dass $\mathcal{R}$ den Erwartungswert der gewählten Aktion maximieren möchte. Als Richtwert verwenden wir den Fall der kompletten Information. In diesem Fall kennen sowohl $\mathcal{S}$ als auch $\mathcal{R}$ alle Typen, wissen jedoch beide nicht, welcher Typ welcher Aktion zugeordnet wird. Wir verwenden wiederum die geometrische Interpretation der Instanz, um unser Signalschema aufzustellen.
Weiterhin betrachten wir den Fall, dass die Werte der Typen unbekannt sind. Hier muss $\mathcal{S}$ erst einmal eine gewisse Anzahl von Aktionen beobachten, um eine fundierte Entscheidung treffen und ein gutes Signal senden zu können. Damit gelingt es, einen konstanten Approximationsfaktor bezüglich des Richtwerts für $\mathcal{S}$ zu garantieren. Zusätzlich betrachten wir die beiden beschriebenen Szenarien in dem Fall, dass $\mathcal{R}$ nach der Ablehnung einer Aktion auch erfährt, welchen Typ diese Aktion hatte. Für den Fall mit kompletter Information bedeutet dies nur einen Verlust eines konstanten Faktors für $\mathcal{S}$. Im Fall der unbekannten Typen führt diese zusätzliche Information für $\mathcal{R}$ jedoch dazu, dass keine konstante Approximation im Bezug auf den Richtwert mehr möglich ist. Stattdessen zeigen wir, dass $\mathcal{S}$ nur eine Approximation von $\Theta\left(^1/_n\right)$ erreichen kann.
Im folgenden Abschnitt 4.2.2 behandeln wir den Fall, dass $\mathcal{R}$ nur Interesse an der besten Aktion hat und die Erfolgswahrscheinlichkeit, diese Aktion auszuwählen, maximieren möchte. Wir betrachten wieder dieselben 4 Fälle, also komplette Information beziehungsweise keine Information über die möglichen Typen der Aktionen, jeweils mit und ohne Aufdeckung der abgelehnten Aktionen. In allen Fällen gelingt es uns, konstante Approximationsfaktoren zu erzielen.

Delegated Search

In Kapitel 5 betrachten wir Delegated Search, was sich ungefähr mit „Delegierter Suche“ übersetzen lässt. Ein grundlegender Unterschied zu Bayesian Persuasion besteht darin, dass nun $\mathcal{R}$ Commitment Power besitzt und nicht mehr $\mathcal{S}$. Wiederum möchte $\mathcal{R}$ eine Aktion aus n möglichen auswählen, möchte aber nicht selbst die Suche nach der besten Aktion durchführen. Hier kommt die Delegierung an $\mathcal{S}$ ins Spiel. Da $\mathcal{S}$ jedoch auch einen Wert von der letztendlich gewählten Aktion erhält und $\mathcal{R}$ die Werte der Aktionen nicht kennt, ist es nicht unbedingt vorteilhaft, $\mathcal{S}$ die komplette Entscheidung zu überlassen.
Formal betrachten wir den folgenden Aufbau: Für jede Aktion existiert eine Verteilung, aus welcher der Typ unabhängig von den anderen gezogen wird. Diese Verteilung ist so-

wohl $\mathcal{S}$ als auch $\mathcal{R}$ bekannt. Damit ist $\mathcal{R}$ in der Lage, die Menge der akzeptablen Typen zu beschränken, also bestimmte Typen von vornherein auszuschließen. Aufgrund von Commitment Power kann sich in diesem Fall $\mathcal{R}$ glaubhaft darauf festlegen, bestimmte Typen (mit gewisser Wahrscheinlichkeit) zu akzeptieren beziehungsweise Typen von vornherein auszuschließen. Wir bezeichnen das Akzeptanzschema mit φ. Mit dem Wissen um φ kann $\mathcal{S}$ nun für sich selbst optimieren – anders als $\mathcal{R}$ ist $\mathcal{S}$ in der Lage, die realisierten Typen der Aktionen zu beobachten. Nach der Entscheidung für eine Aktion wird $\mathcal{S}$ diese vorschlagen. Mit einem Vorschlag für eine Aktion erfährt $\mathcal{R}$ auch den Typ dieser Aktion. Abhängig von φ und dem Typen der vorgeschlagenen Aktion wird $\mathcal{S}$ diese akzeptieren oder ablehnen und der Prozess endet. Kleinberg und Kleinberg [58] betrachteten dieses Szenario aus algorithmischer Sicht und zeigten für den Fall von identischen Verteilungen für alle Aktionen, dass $\mathcal{R}$ in Erwartung nur einen kleinen konstanten Faktor des optimalen Nutzens verliert – den $\mathcal{R}$ durch eine eigene Suche erreichen könnte. Hierfür verwendeten Kleinberg und Kleinberg Techniken und Resultate aus der Online-Optimierung.

Wir betrachten das Problem aus der Online-Perspektive und beantworten damit die folgende Frage: Erhält $\mathcal{R}$ immer noch eine gute Approximation, wenn $\mathcal{S}$ die Aktionen einzeln nacheinander sieht und jeweils entscheiden muss, diese vorzuschlagen oder endgültig zu verwerfen?

Als Richtwert für die Approximation im Online-Fall verwenden wir den Wert, den $\mathcal{R}$ durch eine Online-Suche erzielen könnte.

Als Antwort auf diese Frage zeigen wir, dass $\mathcal{R}$ im Allgemeinen nur eine $\Theta\left({}^{1}\!/{}_{n}\right)$-Approximation erreichen kann bezüglich des Wertes, den $\mathcal{R}$ durch eine Online-Suche erzielt. Dies gilt, selbst wenn alle Aktionen aus derselben Verteilung gezogen werden. Allerdings nutzt unsere Beispielinstanz eine exponentielle Diskrepanz in den Werten für $\mathcal{S}$. Entsprechend betrachten wir zwei parametrisierte Fälle. Im ersten beschränken wir den Faktor zwischen dem minimalen und dem maximalen Wert für $\mathcal{S}$ mit $\alpha \geq 1$. Für diesen Fall zeigen wir eine $\Theta\left(\frac{\log\log\alpha}{\log\alpha}\right)$-Approximation. Wenn die Werte für $\mathcal{S}$ also keine exponentielle Diskrepanz haben, ist folglich auch die Approximation für $\mathcal{R}$ eine bessere. Weiterhin betrachten wir einen zweiten Parameter $\beta \geq 1$. Dieser beschränkt die Verhältnisse zwischen den Werten für $\mathcal{S}$ und $\mathcal{R}$ von zwei verschiedenen Typen. Intuitiv bedeutet diese Parametrisierung, dass ein Typ, der besser für $\mathcal{S}$ ist als ein anderer Typ, auch besser für $\mathcal{R}$ sein muss (bis auf einen Faktor β). Hierfür erreichen wir einen Approximationsfaktor von mindestens $\Omega\left(\frac{1}{\log\beta}\right)$. Analog zur oberen Schranke für Parameter α gilt auch hier eine obere Schranke von $O\left(\frac{\log\log\beta}{\log\beta}\right)$.

Zusätzlich betrachten wir noch den Fall, dass $\mathcal{R}$ statt des kompletten Typs bei einer Empfehlung nur den eigenen Wert erfährt. Weil verschiedene Typen denselben Wert für $\mathcal{R}$ haben können, verkompliziert diese Annahme den Aufbau möglicher Akzeptanzschemata, sodass auch die erreichbaren Approximationsfaktoren sinken. Wir zeigen eine Approximationsgarantie von $\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$ für Parameter α und einen Approximationsfaktor von $\Omega\left({}^{1}\!/{}_{\beta}\right)$ für Parameter β. Die oberen Schranken sind jeweils $O\left({}^{1}\!/{}_{\sqrt{\alpha}}\right)$ beziehungsweise $O\left({}^{1}\!/{}_{\sqrt{\beta}}\right)$.

Fazit

In dieser Dissertation betrachten wir verschiedene Szenarien von strategischer Kommunikation zwischen zwei Parteien mit Commitment Power. Im ersten Teil legen wir den Fokus auf Commitment Power bei der sendenden Partei. Diese besitzt Zugang zu Informationen über den Zustand der Welt, die der empfangenden Partei nicht zur Verfügung stehen. Wir beschreiben eine Vielzahl von Signalschemata beziehungsweise Algorithmen zur Bestimmung derselben, mit denen $\mathcal{S}$ die zusätzlichen Informationen ausnutzen kann, um ein gutes Ergebnis für sich selbst zu garantieren. Insbesondere bedeutet dies für $\mathcal{R}$, dass das zu erwartende Ergebnis nicht (viel) besser ist als eine Wahl einer Aktion ohne die Signale von $\mathcal{S}$, falls die Interessen der beiden Parteien nicht ähnlicher Natur sind. Eine bemerkenswerte Ausnahme existiert allerdings auch hierzu im Online-Fall mit zufälliger Reihenfolge und a priori unbekannten Werten sowie Aufdeckung von abgelehnten Typen: Wenn $\mathcal{R}$ den Erwartungswert der gewählten Aktion maximieren möchte, erreicht $\mathcal{S}$ im Allgemeinen nur einen Wert von $\Theta\left(1/n\right)$. Hier kann $\mathcal{S}$ die zusätzliche Information also nicht ausnutzen. Für den Fall, dass $\mathcal{R}$ die Wahrscheinlichkeit maximieren möchte, den besten Typen auszuwählen, gibt es allerdings einen komplett symmetrischen Mechanismus. Dieser garantiert sowohl $\mathcal{S}$ als auch $\mathcal{R}$ eine Erfolgswahrscheinlichkeit, den jeweiligen besten Aktionstypen zu erhalten von $1/4$. In diesem Fall arbeiten beide Parteien in gewissem Sinne zusammen, um eine gute Approximation zu erhalten.

Im zweiten Teil betrachten wir umgekehrt den Fall, dass $\mathcal{R}$ Commitment Power besitzt. Dies kann entsprechend $\mathcal{S}$ ungemein beschränken, da die Einschränkungen durch das Akzeptanzschema möglicherweise genau die guten Typen für $\mathcal{S}$ aussortieren. Letztendlich kann $\mathcal{S}$ zwar frei optimieren, allerdings nur im durch $\mathcal{R}$ vorgegebenen Rahmen. Wie wir allerdings zeigen, kann $\mathcal{R}$ nicht unbedingt von der Commitment Power profitieren. Durch die Schranke von $\Theta\left(1/n\right)$ im Allgemeinen wird klar, dass es sich für $\mathcal{R}$ durchaus lohnen kann, selbst zu suchen. Insbesondere wird auch die untere Schranke von $\Omega\left(1/n\right)$ dadurch erreicht, die beste Aktion a priori auszuwählen. Hierzu würde kein „Experte“ mit Information über den Zustand der Welt gebraucht. Allerdings gelingt es uns mithilfe einer natürlichen Parametrisierung, bessere Resultate für $\mathcal{R}$ zu erzielen. Wenn die Werte für $\mathcal{S}$ beschränkt sind (Parameter α) oder gute Optionen für $\mathcal{S}$ auch nicht sehr schlecht für $\mathcal{R}$ sind (Parameter β), gelingt es, logarithmische Approximationsfaktoren in Abhängigkeit der jeweiligen Parameter zu erzielen.
Zusätzlich betrachten wir ein Szenario, in dem $\mathcal{R}$ weniger Informationen erhält. In diesem wird nicht der komplette Typ der vorgeschlagenen Aktion offengelegt, sondern nur der Wert für $\mathcal{R}$. Dies bedeutet, dass in einer Runde alle Typen mit demselben Wert für $\mathcal{R}$ ununterscheidbar sind. Die Schranke von $\Theta(1/n)$ gilt weiterhin, für Parameter α und β sind nun jedoch nur noch polynomielle Approximationsfaktoren erreichbar.

Insgesamt gelingt es uns für fast alle Fälle von strategischer Kommunikation, optimale oder beinahe optimale Algorithmen zu beschreiben, die die jeweiligen Probleme effizient lösen können. In den meisten Fällen können wir zusätzlich zeigen, dass diese Algorithmen gute Approximationsfaktoren bezüglich natürlicher Richtwerte garantieren. In fast allen Fällen gelingt es jedoch nur der Partei mit Commitment Power, ein gutes Ergebnis zu erzielen. Auch wenn die andere Partei jeweils versucht, das bestmögliche für sich selbst zu erreichen, so gibt es keine Garantien, die unabhängig von der jeweiligen Instanz sind.

Acknowledgments

First and foremost, I want to thank my advisor Martin Hoefer. I would like to thank him for introducing me to several interesting research problems, e.g., the domain of Bayesian persuasion, which inspired this thesis. At the same time, he always gave me the freedom to explore my own interests. All throughout my time as PhD student, I was able to count on his support and I knew that any questions I might have would always be answered immediately.

Additionally, I would like to thank him for introducing me to our co-authors Rann Smorodinsky, and, by extension, Ronen Gradwohl. The research visits both in Frankfurt as well as Haifa and Tel Aviv always resulted in answers for research questions on which I had been working and new ideas for questions to pursue. Working with Rann and Ronen broadened my horizons, both in an academic as well as a personal capacity. Thank you for the interesting discussions we had and the opportunity to experience parts of Israel.

I am grateful for everyone that I have met on my journey, not least all the people in our group. Thank you to Daniel, Giovanna, Jutta, Lisa, Marco, Paresh, Steffen, and Tim. I truly enjoyed our interesting talks about current research problems or life in general.

Finally, I want to express my gratitude toward my friends and family for always being there for me. Most of all, I am deeply grateful to my parents, Andrea and Matthias, for allowing me to become who I am and always respecting and supporting my choices.

Contents

Chapter 1

Introduction

In this thesis, we consider algorithmic aspects of strategic communication with commitment power in varying scenarios. The characteristic property of this domain is the interaction between two selfish, rational agents with individual objectives, where one of the agents has an informational advantage. However, for the overall outcome, the other agent plays a key role. We consider efficient algorithms that compute (near-)optimal schemes of action for one of the agents, where we take the perspective of both agents in the respective parts of the thesis.

For an example of strategic communication, consider a customer shopping for a car. Clearly, the employees of the retailer know more about the different cars that are offered. Hence, a sales representative might try and persuade the customer to purchase a specific car, highlighting some of the qualities of that particular car while at the same time concealing that a different, cheaper model meets the requirements of the customer. In the long term, the satisfaction of the customers affects the reputation of the retailer. Hence, it should be credible that the advice given by the sales representatives does indeed help potential customers in choosing a car that fits their specific needs. Otherwise, the retailer would not sell any cars and thus would not be able to make any profits.

Another example is the delegation of a task to an expert. Consider a client seeking advice regarding investments. The client consults an expert to give a recommendation. The expert might receive a provision when selling certain products and will not (only) have the client's financial well-being in mind when offering potential investment opportunities. Still, without the client's investment, neither the expert nor the client will receive any profit. Hence, if the client has some reservations towards some specific financial products, e.g. investments in fossil fuels, and decides not to invest in any of them, the expert's search space is somewhat restricted. Oftentimes, investment opportunities appear on a short notice and require a quick reaction before they are gone for good. We model this dynamic using an online framework, i.e., actions are observed sequentially in a round-wise fashion and a decision has to be made immediately and irrevocably.

The two examples above represent the two main models which we study in this thesis, namely *Bayesian persuasion* (first example) and *delegated search* (second example). In both scenarios, there are two agents who interact with each other in order to increase their respective utility. The final decision which determines the utility for both agents is made by only one of the agents, who has to take one of n possible actions with a priori unknown utility values. For simplicity, the utility for each agent is expressed by a non-negative real number. Clearly, the utility does not have to represent a (solely) monetary value. In contrast to the agent making the final decision, the other agent is able to observe information on the state of nature, i.e., the realizations of the actions and their utility values. Hence, both agents are trying to leverage their own position in a hope to maximize their individual outcome.

Another important factor in both examples is the existence of commitment power. In the first example, the sales representatives need to commit to providing helpful advice in order to keep a good reputation. In the second one, the client commits to rejecting some investment products, thereby restricting the set of potential investments. Without the power to commit, the communication problem becomes an instance of *Cheap Talk* [30, 45, 46]. Although interesting in its own right, the discussion of cheap talk is outside the scope of this thesis. Instead, we focus on Bayesian persuasion and delegated search. In both scenarios, commitment power can strictly increase the utility for the agent who holds it. It is clear that agents can weakly increase their expected utility as they could commit not to use their commitment power. The following example shows a scenario in which a strict increase in utility is possible due to commitment power. It is adopted from [44].

Consider the following situation. During a pandemic, the government is concerned with public health and considers shutting down public life. Clearly, this drastic step should only be taken if it increases public health. An expert who is tasked with evaluating the situation is more concerned with the economic impact the pandemic might have on businesses. If it is profitable for shops to stay open, the expert opposes a shutdown, regardless of the implications for public health. If, on the other hand, keeping the shops open negatively affects the economy, the expert favors a strict lockdown, even if the effect of the pandemic on public life in general is negligible. Hence, there are two different actions, imposing a lockdown (L) or keeping everything open (O). There are four different states of nature, namely all combinations $\{H, \bar{H}\} \times \{B, \bar{B}\}$. Here, H means that public health is improved if a lockdown is imposed and B represents the case that keeping businesses open is profitable. $\bar{H}$ and $\bar{B}$ represent the opposites of H and B, respectively. The utilities for government and expert when action L is taken as well as the probabilities for the states are given in the following table.

State	(H, B)	$(\bar{H}, B)$	$(H, \bar{B})$	$(\bar{H}, \bar{B})$
Probability	0.2	0.3	0.3	0.2
Utility Gov.	1	0	1	0
Utility Exp.	1	1	0	0

Table 1: Utility values for action L

For action O, the utilities are $1 - u$, where u is the corresponding utility for action L. Without commitment power, the only equilibrium is the so-called babbling equilibrium, i.e., the expert not revealing any information and the government using only the prior to come to a conclusion. This gives both an expected utility of 0.5. If, on the other hand, the expert does have commitment power, the following scheme is strictly more profitable for the expert while not decreasing the expected utility for the government. Whenever both expert and government have aligned interests, the expert recommends the optimal action for both. When interests are misaligned, the expert recommends the better option for the government only with probability $1/6$. Hence, the state is H conditional on a recommendation for L with probability $1/2$. Analogously, the state is $\bar{H}$ conditional on a recommendation for action O with probability $1/2$. This implies that the government wants to follow the recommendation, getting a utility of 0.5. This leads to a utility of 0.9 for the expert.

For the case of delegated search, if the government has commitment power in the same example, a restriction to the action that is profitable for the government always ensures a utility of 1 for the government, thereby increasing the expected utility to 1.

Clearly, in real-world recommendation systems, the recommendations oftentimes only reflect a small subset of possible actions. For example, recommendation systems for restaurants often use a fixed number of stars, e.g. 5 stars. Most consumers take this system as a binary recommendation. If the restaurant has more than x stars, it is considered good, otherwise, one probably should not eat there. Clearly, this does not differentiate between individual meals being offered or whether the chefs preparing lunch are better or worse than those preparing dinner. To model this, we use restricted signal spaces for some of the Bayesian persuasion instances we study.

Similar to the online model for delegated search, we also study an online version of Bayesian persuasion to model dynamic recommendations. As an example, one could think of social media feeds showing users new posts of friends or influencers. Whenever a new post is published, the system has to decide on the fly whether to show it to a user, trying to increase the likelihood of engagement to generate ad revenue. If the user is not interested in the post and "rejects" it, a new piece of content has to be offered immediately. Otherwise, if the user decides to view the piece, the system will not recommend new content (at least for a short time).

1.1 Overview

After a review of related literature in the next section, we discuss some general preliminaries in Chapter 2. Here, we define the basic models. In the next chapters we then go into detail on the main contributions of this thesis. The content is further subdivided into two main parts. In the first part, we discuss variants of *Bayesian persuasion*, in the second part, we study *delegated search*. The main difference between these two parts lies in who has the power to commit. In the first part on Bayesian persuasion, the agent with informational advantage, the *sender*, commits to a *signaling scheme*. Such a scheme specifies which signals can be sent, depending on the observed state of nature as well as (possibly) some internal randomization. In the second part on delegated search, the other agent, the *receiver*, is the one to commit to an *acceptance scheme*. An acceptance scheme determines the reaction by the receiver upon seeing a proposal, i.e., whether to accept or reject the corresponding action. Note that acceptance schemes can also include randomization. In both settings, we adopt the perspective of the agent with commitment power and describe mechanisms that optimize their respective utility.

1.1.1 Bayesian Persuasion

We start by discussing an offline case of Bayesian persuasion in Chapter 3, where we focus on scenarios with limited signal spaces. Consider a scenario with a noisy communication channel between the two agents. To ensure that messages can still be understood as intended, this requires some redundant information in the signals, effectively limiting the number of bits that can be transferred without noise. We build on hardness results by Dughmi, Kempe and Qiang [34] and Dughmi and Xu [36]. In [34], the authors show that it is $\mathcal{NP}$-hard to determine a signaling scheme which approximates the optimal utility within a constant factor when the signal space is limited. Even when the signal space is unconstrained, in the case of independently drawn types, it is $\#\mathcal{P}$-hard to compute the optimal expected utility as shown in [36]. For instances with types drawn IID, i.e., all drawn independently from a single distribution, the authors provide a positive result and give a polynomially-sized linear program to compute the optimal signaling scheme. We extend these results with our polynomial-time algorithms which compute (near-)optimal signaling schemes under some conditions.

First, we focus on symmetric instances. We describe a polynomial-time algorithm which determines an optimal signaling scheme for symmetric instances with a succinct representation, generalizing the IID scenario. For each individual state of nature, the algorithm utilizes the geometric interpretation of that state to determine the optimal signals. This approach works even in the setting of constrained communication.

Secondly, we study independent instances with limited signal spaces of size k and identify a condition which allows for constant-factor approximation algorithms in polynomial time. We describe two algorithms, both of which consist of two steps. In step one, a good subset S of k actions is identified and in step two, a good signaling scheme for S is computed. We describe two different approaches for step one; the second step remains unchanged for both algorithms. At first, we consider the classic greedy approach to submodular optimization which guarantees at least a $(1-{}^1\!/\!_e)$-approximation. Secondly, we use a more elaborate knapsack-style FPTAS to compute the set S. This way, for every constant $\varepsilon > 0$, the resulting set S guarantees a $(1-\varepsilon)$-approximation to the optimal set S^* in polynomial time. In step two, we use a linear program to determine a signaling scheme for the actions in S. Here, an additional factor of at most $1 - {}^1\!/\!_e$ is lost. We conclude the chapter on offline persuasion with performance guarantees for instances with a limited signal space in comparison to their unlimited counterparts.

In Chapter 4, we study Bayesian persuasion with an online aspect. The setup in the first section is reminiscent of the classic *prophet inequality* problem [60, 61], extended to the two-dimensional setting of persuasion. Types are sequentially drawn from known distributions in a predetermined order and the problem becomes finding the optimal stopping time. The first result is a positive one: for every set of distributions, the optimal signaling scheme can be computed in polynomial time using backwards induction. Unfortunately, there are instances in which even the optimal online scheme cannot guarantee a positive approximation to the expected utility obtainable by a prophet sender, i.e., using a signaling scheme which has access to all realizations. In contrast to this negative result, we identify a condition which allows a simple constant-factor approximation in Section 4.1.3.

A different online setting is studied in Section 4.2. Here, we consider a setting with a random-order assumption similar to the classic *secretary* problem [38]. We study this setting in 16 different variants with different levels of knowledge as well as different objective functions for both agents. We distinguish a *cardinal* and an *ordinal* objective for each agent. In the cardinal setting, the agents want to maximize their respective expected utility. In the ordinal case, they only derive utility from the best possible type and thus want to maximize their respective success probability. As a *basic setting* and our benchmark, we assume that all types are known a priori, and only the order in which they are observed is a priori unknown. Additionally, we consider our *secretary setting* without prior knowledge of the types. Finally, we study both settings in two variants, namely with and without *disclosure* of rejected actions. In a scenario with disclosure, the receiver is notified about the type of the action that was rejected in the previous round. Analogously, without disclosure, the receiver does not know which types have already been rejected. For the cases in which the receiver wants to optimize the cardinal utility, we use a geometric interpretation of the types to describe our mechanisms. For all but the most challenging secretary variant with disclosure, these mechanisms guarantee the sender a constant-factor approximation of the utility obtainable in the benchmark setting. In many of the settings, we provide asymptotically matching upper bounds. In contrast to constant-factor approximations in all other scenarios, in the secretary setting with disclosure, there are instances in which no signaling scheme can guarantee more than a $\Theta\left(1/n\right)$-approximation.

For ordinal receiver utility, we describe mechanisms that achieve a constant-factor approximation for all variants, and we are able to provide asymptotically matching upper bounds for all scenarios. Additionally, we can show that in the secretary scenario without disclosure, the sender is able to asymptotically match the optimal utility of the one-dimensional secretary scenario, i.e., the sender gets the best type with a probability of $1/e$. In contrast to the case of cardinal receiver utility, even in the secretary scenario with disclosure, the sender can achieve a constant factor of $1/4$ using a simple mechanism. The mechanism simply recommends the first action after a sample phase which has a type that is better than all previously observed ones, either for the sender or for the receiver.

1.1.2 Delegated Search

We discuss an extension of the delegated search model studied by Kleinberg and Kleinberg [58] to online algorithms in Chapter 5. They used tools and results from the realms of the online stopping problem and prophet inequalities to achieve constant-factor approximation algorithms for the offline delegated search problem, specifically for IID instances. The benchmark for these algorithms is the best type for the receiver in hindsight, or, equivalently, the result of a one-dimensional search performed by the receiver. We extend this model to online optimization. We study a sender who observes the actions in an online fashion and has to decide on the spot whether to propose the current one. Upon a proposal, the receiver learns the type of the proposed action and accepts or rejects the action based on the pre-determined acceptance scheme.

Our first result is that the constant-factor approximation does not extend to this online variant. Instead, we show a tight bound of $\Theta\left(1/n\right)$ using an IID instance. Since this worst-case instance is using receiver values that are exponential in n, we consider two different natural parameters for the problem. First, we assume that the sender's utility is bounded by $\alpha \geq 1$, i.e., the ratio of any pair of values for $\mathcal{S}$ is at most α. For such instances, we describe an algorithm which computes an acceptance scheme that guarantees the receiver an $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$-approximation with respect to the best value in hindsight. Additionally, we study parameter $\beta \geq 1$ which limits the ratios of the sender-values and the receiver-values of two types. Intuitively, if type θ is better than another type θ' for one of the agents, then θ is also better than θ' for the other agent (up to a factor β). For instances satisfying the condition for parameter β, our algorithm determines an acceptance scheme guaranteeing an $\Omega\left(\frac{1}{\log\beta}\right)$-approximation. For both settings, the upper bound applies, i.e., if α or β are in the order of n^n, no acceptance scheme can guarantee an approximation better than $O\left(\frac{\log\log\alpha}{\log\alpha}\right)$ or $O\left(\frac{\log\log\beta}{\log\beta}\right)$, respectively. Finally, we study both settings with limited informational gain for the receiver. More formally, we assume that a proposal only reveals the receiver value. This means that types with the same value for the receiver cannot be distinguished. For this limited information case, we can show an $\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$-approximation for instances with parameter α and an $\Omega\left({}^{1}/_{\beta}\right)$-approximation for instances with parameter β. In terms of upper bounds, we give an instance which proves that no better approximation ratio than $O\left({}^{1}/_{\sqrt{\alpha}}\right)$ or $O\left({}^{1}/_{\sqrt{\beta}}\right)$ exists.

1.2 Related Work

The study of strategic communication and information design has a rich history, especially in economics. In recent years, research focused on algorithmic aspects of strategic communication has enjoyed increased attention.

1.2.1 Bayesian Persuasion

The study of Bayesian persuasion has originally been established in 1966 by Aumann and Maschler [12], where they discussed repeated games with incomplete information in the context of arms control negotiations during the Cold War. Interest in the topic experienced a resurgence following the seminal work by Kamenica and Gentzkow [55] in 2011. Since then, the concept was studied in a plethora of variants and for several different applications. Overviews are given by [22, 54, 42, 24]. Exemplary applications include the work on advertisement from the perspective of a seller [9, 27, 62, 32] or an auctioneer [41, 17] as the sender, trying to maximize their revenue, in various settings. Bergemann et al. [21] consider a different approach. In their paper, the seller is able to set discriminatory prices depending on the buyers' types. A central planner, who optimizes some weighted sum of the seller's and the buyers' surplus, plays the role of the sender and decides accordingly what information about the buyers to reveal.

Clearly, persuasion is prevalent in settings where no direct monetary consequences follow. Alonso and Câmara [4] consider politicians trying to influence voters. Alizamir et al. [3] studied how a public organization with early access to information on recurring harmful events can efficiently inform members on the severity of upcoming crises in order to solicit adequate responses. Similarly, in [31], this setting was studied on a more local level, with a government weighing the implications on the economy as well as public health when issuing warnings to the citizens.

Stackelberg security games with signaling were studied in [68, 74, 73], showing that partial information disclosure can be beneficial for the defender, the leader of the Stackelberg game.

While the previously discussed literature focuses on a static version of Bayesian persuasion, there are also a lot of dynamic settings in the literature, which are loosely related to our online Bayesian persuasion models. From the perspective of platforms offering recommendations (e.g., for restaurants), enough consumers need to be incentivized to be in exploration mode, i.e., be willing to try venues that do not have a sufficient number of evaluations. This scenario was studied as an instance of Bayesian persuasion by [67]. Hence, this leads to some obfuscation of knowledge by the platform. Similarly, the problem whether to join an unobservable queue for a service was studied in [64, 8]. In [64], service is offered at a fixed price and potential customers are sensitive to delays. Yet, the service provider wants to persuade as many customers as possible to join the queue in order to maximize revenue. In contrast, a social service is offered in the setting of [8]. Here, social welfare is to be maximized. Due to the limited capacities, members of society who can afford an external option should sometimes be discouraged from joining the queue.

Ely [39] studies the case of an evolving state of nature. Similar to our online setting, the sender gets to observe the current state and signals the receiver afterwards, who then takes an action according to the updated beliefs. While our model of online persuasion might be cast in the framework of having a state that evolves over time, the specific details of our online model would be very different from Ely's approach using Poisson transitions over states. In Au [11], the sender tries to convince the receiver to take some action, which is the same throughout all rounds. Unlike the receiver's utility, which depends on the non-changing state of nature, the sender's utility is only dependent on the receiver taking the action. Rather than using a single signaling scheme, the sender designs a signaling policy for each round. Renault et al. [69] study a Bayesian persuasion instance where a financial advisor tries to persuade an investor to repeatedly take a short-lived but risky investment. The advisor's fee when the investor takes the risky option is state-independent. The receiver's payoff depends on the state of nature which evolves according to an irreducible Markov chain. In [40], the notions of suspense and surprise are studied, where suspense is the variance in the next update of belief and surprise the distance of two consecutive beliefs. The sender uses signals designed on these two objectives for a finite number of steps.

More loosely related is the recent work on online learning and Bayesian persuasion. In the model of [25], the sender is not aware of the receiver's type. Rather, the sender uses online learning to dynamically improve the signaling scheme and minimize regret. The model was subsequently extended to accommodate multiple receivers in [26]. In a related paper, Zu et al. [76] study a scenario in which neither sender nor receiver know the prior distribution. In each round, the sender observes the realized type. Using this information, the sender tries to persuade the receiver and learn the underlying distribution at the same time. A different perspective of regret minimization was studied in [16], where the sender does not know the receiver's utility. Here, the process is not dynamic. Instead, there is only a single exchange and the regret is measured against the utility the sender could have achieved with knowledge of the receiver's utility.

In the settings described above, the classic setup of a single sender and a single receiver who interact with each other is not always applicable. An often-used setting is that of a single sender who interacts with multiple receivers using public or private signals. Clearly, private signals allow the sender more flexibility. Algorithmic aspects of private persuasion have been studied in [9, 14]. The implications of public and private signals in an otherwise unchanged setup were studied by Dughmi and Xu [37]. Recently, Babichenko et al. [15] further diversified this approach and studied multi-channel signaling, interpolating between entirely public and strictly private signals.

For further work on algorithmic aspects of Bayesian persuasion, the survey by Dughmi [33] offers a good starting point. Bhaskar et al. [23] and Rubinstein [70] study scenarios in which the receivers are players in games, proving various hardness results. These results are complemented by efficient approximation algorithms for some subclasses of these scenarios in [72]. Dughmi et al. [35] describe (near-)optimal persuasion schemes using Lagrangian duality. For some of their scenarios, they assume symmetry of the actions that is similar to our symmetric instances.

Other important results for algorithmic Bayesian persuasion with strong ties to this thesis include [36] and [34]. Dughmi and Xu [36] give a linear program that optimally solves offline Bayesian persuasion. Unfortunately, this linear program has an exponential size when the distribution over states of nature is not explicitly given, e.g., when types are drawn independently from different distributions. For the IID setting, they are able to leverage symmetry and reduce the size of the linear program to a polynomial one. Further, they show $\#\mathcal{P}$-hardness of calculating the optimal sender utility for independent actions. In the thesis, we broaden the class of symmetric instances with optimal schemes that can be computed in polynomial time.

Dughmi et al. [34] showed another hardness result for independent instances, namely $\mathcal{NP}$-hardness to compute a scheme which approximates the optimal sender utility to within a constant factor if the signal space is limited. Contrasting this negative result, we identify a condition for independent instances which allows a constant-factor approximation in polynomial time.

In [13], Aybas and Turkel study a similar setting with a limited signal space. They show that the sender loses at most a factor $2/k$ of the expected utility when the number of signals decreases from k to $k-1$. For symmetric instances, we show that using k instead of n signals, the sender can still extract a utility of at least k/n and this is tight. Asymptotically, we are also able to show this result for independent instances satisfying our condition.

Somewhat related is the paper of Le Treust and Tomala [63], who study a repeated setting with limited communication through a noisy channel.

1.2.2 Delegated Search

The study of delegated search goes back to Holmström's PhD thesis in 1977 [52] and subsequent work of his [53]. Holmström studied the bi-level optimization problem between the receiver (called the principal in Holmström's model) and the sender (simply called agent) on an interval. Here, the receiver is able to commit to a subset of the search space, allowing the sender to freely optimize in this restricted space. Holmström identified sufficient conditions for an optimal solution to exist. The impact of varying degrees of (mis-)alignment between the receiver and sender was further studied in [65]. The same objective, i.e., alignment of receiver and sender, was further considered for more general distributions and utility functions in [5].

Instead of strictly disallowing choices, costs for certain options were considered in [6, 7]. Note that these costs do not have to be monetary. Instead, the receiver might impose strict bureaucratic requirements for the choice of some options. Similarly, Armstrong and Vickers [10] studied a model where the receiver is able to pay the sender to choose a more receiver-favorable solution. Additionally, they consider a scenario in which the sender has to exert effort to even find an option. They describe sufficient conditions for optimal solutions to exist in these models. A communication model with evidence, i.e., verifiable facts intrinsic to the different states of nature, was recently studied by Hoefer et al. [51]. Here, they compare the setting of delegation with that of Bayesian persuasion as well as cheap talk, i.e., communication without commitment power.

Most closely related to our online delegated search model is the paper by Kleinberg and Kleinberg [58]. They study an offline IID scenario where n types are drawn from a known distribution. As benchmark, they use the optimal receiver utility, i.e., the best realization of the n draws for the receiver. Via a reduction of this offline delegation problem to the online stopping problem of prophet inequalities, they are able to show constant-factor approximations with respect to this benchmark for the receiver. We extend this model to the purely online case, i.e., the sender can only react to the current realization.

Another related paper which was inspired by [58] is the work of Bechtel and Dughmi [19]. They combined delegated search with stochastic probing, which means that instead of observing a fixed set of types, the sender needs to actively probe elements to observe their types. The subset of elements that can be probed is subject to some constraints. Although the sender is able to select multiple elements from the probed ones, there is a second set of constraints which defines the feasible solutions. For several downwards-closed constraint systems, they are able to provide constant-factor approximation algorithms, also using tools and techniques borrowed from the literature on prophet inequalities.

Chapter 2

Preliminaries

In this chapter, we discuss the main underlying models and introduce some general notation.

2.1 Model

In all variants of strategic communication we consider in this thesis, the following notations are used. There are two rational agents, the *sender* $\mathcal{S}$ and the *receiver* $\mathcal{R}$. Ultimately, $\mathcal{R}$ chooses one of n possible *actions* which we denote by the set $[n] = \{1, \ldots, n\}$. Each action i has some type θ_i. The vector of all n types $(\theta_1, \ldots, \theta_n)$ is called the *state of nature* $\boldsymbol{\theta}$. The state of nature is drawn from an abstract state space Θ according to a distribution q. If action $i \in [n]$ is chosen, the corresponding type θ_i yields some nonnegative utility $\varrho(\theta_i) \geq 0$ to $\mathcal{R}$ and $\xi(\theta_i) \geq 0$ to $\mathcal{S}$. The sender will have an informational advantage over $\mathcal{R}$ with respect to the state of nature $\boldsymbol{\theta}$, depending on the setting. $\mathcal{S}$ will try to leverage this advantage by sending a signal $\sigma \in \Sigma$ to $\mathcal{R}$. We denote the size of the signal space by $k = |\Sigma|$.

Clearly, both agents strive to maximize their respective objective in expectation with respect to the randomization of the input as well as the internal randomization employed by $\mathcal{S}$ and $\mathcal{R}$.

2.1.1 Bayesian Persuasion

Before learning the state of nature, $\mathcal{S}$ commits to a signaling scheme $\varphi : \Theta \times \Sigma \to [0, 1]$, where $\varphi(\boldsymbol{\theta}, \sigma)$ denotes the probability that signal σ is sent when the state of nature is $\boldsymbol{\theta}$. For a scheme φ, we denote by $u_{\mathcal{S}}(\varphi)$ and $u_{\mathcal{R}}(\varphi)$ the expected utility for $\mathcal{S}$ and $\mathcal{R}$, respectively, when $\mathcal{S}$ uses φ to send recommendations and $\mathcal{R}$ best responds to φ. For simplicity, we assume the following. If there are several actions maximizing the receiver's conditional expected utility, $\mathcal{R}$ will choose an action that maximizes the sender's expected utility among these actions. We call a scheme φ *direct* if each signal $\sigma \in \Sigma$ directly corresponds to a distinct action $i \in [n]$. A scheme φ is *persuasive* if it is in the receiver's interest to follow the recommendation, i.e., given a signal σ, the conditional expected utility for $\mathcal{R}$ when following the recommendation is at least as high as the conditional expected utility for $\mathcal{R}$ when deviating from the recommendation and choosing a different action. A useful quantity when considering persuasiveness will be the value $\varrho_E := \max_{i \in [n]} \sum_{\boldsymbol{\theta} \in \Theta} q_{\boldsymbol{\theta}} \cdot \varrho(\theta_i)$, which denotes the highest a priori expectation of any of the n actions. Obviously, each persuasive scheme needs to guarantee an expected utility of at least ϱ_E to $\mathcal{R}$.

We now show that it is without loss to only consider direct and persuasive schemes.

Lemma 2.1
There exists an optimal scheme with k signals that is direct and persuasive and uses the signals to recommend k distinct actions.

Proof. We assume that both $\mathcal{S}$ and $\mathcal{R}$ act rationally. Hence, both want to maximize their respective expected utility. Now, consider an arbitrary signaling scheme φ. Having seen the signal σ, $\mathcal{R}$ is then able to perform a Bayesian update and choose an action maximizing the expected utility conditional on having received σ. If there are two signals σ, σ' for which $\mathcal{R}$ chooses the same action, the sender can simply send σ instead of σ' to achieve the same outcome. Hence, we can assume that each of the signals corresponds to a single distinct action, i.e., one of the actions maximizing the receiver's conditional expected utility. This means that $\mathcal{S}$ can simply recommend that action. □

Throughout this thesis, when discussing Bayesian persuasion, we adopt the perspective of $\mathcal{S}$. Hence, all signaling schemes we describe maximize the sender's objective function. To avoid technicalities, we assume that $\mathcal{R}$ breaks ties in favor of $\mathcal{S}$. Using slight perturbations of the signaling probabilities, the sender can always sacrifice a negligible portion of expected utility to ensure that no ties exist and the better choice for $\mathcal{S}$ is also (slightly) beneficial for $\mathcal{R}$.[1] Hence, even a receiver using a different tie-breaking rule would choose in favor of $\mathcal{S}$.

2.1.2 Delegated Search

In contrast to Bayesian persuasion, in delegated search $\mathcal{R}$ is the one with commitment power. Before $\mathcal{S}$ observes the state of nature $\boldsymbol{\theta}$, $\mathcal{R}$ commits to a mechanism, consisting of a set Σ of signals $\mathcal{S}$ may send as well as an allocation function specifying which action $\mathcal{R}$ will choose when a certain signal is sent. Clearly, this requires $\mathcal{R}$ to know the possible types a priori. This is the case in the model of delegated search we consider, where n finite distributions $\mathcal{D}_i = (\Theta_i, \boldsymbol{q}_i)$ for $i \in [n]$ are given and known a priori by $\mathcal{R}$ and $\mathcal{S}$. For simplicity, we assume that $|\Theta_i| = m$ and denote the types by $\{\theta_{i1}, \theta_{i2}, \ldots, \theta_{im}\}$ for all $i \in n$. Note that this assumption is without loss of generality as dummy types with probability 0 can be added to a distribution with fewer than m types. The state of nature $\boldsymbol{\theta}$ then consists of the types of the n actions, where type θ_i of action i is drawn independently from $\mathcal{D}_i$. For a shorter notation, we will sometimes write ϱ_{ij} and ξ_{ij} instead of $\varrho(\theta_{ij})$ and $\xi(\theta_{ij})$ to denote the utility for $\mathcal{R}$ and $\mathcal{S}$ of type j from distribution $\mathcal{D}_i$.

For the delegated search model studied by Kleinberg and Kleinberg [58], they show that it is without loss of generality to consider only *single proposal mechanisms*, which is a translation of direct signaling schemes as in Lemma 2.1 to the realm of delegated search. In a single proposal mechanism, $\mathcal{R}$ restricts the set of types to a subset of

[1]The only special case where this procedure does not work is the case that all types provide no utility for $\mathcal{R}$. Regardless of the signaling scheme, the receiver's expected utility will be 0. For this special case, we assume for simplicity that $\mathcal{R}$ still breaks ties in favor of $\mathcal{S}$.

acceptable ones. $\mathcal{S}$ is then able to propose (only) one of the acceptable types which $\mathcal{R}$ takes.

We extend this model in two ways. First and foremost, we consider an online version of the delegated search problem. The type of action i is drawn in round i from distribution $\mathcal{D}_i$. $\mathcal{S}$ has to decide immediately and irrevocably what kind of signal to send upon observing the type, i.e., whether to propose the current action or not. To capture the essence of the online model for the sender, we assume that the process ends after the first proposal, regardless of the decision by the receiver.

Secondly, we extend the notion of single proposal mechanisms to allow for randomized acceptance schemes. Formally, we model this in the following way. $\mathcal{R}$ commits to an acceptance scheme φ, which maps to each possible type a probability that $\mathcal{R}$ accepts an action with that type. Hence, $\varphi = (\varphi_{ij})_{i\in[n],j\in m}$ with $\varphi_{ij} \in [0,1]$. Upon a proposal by $\mathcal{S}$, $\mathcal{R}$ learns the type of the proposed action and accepts or rejects that action according to φ. This allows $\mathcal{S}$ to maximize the expected utility with respect to the acceptance scheme. For $\mathcal{R}$, this means that the goal is to restrict the sender's search space effectively. The acceptance scheme should be permissive enough such that $\mathcal{S}$ eventually proposes an action but at the same time restrictive enough such that $\mathcal{S}$ cannot propose any action with an arbitrary type which might be bad for $\mathcal{R}$. If no action is proposed or $\mathcal{R}$ rejects the proposed action, both $\mathcal{R}$ and $\mathcal{S}$ get a utility of 0.

For delegated search, all mechanisms we discuss are constructed with the receiver's utility in mind. Since $\mathcal{R}$ is the agent with commitment power, this is analogous to the discussion of Bayesian persuasion. Hence, we further assume that $\mathcal{S}$ breaks ties in favor of $\mathcal{R}$ in our delegated search settings. Similar to the Bayesian persuasion setting, slightly perturbing the acceptance probabilities can always incentivize the sender to choose in favor of $\mathcal{R}$ at a negligible cost to the overall expected utility for $\mathcal{R}$.[2]

[2] Similar to the Bayesian persuasion setting, no perturbation of acceptance probabilities can incentivize $\mathcal{S}$ to optimize for $\mathcal{R}$ when all (acceptable) values for $\mathcal{S}$ are 0. Again, we assume for simplicity that $\mathcal{S}$ still breaks ties in favor of $\mathcal{R}$ in this special case.

Chapter 3

Bayesian Persuasion

This chapter is based mainly on [43]. It focuses on an offline Bayesian persuasion setting. In such scenarios, the signaling process is the following.

1. Both $\mathcal{S}$ and $\mathcal{R}$ know the distribution over states of nature.
2. $\mathcal{S}$ commits to a signaling scheme φ.
3. $\mathcal{R}$ learns the scheme φ.
4. $\mathcal{S}$ learns the state of nature $\boldsymbol{\theta}$.
5. $\mathcal{S}$ sends a signal $\sigma \in \Sigma$ according to φ to $\mathcal{R}$.
6. $\mathcal{R}$ chooses an action.

As we have seen in Chapter 2, we can assume without loss of generality that the signaling scheme φ that is used by $\mathcal{S}$ is a direct scheme. Hence, n signals suffice to recommend any of the n actions. Later on in this chapter, we will also consider a more restrictive signaling space with only $k \leq n$ possible signals.

Let us start by giving a simple introductory example with $|\Sigma| \geq n$. Assume that the set $\mathcal{C}$ of types of the n actions is known to both $\mathcal{S}$ and $\mathcal{R}$ but the permutation π of the types is unknown. Let us further assume that π is drawn uniformly at random from the set of permutations of $[n]$.

Since each type has attached a nonnegative value for $\mathcal{R}$ and a nonnegative value for $\mathcal{S}$, we can interpret the state of nature as points on the two-dimensional plane as shown in Figure 1 below. Any point consisting of the expected utility for $\mathcal{S}$ and $\mathcal{R}$ is required to be inside the convex hull of the points. Since the values are known in advance to $\mathcal{R}$, by choosing an action uniformly at random, $\mathcal{R}$ can ensure an expected utility of at least ϱ_E by ignoring the sender's signal. Conversely, this means that the sender's signaling scheme φ needs to guarantee an expected utility of at least ϱ_E for $\mathcal{R}$ – otherwise, $\mathcal{R}$ would not follow the signal. Hence, when using a persuasive signaling scheme, the highest expected utility $\mathcal{S}$ can hope for lies on the intersection of the vertical line $\varrho = \varrho_E$ and the Pareto frontier. Graphically speaking, the Pareto frontier of an action type set $\mathcal{C}$ can be assumed to start from a type with largest sender utility with a horizontal line (possibly of length 0) with slope 0 and end at a type with largest receiver utility with a vertical line (again, possibly of length 0) with slope $-\infty$. Figure 1 shows the Pareto frontier for a small example.

	1	2	3	4	5	6	7	8	
ϱ	1	3	6	7	12	13	14	16	$\varrho_E = 9$
ξ	11	1	10	4	8	5	6	2	

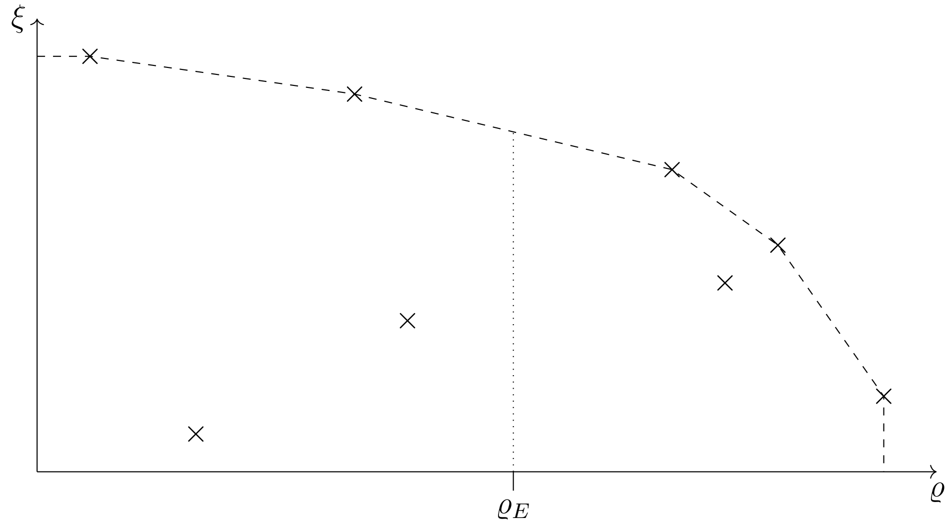


Figure 1: Geometric interpretation of the action type set $\mathcal{C}$ with the corresponding Pareto frontier as dashed line. The dotted line at ϱ_E represents the expected utility for $\mathcal{R}$ when drawing an action uniformly at random.

Our so-called *Basic Pareto Mechanism* (Algorithm 1) uses this geometric interpretation of the state of nature to achieve the optimal expected utility for $\mathcal{S}$. First, the mechanism computes ϱ_E, the a priori expectation for $\mathcal{R}$ from any action, and identifies $c_{\mathcal{S}}$, the type with the highest utility for $\mathcal{S}$. If $\varrho_{c_{\mathcal{S}}} \geq \varrho_E$, the mechanism will send a signal for the action with type $c_{\mathcal{S}}$. This means that $\mathcal{S}$ gets the highest possible utility and $\mathcal{R}$ enjoys an increased utility compared to the a priori expectation of ϱ_E. If, on the other hand, $\varrho_E > \varrho_{c_{\mathcal{S}}}$, the mechanism identifies two types a and b such that both types are on the line segment which intersects the line $\varrho = \varrho_E$. Then, using a convex combination $\alpha \cdot \varrho_a + (1-\alpha) \cdot \varrho_b = \varrho_E$, the mechanism determines the probability α to send a signal for the action with type a. With probability $1-\alpha$, the action with type b is recommended.

In Figure 2, we illustrate the mechanism using the same action types as in Figure 1.

Proposition 3.1
The Basic Pareto Mechanism is an optimal persuasive mechanism in the random-order scenario with known utility values. It can be computed in time polynomial in n.

Proof. We denote by $\boldsymbol{\xi}$ and $\boldsymbol{\varrho}$ the vectors of utilities for $\mathcal{S}$ and $\mathcal{R}$, respectively.

Consider the Basic Pareto Mechanism and the event that $\mathcal{R}$ gets a signal $\sigma = i$ to take action i. Clearly, the distribution $\boldsymbol{x}$ over types of action i is the same regardless of the value of i. Namely, type a has probability α, type b has probability $1-\alpha$ and all other types have probability 0. Similarly, for each action $i' \neq i$, the distribution $\boldsymbol{y}$ over types of action i' is the same, namely probability $\frac{1}{n-1}$ for all types $k \in [n] \setminus \{a, b\}$, probability $\frac{1-\alpha}{n-1}$ for type a, and probability $\frac{\alpha}{n-1}$ for type b.

	1	2	3	4	5	6	7	8	
ϱ	1	3	6	7	12	13	14	16	$\varrho_E = 9$
ξ	8	1	10	4	8	5	6	2	

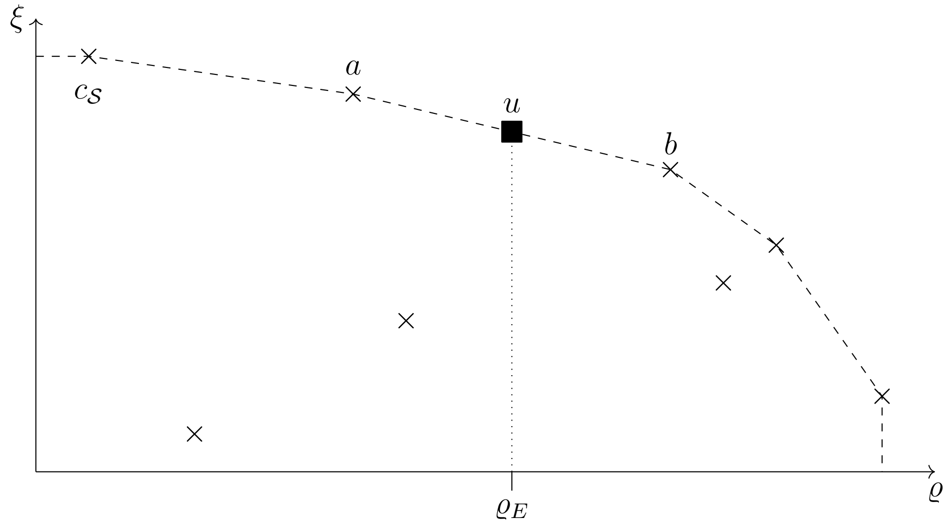


Figure 2: Visualization of the Basic Pareto Mechanism. Type $1 = c_{\mathcal{S}}$ has the best utility for $\mathcal{S}$, but provides less than ϱ_E to $\mathcal{R}$. Types $a = 3$ and $b = 5$ are the types identified in line 6 of the mechanism. Both a and b are recommended with a probability of $1/2$ each. The point $u = (u_{\mathcal{R}}, u_{\mathcal{S}})$ denotes the expected utility for $\mathcal{R}$ and $\mathcal{S}$, respectively.

For $\mathcal{R}$, the expected utility of following a recommendation of action i is $\boldsymbol{x}^T \boldsymbol{\varrho} \geq \varrho_E$ by construction of the mechanism. By simply taking action 1 deterministically, $\mathcal{R}$ gets an expected utility of ϱ_E and a signal $\sigma = 1$ with probability $1/n$. Hence, $(\boldsymbol{x} + (n-1) \cdot \boldsymbol{y})^T \boldsymbol{\varrho} = n \cdot \varrho_E$. Consequently, the expected utility of $\mathcal{R}$ for taking an action $i' \neq i$ when getting a signal $\sigma = i$ is $\boldsymbol{y}^T \boldsymbol{\varrho} \leq \varrho_E$. Thus, the mechanism is persuasive.

Now, consider any persuasive mechanism φ used by $\mathcal{S}$. Let x_t denote the probability that (over random order and randomization in φ) a signal for type t is sent by $\mathcal{S}$. We use $\boldsymbol{x} = (x_1, \ldots, x_n)$ to denote the vector. Persuasiveness of φ implies the following constraints:

1. $\boldsymbol{x}^T \boldsymbol{\varrho} \geq \varrho_E$, since this expected utility can be achieved by $\mathcal{R}$ simply by taking action 1 deterministically, and

2. $\|\boldsymbol{x}\|_1 = 1$, since the mechanism can be considered direct without loss of generality and will therefore send a single signal for exactly one action.

Hence, the distribution $\boldsymbol{x}$ resulting from the optimal persuasive mechanism is a feasible solution to the following maximization problem for the expected sender utility

$$
\begin{array}{lrcll}
\text{Max.} & \boldsymbol{x}^T \boldsymbol{\xi} & & & \\
\text{s.t.} & \boldsymbol{x}^T \boldsymbol{\varrho} & \geq & \varrho_E & \\
& \|\boldsymbol{x}\|_1 & = & 1 & \\
& x_t & \geq & 0 & \text{for all } t = 1, \ldots, n
\end{array}
\tag{3.1}
$$

Algorithm 1: Basic Pareto Mechanism

Input: A set of type value pairs $(\varrho_t, \xi_t)_{t\in[n]}$ and the state of nature $\boldsymbol{\theta}$, revealing permutation π from types to actions

1 Set $\varrho_E = {}^1\!/_n \cdot \sum_{t=1}^{n} \varrho_t$ and set $c_{\mathcal{S}} = \arg\max_{t' \in \arg\max_{t\in[n]} \xi_t} \varrho_{t'}$.

2 Let $\mathcal{C} = \{(\varrho_t, \xi_t) \mid t \in [n]\}$ and $conv(\mathcal{C})$ be the convex hull of $\mathcal{C}$.

3 Let $PC(\mathcal{C})$ be the Pareto frontier of $conv(\mathcal{C})$.

4 **if** $\varrho_{c_{\mathcal{S}}} \geq \varrho_E$ **then** Set $a = b = c_{\mathcal{S}}$.

5 **else**

6   Find types $a, b \in [n]$ s.t. (θ_a, θ_b) is the segment of $PC(\mathcal{C})$ that intersects with line $\varrho = \varrho_E$.

    `// See Fig. 1 and 2 for an illustration;` $a = b$ `possible`

`// Determine probability for type` a`:`

7 **if** $\varrho_a = \varrho_b$ **then** Set $\alpha = 1$.

8 **else if** $\varrho_a \neq \varrho_b$ *and* $\xi_a = \xi_b$ **then** Set $\alpha = 0$.

9 **else** Set $\alpha = \frac{\varrho_E - \varrho_b}{\varrho_a - \varrho_b}$.

10 Draw $x \sim \text{Unif}[0, 1]$.

11 **if** $x \leq \alpha$ **then** Set $c = a$.

12 **else** Set $c = b$.

13 Send signal $\pi(c)$.

The distribution $\boldsymbol{x}$ computed by the Basic Pareto Mechanism clearly represents an optimum solution to the above linear program. Hence, the mechanism is an optimal persuasive mechanism.

To show that the Basic Pareto Mechanism requires a polynomial running time, consider its individual steps. The first line requires a time linear in n to determine ϱ_E and $c_{\mathcal{S}}$. For lines 3-6, observe the following: If the input is sorted by ascending values of ϱ and secondly by descending values of ξ, the Pareto frontier can be computed in time linear in n. Hence, the running time is upper bounded by $\mathcal{O}(n \log n)$ for sorting the n entries. Finding the types a and b is then a matter of traversing the Pareto frontier with at most n nodes and determining the probability α. Finally, we assume that drawing $x \sim \text{Unif}[0, 1]$ can be done in constant time. In total, this gives us a running time of $\mathcal{O}(n \log n)$ for the Basic Pareto Mechanism. □

Clearly, the optimality of the Basic Pareto Mechanism hinges on the fact that both $\mathcal{S}$ and $\mathcal{R}$ a priori know the types of the actions, only the exact order is unknown. In the following, we want to discuss more general cases of this basic offline persuasion setting.

Dughmi and Xu [36] gave the following formulation of the Bayesian persuasion problem as a linear program.

$$\begin{aligned}
\max \sum_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} q_{\boldsymbol{\theta}} \cdot \varphi(\boldsymbol{\theta}, i) \cdot \xi(\theta_i) & & \\
\text{s.t.} \sum_{i=1}^{n} \varphi(\boldsymbol{\theta}, i) = 1 & & \text{for } \boldsymbol{\theta} \in \Theta \\
\sum_{\boldsymbol{\theta}\in\Theta} q_{\boldsymbol{\theta}} \cdot \varphi(\boldsymbol{\theta}, i) \cdot \varrho(\theta_i) \geq \sum_{\boldsymbol{\theta}\in\Theta} q_{\boldsymbol{\theta}} \cdot \varphi(\boldsymbol{\theta}, i) \cdot \varrho(\theta_j) & & \text{for } i, j \in [n]
\end{aligned}$$

$$\varphi(\boldsymbol{\theta}, i) \geq 0 \qquad \text{for } \boldsymbol{\theta} \in \Theta, i \in [n]$$

If all states of nature and the distribution over them are given explicitly, this linear program has a size linear in terms of the input size. In many interesting scenarios, this is not the case. Consider an example with n independent distributions with m types for each action. Even if these distributions individually are given explicitly, this means that there are m^n possible states of nature in total. This results in the linear program having an exponential size compared to the input, making this linear program infeasible for solving the general Bayesian persuasion problem. If the action types are distributed independently and identically, the resulting symmetry can be leveraged to reduce the linear program to a polynomially sized one. This leads to the following result.

Theorem 3.2 (Dughmi, Xu [36, Theorem 3.4])
Consider the Bayesian persuasion problem with n independently and identically distributed actions and m types, with parameters $\boldsymbol{q} \in \Delta_m$, $\boldsymbol{\xi} \in \mathbb{R}^m$ and $\boldsymbol{\varrho} \in \mathbb{R}^m$ given explicitly. An optimal and persuasive signaling scheme can be implemented in time polynomial in m and n.

Additionally, in the same paper, they showed that computing the optimal expected utility for $\mathcal{S}$ is $\#\mathcal{P}$-hard if action types are independently – but not identically – distributed.

Theorem 3.3 (Dughmi, Xu [36, Theorem 4.1])
Consider the Bayesian persuasion problem with independent actions, with action-specific payoff distributions given explicitly. It is $\#\mathcal{P}$-hard to compute the optimal expected sender utility.

These results hold when the signal space is unconstrained. This means that there are at least n different signals, one for each of the n actions. For constrained signal spaces, Dughmi et al. [34] gave the following hardness result.

Theorem 3.4 (Dughmi, Kempe, Qiang [34, Theorem 1.5])
For any constant $\gamma > 0$, it is $\mathcal{NP}$-hard to construct a signaling scheme approximating the maximum expected sender utility to within a factor γ, given an explicit representation of a Bayesian persuasion game and a bound k on the number of signals.

They show this hardness of approximation via a reduction from the gap version of Independent Set, i.e., given a graph G with n vertices and a constant $\varepsilon > 0$, deciding whether the largest independent set in G has size less than n^{ε} or more than $n^{1-\varepsilon}$ [49].

We study the offline Bayesian persuasion problem further and consider instances which permit computing (near-)optimal signaling schemes in polynomial time – even when the communication complexity is limited and there are only $k < n$ signals available to $\mathcal{S}$, contrasting the result of Theorem 3.4.

3.1 Symmetric Instances

In this section, we discuss *symmetric instances*, i.e., instances for which the following holds. For every state of nature $\boldsymbol{\theta}$, each permutation $\pi(\boldsymbol{\theta})$ has the same probability, i.e., $q_{\boldsymbol{\theta}} = q_{\pi(\boldsymbol{\theta})}$. It is straightforward to see that the IID scenario or the random order example at the beginning of this chapter satisfy this symmetry condition. Hence, for some types of symmetric instances and with an unlimited signal space, we have already seen a solution to this problem – namely in Theorem 3.2 as well as the Basic Pareto Mechanism. In the following, we discuss some other classes of symmetric instances which have a succinct representation but allow for an optimal scheme in polynomial time. In particular, we study the *prophet-secretary* as well as the *d-random-order* scenario. The name of the former stems from the literature on online stopping theory. There are n independent distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$ which are known a priori, but their order is drawn uniformly at random. In the latter scenario, there exist $d \geq 1$ vectors with n types each that are explicitly given. Additionally, a distribution $\mathcal{D}$ over these d vectors is known. The state of nature is then drawn as follows. First, one of the vectors is drawn according to distribution $\mathcal{D}$. Then, the elements of the drawn vector are permuted uniformly at random. One can easily see that both of these scenarios satisfy our definition of symmetric instances.

In Section 3.1.1, we discuss a characterization of optimal schemes for symmetric instances. In Section 3.1.2, we give our algorithm for finding an optimal scheme. Finally, in Section 3.1.3, we show that our algorithm runs in time polynomial in the input size for the classes of symmetric instances mentioned above. From the start, we assume that $|\Sigma| = k \leq n$.

3.1.1 Characterization of Optimal Schemes

In this section, we discuss our approach of finding an optimal scheme for symmetric instances. Due to Lemma 2.1, we restrict our attention to direct and persuasive schemes which recommend a set of k distinct actions. Note that this includes the case of $k = n$, i.e., the case of unconstrained communication. Using the following lemma, we can further refine the search space to schemes that recommend actions from the set $[k]$.

Lemma 3.5
In symmetric instances, there is an optimal direct and persuasive scheme in which $\mathcal{S}$ recommends only the actions from $[k]$.

Proof. From Lemma 2.1, we know that it is without loss to only consider direct and persuasive schemes that recommend k distinct actions. In a symmetric instance, we now show that it is without loss to only consider the first k actions $1, \dots, k$. Consider an optimal direct and persuasive scheme φ. As it is direct, this scheme only recommends actions from some set $K \subseteq [n]$ with $|K| = k$. We can relabel the actions using a permutation π such that the permutation of the set K under π is $[k]$. Using the same permutation inside the scheme φ, we get a new scheme φ' that only recommends actions from $[k]$. Since all permutations of states of nature have the same probability in symmetric instances, using φ' yields the same conditional expectations for $\mathcal{R}$. This implies that, when $\mathcal{S}$ signals a recommendation for some $i \in [k]$ using φ', it is – in expectation – at least as good for $\mathcal{R}$ to take action i as it is to take a different action $i' \neq i$. Hence, φ' is a direct and persuasive scheme which only recommends actions from the set $[k]$. Additionally, the sender's expectation is the same as in the original scheme φ, i.e., $u_{\mathcal{S}}(\varphi') = u_{\mathcal{S}}(\varphi)$. □

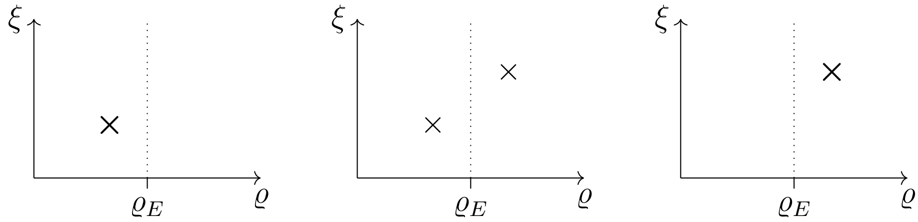


Figure 3: Simple IID-Instance and the geometric interpretation of its possible states of nature. There are 2 actions. The type of each action is drawn uniformly and independently from the set $\{1, 2\}$, where type $i \in \{1, 2\}$ provides a utility of i to both $\mathcal{R}$ and $\mathcal{S}$. The bold marker denotes both actions having the same type.

For every given realization $\boldsymbol{\theta}$ of the state of nature, we encounter a similar situation as before, when discussing the introductory example and introducing the Basic Pareto Mechanism. Again, we can interpret the types as points in the two-dimensional plane. In the introductory example above, the only difference between different states of nature was in the permutation of the types – the geometric interpretation on the other hand remained the same for all states of nature. The geometric interpretation only depends on the utility values of the types and not their ordering. In an enriched instance with more diverse type spaces, this is not the case, different states of nature will result in different geometric interpretations. Additionally, while $\mathcal{R}$ will obviously still want to achieve an expectation of at least ϱ_E, there exist instances with states of nature in which all drawn types have a value below ϱ_E for $\mathcal{R}$. Hence, a point-wise guarantee of an expected utility of at least ϱ_E for every state of nature is impossible. As an illustration, consider Figure 3 which shows the geometric interpretation for a very simple IID-instance. Yet, as it turns out, the complete state of nature is not required and – similar to the Basic Pareto Mechanism in the introductory example – we will be able to utilize the graphical interpretation of the types of the first k actions.

The following lemma shows that for a symmetric instance with k signals it suffices to only consider the types of the first k actions.

Lemma 3.6
In a symmetric instance with n actions and k signals, there is an optimal direct and persuasive scheme which only considers the action types of the first k actions.

Proof. Consider an optimal scheme φ which considers all n actions but only recommends actions from the set $[k]$. We can assume that φ exists due to Lemma 3.5. Now, consider a scheme ψ which only has access to the first k actions. ψ utilizes φ in the following way. First, ψ draws the types of the remaining actions $k+1, \dots, n$ according to the known distribution. Then, ψ runs φ with all n action types and forwards φ's signal to $\mathcal{R}$. This way, ψ is persuasive and achieves the same expected utility using only the first k action types as φ does using the complete state of nature with n actions. Hence, there exists an optimal direct and persuasive scheme that only requires the first k action types to give a recommendation. $\square$

Using Lemma 3.6, we concentrate on the actions from $[k]$ and generalize the notation we used for the Basic Pareto Mechanism. We use $\mathcal{C}$ to denote the realized set of action types of the first k actions. We call such a set $\mathcal{C}$ a *k-type set*. Analogous to the probability for a state of nature, we denote the probability that a k-type set $\mathcal{C}$ is

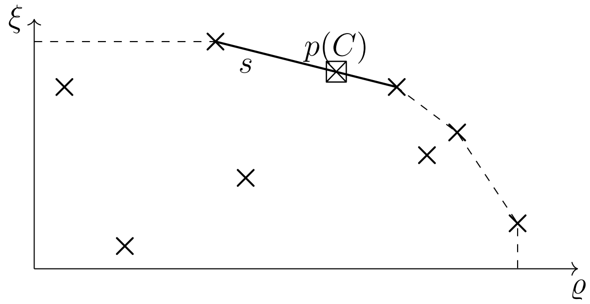


Figure 4: A k-type set $\mathcal{C}$ for which the point $p(\mathcal{C})$ corresponds to slope s on the Pareto frontier of $\mathcal{C}$ (dashed line).

realized by $q_\mathcal{C} = \Pr\left[\bigcup_{i\in[k]}\{\theta_i\} = \mathcal{C}\right]$. In the following, the schemes will only take the first k actions as input.

For a k-type set $\mathcal{C}$ and a persuasive and direct scheme φ, consider the point $(\mathbb{E}\left[u_\mathcal{R}(\varphi) \mid \mathcal{C}\right], \mathbb{E}\left[u_\mathcal{S}(\varphi) \mid \mathcal{C}\right])$ composed of the expected utilities for $\mathcal{R}$ and $\mathcal{S}$ conditioned on the k-type set being $\mathcal{C}$. We call this point the *recommendation point for $\mathcal{C}$* of φ.

More generally, we define the notion of a *point collection*. For each possible k-type set $\mathcal{C}$, a point collection $\mathcal{P}$ contains a point $p(\mathcal{C}) = (p_\mathcal{R}(\mathcal{C}), p_\mathcal{S}(\mathcal{C}))$ inside the convex hull of $\mathcal{C}$. For a point collection $\mathcal{P}$, we define the utilities of $\mathcal{R}$ and $\mathcal{S}$ by

$$u_\mathcal{R}(\mathcal{P}) = \sum_\mathcal{C} q_\mathcal{C} \cdot p_\mathcal{R}(\mathcal{C}) \qquad \text{and} \qquad u_\mathcal{S}(\mathcal{P}) = \sum_\mathcal{C} q_\mathcal{C} \cdot p_\mathcal{S}(\mathcal{C}) \ ,$$

respectively.

Consider a direct and persuasive scheme φ. The recommendation points for each k-type set $\mathcal{C}$ of φ form a point collection. Observe that the utility for both $\mathcal{S}$ and $\mathcal{R}$ of the point collection equals the respective utility of the scheme. However, it is easy to imagine a point collection which does not correspond to a persuasive signaling scheme so the converse does not hold.

Since we are interested in finding an optimal persuasive scheme using point collections, we consider point collections with the following properties. Firstly, the overall utility for $\mathcal{R}$ should be at least ϱ_E, since this is a necessary condition for a scheme to be persuasive. Secondly, the point for each typeset $\mathcal{C}$ should be on the Pareto frontier. Otherwise, the utility for $\mathcal{S}$ could easily be improved – while keeping the utility for $\mathcal{R}$ the same. We know from the description of the Pareto frontier at the beginning of this chapter that for every slope $s \in [0, -\infty]$, there is a point on the Pareto frontier such that a line with slope s lies tangent to the Pareto frontier at this point. We say that a point *corresponds to a slope s* if a line with slope s lies tangent to the Pareto frontier in the point. We extend this definition to action types θ. A type θ *corresponds to a slope s* if a line with slope s lies tangent to the Pareto frontier in the point $(\varrho(\theta), \xi(\theta))$. In Figure 4, we depict a k-type set $\mathcal{C}$ for which the point $p(\mathcal{C})$ corresponds to a slope s on the Pareto frontier of $\mathcal{C}$. It turns out that a common slope for all points of a point collection will be useful to identify the optimal persuasive scheme.

We formalize the two properties above using the following definition for point collections.

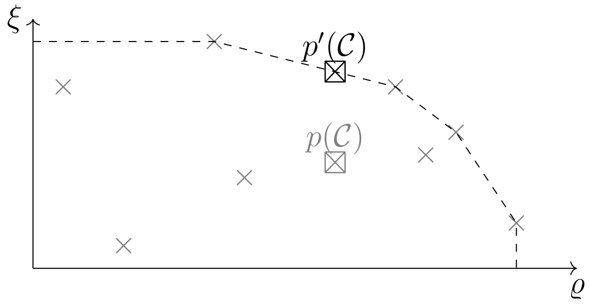


Figure 5: A k-type set $\mathcal{C}$ and a direct and persuasive scheme φ with recommendation point $p(\mathcal{C}) = (\mathbb{E}[u_{\mathcal{R}}(\varphi) \mid \mathcal{C}], \mathbb{E}[u_{\mathcal{S}}(\varphi) \mid \mathcal{C}])$. The expected utility for $\mathcal{S}$ can be improved by moving $p(\mathcal{C})$ vertically upwards to $p'(\mathcal{C})$ on the Pareto frontier of $\mathcal{C}$.

Definition 3.7
*For $s \leq 0$, a point collection $\mathcal{P}$ is s-*Pareto *if the following two conditions hold.*

1. *$u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E$.*
2. *For every k-type set $\mathcal{C}$, $p(\mathcal{C})$ is on the Pareto frontier of $\mathcal{C}$ and corresponds to slope s.*

Our first main result is a characterization of an optimal scheme via an s-Pareto point collection.

Theorem 3.8
For every symmetric instance, there is an optimal scheme whose recommendation points constitute a sender-optimal s-Pareto point collection, over all $s \leq 0$.

We prove the theorem using the following three lemmas. First, we show that for every persuasive scheme φ, there is an s-Pareto point collection $\mathcal{P}$ with $u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi)$.

Lemma 3.9
For every direct and persuasive scheme φ, there is an s-Pareto point collection $\mathcal{P}$ with $u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi)$.

Proof. Let $\mathcal{P}(\varphi)$ be the point collection of recommendation points of some direct and persuasive scheme φ. This means that $\mathcal{P}(\varphi)$ satisfies the first condition of being s-Pareto, namely that the utility for $\mathcal{R}$ is at least ϱ_E.

To show the second condition, we adjust the recommendation points of $\mathcal{P}(\varphi)$ using two steps that retain the utility for $\mathcal{R}$ but can only increase the utility for $\mathcal{S}$.

First, we can increase the utility for $\mathcal{S}$ by moving recommendation points vertically upwards to the Pareto frontier. This does not change the utility for $\mathcal{R}$ and thereby does not affect the first condition, i.e., $u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E$, but can only increase the utility for $\mathcal{S}$. We denote the adjusted point collection by $\mathcal{P}'$. After this step, we have $u_{\mathcal{S}}(\mathcal{P}') \geq u_{\mathcal{S}}(\varphi)$. Figure 5 outlines this improvement for a k-type set $\mathcal{C}$.

If the second condition for Definition 3.7 is not met, there must exist two k-type sets $\mathcal{C}_1 \neq \mathcal{C}_2$ such that there exists no common slope s that $p(\mathcal{C}_1)$ and $p(\mathcal{C}_2)$ correspond to. Without loss of generality, we assume that $q_{\mathcal{C}_1} > 0$ and $q_{\mathcal{C}_2} > 0$, otherwise, (at least) one of the states of nature does not occur and there is nothing to do. For convenience, we use the short notation $p_1 = p(\mathcal{C}_1) = (\varrho_1, \xi_1)$ and $p_2 = p(\mathcal{C}_2) = (\varrho_2, \xi_2)$. For $i = 1, 2$, the point p_i corresponds to an interval of slopes $[s_i^{(r)}, s_i^{(l)}]$, where $s_i^{(r)}$ is the slope of the line segment to the right (i.e., with increasing value ϱ) of p_i and $s_i^{(l)}$ is the slope of the line segment to the left of the slope. If p_i is not on an endpoint of a line segment, $s_i^{(r)} = s_i^{(l)}$. If there is no common slope, we assume without loss of generality that $s_2^{(l)} < s_1^{(r)}$. This means that all line segments adjacent to p_2 are steeper than the ones adjacent to p_1 – recall that all slopes are non-positive. In the following, we will use $s_1 = s_1^{(r)}$ and $s_2 = s_2^{(l)}$.

We construct a new point collection $\mathcal{P}_1$ from $\mathcal{P}$ which has an increased utility for $\mathcal{S}$ by adjusting only p_1 and p_2 such that they correspond to a common slope. For all other k-type sets $\mathcal{C} \neq \mathcal{C}_1, \mathcal{C}_2$, we keep the point $p(\mathcal{C})$. We update p_1 and p_2 to

$$p_1' = (\varrho_1 + \delta \cdot q_{\mathcal{C}_2}, \xi_1 + \delta \cdot q_{\mathcal{C}_2} \cdot s_1) \qquad \text{and} \qquad p_2' = (\varrho_2 - \delta \cdot q_{\mathcal{C}_1}, \xi_2 - \delta \cdot q_{\mathcal{C}_1} \cdot s_2)$$

using a sufficiently small $\delta > 0$. Since $s_1 > s_2$ and s_1 is the slope of the line segment to the right of p_1 and s_2 the slope of the line segment to the left of p_2, such a δ necessarily exists and it is possible to move p_1 and p_2 along their respective Pareto frontiers on the line segments with slopes s_1 and s_2, respectively. Graphically speaking, we move p_1 to the right (i.e., we increase ϱ) by $\delta \cdot q_{\mathcal{C}_2}$ and move p_2 to the left (i.e., we decrease ϱ) by $\delta \cdot q_{\mathcal{C}_1}$.

Hence, the receiver utility does not change as

$$\begin{aligned} u_{\mathcal{R}}(\mathcal{P}_1) &= \sum_{\mathcal{C} \neq \mathcal{C}_1, \mathcal{C}_2} q_{\mathcal{C}} \cdot p_{\mathcal{R}}(\mathcal{C}) + q_{\mathcal{C}_1} \cdot (\varrho_1 + \delta \cdot q_{\mathcal{C}_2}) + q_{\mathcal{C}_2} \cdot (\varrho_2 - \delta \cdot q_{\mathcal{C}_1}) \\ &= u_{\mathcal{R}}(\mathcal{P}) + q_{\mathcal{C}_1} \cdot \delta \cdot q_{\mathcal{C}_2} - q_{\mathcal{C}_2} \cdot \delta \cdot q_{\mathcal{C}_1} \quad = \quad u_{\mathcal{R}}(\mathcal{P}) \ . \end{aligned}$$

Since $\mathcal{P}$ satisfied $u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E$, so does $\mathcal{P}_1$ and the first condition of being s-Pareto is fulfilled. Additionally, the utility for $\mathcal{S}$ grows to

$$\begin{aligned} u_{\mathcal{S}}(\mathcal{P}_1) &= \sum_{\mathcal{C} \neq \mathcal{C}_1, \mathcal{C}_2} q_{\mathcal{C}} \cdot p_{\mathcal{S}}(\mathcal{C}) + q_{\mathcal{C}_1} \cdot (\xi_1 + \delta \cdot q_{\mathcal{C}_2} \cdot s_1) + q_{\mathcal{C}_2} \cdot (\xi_2 - \delta \cdot q_{\mathcal{C}_1} \cdot s_2) \\ &= u_{\mathcal{S}}(\mathcal{P}) + \underbrace{q_{\mathcal{C}_1} \cdot \delta \cdot q_{\mathcal{C}_2}}_{>0} \cdot \underbrace{(s_1 - s_2)}_{>0} \quad > \quad u_{\mathcal{S}}(\mathcal{P}) \quad \geq \quad u_{\mathcal{S}}(\varphi) \ , \end{aligned}$$

since $0 \geq s_1 > s_2$. Overall, $\mathcal{P}_1$ continues to satisfy the first condition of being s-Pareto and improves the utility for the sender.

The value δ is chosen such that p_1' and p_2' both stay on the line segments with slopes s_1 and s_2, respectively. Repeated application of this modification yields point collections $\mathcal{P}_2, \mathcal{P}_3, \ldots$ until finally points p_1 and p_2 correspond to at least one common slope. Whenever an endpoint of a line segment is reached, if this endpoint does not correspond to a slope of the other point, the process can be continued. Moreover, we can apply this modification repeatedly as long as there are two k-type sets $\mathcal{C}_1 \neq \mathcal{C}_2$ with points that have no common slope. Eventually, we reach an s-Pareto point collection $\mathcal{P}$ with $u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi)$. Figure 6 outlines this procedure. □

Consider any s-Pareto point collection $\mathcal{P}$. Note that this implies that the point $p(\mathcal{C})$ for each k-type set $\mathcal{C}$ is on the Pareto frontier of $\mathcal{C}$. We define a direct scheme φ^* as follows: For any k-type set $\mathcal{C}$ and its corresponding point $p(\mathcal{C})$, φ^* recommends one of the (at most) two actions whose types compose the corresponding line segment of $p(\mathcal{C})$ on the Pareto frontier. The probabilities for each of the potential actions are such that

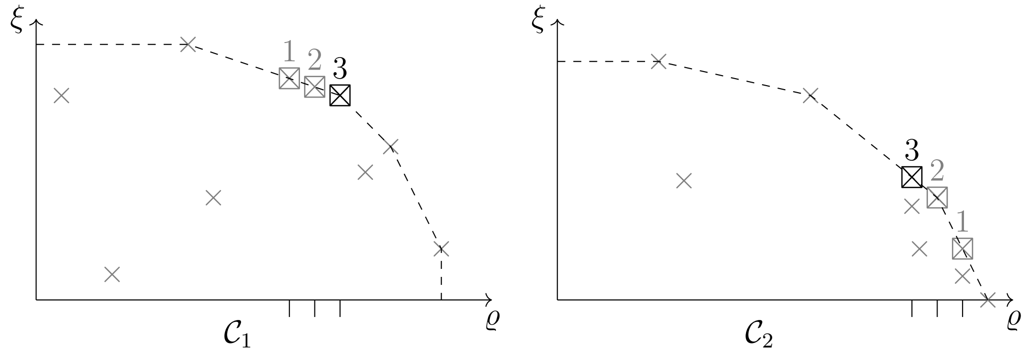


Figure 6: The Pareto frontiers of two different k-type sets $\mathcal{C}_1$ and $\mathcal{C}_2$ with $q_{\mathcal{C}_1} = q_{\mathcal{C}_2}$ and combined iterative improvement of the overall expected sender utility, such that the points labeled with 3 in both sides correspond to a common slope. The points with the same label correspond to a state in the improvement procedure. Whereas the absolute difference δ in ϱ for both $\mathcal{C}_1$ and $\mathcal{C}_2$ is the same in every step, the overall change in ξ is positive.

$p(\mathcal{C})$ corresponds to the expected utilities for $\mathcal{S}$ and $\mathcal{R}$, conditioned on the k-type set being $\mathcal{C}$. This directly implies that $u_{\mathcal{S}}(\varphi^*) = u_{\mathcal{S}}(\mathcal{P})$ and $u_{\mathcal{R}}(\varphi^*) = u_{\mathcal{R}}(\mathcal{P})$. Crucially, the types are chosen independently of the actual action to which they are assigned within the first k actions. We can see that this scheme has a symmetry property for symmetric instances, i.e., the probability that a recommended action has a certain type is independent of the number of the action.

We formalize this insight for general schemes.

A *symmetric scheme* φ (see also [36]) is direct and recommends with each signal a distinct action in $[k]$. The conditional distribution over types (resulting from the prior and φ) is the same for each recommended action, i.e., for every $i, i' \in [k]$ and every $\theta \in \Theta_i = \Theta_{i'}$

$$\Pr[\theta_i = \theta \mid \sigma = i] = \Pr[\theta_{i'} = \theta \mid \sigma = i'] \ .$$

The conditional distribution over types is the same for each non-recommended action in $[k]$ and the same for each non-recommended action in $[n] \setminus [k]$, no matter which (other) action is recommended, i.e., for every $\ell, \ell' \in [k]$ and $i, i' \in [k] \setminus \{\ell, \ell'\}$, and for every $\ell, \ell' \in [k]$ and $i, i' \in [n] \setminus [k]$, we have

$$\Pr[\theta_i = \theta \mid \sigma = \ell] = \Pr[\theta_{i'} = \theta \mid \sigma = \ell']$$

for every $\theta \in \Theta_i = \Theta_{i'}$. Thus, a symmetric scheme gives rise to three distributions over types: a distribution $\mathcal{D}_{yes}$ for any recommended action, a distribution $\mathcal{D}_{no}$ for any non-recommended action in $[k]$, and a distribution $\mathcal{D}_{never}$ for any non-recommended action in $[n] \setminus [k]$. For a symmetric scheme φ in a symmetric instance, we show that $u_{\mathcal{R}}(\varphi) \geq \varrho_E$ is not only necessary but also sufficient for persuasiveness of φ.

Lemma 3.10
In symmetric instances, a symmetric scheme φ is persuasive if and only if $u_{\mathcal{R}}(\varphi) \geq \varrho_E$.

Proof. We have already seen that a scheme φ which guarantees strictly less utility than ϱ_E to $\mathcal{R}$ cannot be persuasive.

Consider a symmetric scheme and the three resulting type distributions $\mathcal{D}_{yes}$, $\mathcal{D}_{no}$ and $\mathcal{D}_{never}$. We denote by ϱ_{yes}, ϱ_{no} and ϱ_{never} the expectations of the utility of $\mathcal{R}$ for the respective distributions. If φ is persuasive, then $\varrho_{yes} = u_{\mathcal{R}}(\varphi) \geq \varrho_E$. Now, for the reverse direction, assume that $\varrho_{yes} \geq \varrho_E$. Clearly, since instance and scheme are symmetric, it holds that $\varrho_{never} = \varrho_E$. Again, due to symmetry, every action $i \in [k]$ gets recommended with probability $1/k$. Hence, $\frac{1}{k} \cdot \varrho_{yes} + \frac{k-1}{k} \cdot \varrho_{no} = \varrho_E$, and $\varrho_{yes} \geq \varrho_E$ implies $\varrho_{no} \leq \varrho_E$. It is not profitable for $\mathcal{R}$ to deviate from the recommended action. Hence, if $\varrho_{yes} \geq \varrho_E$, then φ is persuasive. □

The symmetric scheme φ^* based on an s-Pareto point collection satisfies the constraint in Lemma 3.10 by definition. As such, we obtain the following result, which finishes the proof of Theorem 3.8.

Lemma 3.11
For every s-Pareto point collection $\mathcal{P}$, there is a symmetric, direct, and persuasive signaling scheme φ^* with $u_{\mathcal{S}}(\varphi^*) = u_{\mathcal{S}}(\mathcal{P})$.

Having completed the proof of Theorem 3.8, we know that there exists an optimal scheme which corresponds to an s-Pareto point collection. As we discussed earlier, there is a point corresponding to each $s \in (-\infty, 0]$ on the Pareto frontier of any k-type set. Hence, we first have to identify a value s which corresponds to an optimal s-Pareto point collection. We show how we are able to achieve this in the following section.

3.1.2 Efficient Computation of Optimal Schemes

In this section, we describe our Slope-Algorithm (Algorithm 2) for computing an optimal s-Pareto point collection. For a succinct description, we first use an assumption under which the algorithm performs efficiently – in the next section, we describe how this assumption can be satisfied for some subclasses of symmetric instances.

Our algorithm first restricts the set of possible values for s by identifying a polynomially sized set S of meaningful candidates for slope-values s to enumerate. For every pair of types a, b the algorithm determines the probability (denoted by p_{ab}) that their line segment (denoted by $\overline{ab}$) is contained in the Pareto frontier of the k-type set $\mathcal{C}$. For every pair with $s > 0$, one type Pareto dominates the other and the pair can be discarded. Similarly, if $p_{ab} = 0$, the pair can be discarded. The critical step in the first part of the algorithm is the computation of p_{ab} in line 4. For now, we assume that the algorithm has oracle access to these quantities via a *probability oracle*. We will discuss in Section 3.1.3 how to implement the probability oracle for some classes of symmetric instances in polynomial time.

At the end of the first for-loop, the algorithm has collected in $S = \{s_1, \ldots, s_\ell\}$ all meaningful slopes of non-empty segments that can appear on the Pareto frontier of the possible k-type sets. In addition to the slopes in S, every Pareto frontier can be assumed to contain all slopes from $(-\infty, 0]$. An optimal scheme might not necessarily correspond to a slope $s \in S$. If it does not, it must correspond to some slope r with $s_i < r < s_{i+1}$ for some $i \in [\ell - 1]$. Note that all slopes $r \in (s_i, s_{i+1})$ correspond to the same point on the Pareto frontier. Hence, r_i for $i \in [\ell + 1]$ in line 8 can be chosen arbitrarily.

Note that in some degenerate cases, the set S could be empty – i.e., if there is only a single type on the Pareto frontier for every k-type set $\mathcal{C}$. Then, we add an auxiliary

Algorithm 2: Slope-Algorithm

Input: Symmetric instance with set $\Theta = \Theta_1 = \ldots = \Theta_n$ of action types and distribution q

1 Set $S = \emptyset$ and $L = \emptyset$.
2 **for** *every pair of types* $a, b \in \Theta$, $a \neq b$ **do**
3 Let s be the slope of $\overline{ab}$ and set $p_{ab} = 0$.
4 **if** $s \leq 0$ **then** Determine the probability p_{ab} that $\overline{ab}$ is on the Pareto frontier of k-type sets.
5 **if** $p_{ab} > 0$ **then** Update $S = S \cup \{s\}$.
6 **if** $S \neq \emptyset$ **then**
7 Sort the slopes of S such that $s_1 < s_2 < \ldots < s_\ell$.
8 Pick auxiliary slopes $r_1, \ldots, r_{\ell+1}$ such that $r_1 < s_1 < r_2 < s_2 < \ldots < s_\ell < r_{\ell+1}$.
9 **else**
10 Set $r_1 = -1$.
11 Update $S = S \cup \{r_1, \ldots, r_{\ell+1}\}$.
12 **for** *every slope* $s \in S$ **do**
13 **for** *every type* $c \in \Theta$ **do**
14 Determine the probability $p_c^{(s)}$ that c is the unique point corresponding to s on the Pareto frontier of k-type sets.
15 Solve the following lin. program to determine an s-Pareto point collection:

$$\begin{aligned}
\text{Max.} \quad & \sum_{\substack{c,d \in \Theta, c \neq d \\ \overline{cd} \text{ has slope } s}} p_{cd} \cdot \left(\alpha_{cd}^{(s)} \cdot \xi_c + \left(1 - \alpha_{cd}^{(s)}\right) \cdot \xi_d\right) + \sum_{c \in \Theta} p_c^{(s)} \cdot \xi_c \\
\text{s.t.} \quad & \sum_{\substack{c,d \in \Theta, c \neq d \\ \overline{cd} \text{ has slope } s}} p_{cd} \cdot \left(\alpha_{cd}^{(s)} \cdot \varrho_c + \left(1 - \alpha_{cd}^{(s)}\right) \cdot \varrho_d\right) + \sum_{c \in \Theta} p_c^{(s)} \cdot \varrho_c \geq \varrho_E \\
& \alpha_{cd}^{(s)} \in [0, 1] \text{ for all } c, d \in \Theta
\end{aligned} \tag{3.2}$$

if *Linear program* (3.2) *has a feasible optimal solution* $\boldsymbol{\alpha}^{(s)}$ **then**
16 Update $L = L \cup \{(\boldsymbol{\alpha}^{(s)}, s)\}$.
17 **return** *Best point collection in* L *with corresponding slope*

slope of $r_1 = -1$ so that the for-loop in line 12 does not fail. Again, the actual value of r_1 does not matter. The single type corresponds to all slope-values $s \in (-\infty, 0]$.

Now even if a slope s is attained by some segment $\overline{ab}$ in a k-type set $\mathcal{C}$, it might be that for some other k-type set $\mathcal{C}'$, slope s only corresponds to a single point on the Pareto frontier of $\mathcal{C}'$. As such, the algorithm also determines in line 14 for every $s \in S$ the probability that a single type $c \in \Theta$ corresponds to s on the Pareto frontier of $\mathcal{C}$. This is the critical step in the second part of the algorithm. Again, we assume that the algorithm has oracle access to these quantities via a probability oracle. We defer the discussion on how to implement the probability oracles in polynomial time to Section 3.1.3.

Finally, after having computed all probabilities the algorithm solves the linear program given in (3.2). For the linear program, we assume that s is the common slope of the point collection. For all k-type sets $\mathcal{C}$ where a single point c corresponds to slope s, obviously the point c is chosen. For all other k-type sets $\mathcal{C}$, in which some line segment $\overline{cd}$ with slope s is on the Pareto frontier, there is a choice to pick a point from that segment. Using a convex combination of the points c and d, the point from that segment can be represented by the variable $\alpha_{cd}^{(s)} \in [0,1]$. The program optimizes point locations to maximize the expected utility for $\mathcal{S}$ (in the objective function) and to guarantee at least the average utility of ϱ_E for $\mathcal{R}$. For a given slope s, (3.2) might be infeasible. However, by enumerating all relevant common slopes, the algorithm sees at least one feasible solution. It returns the best feasible solution for the linear program along with the slope s^*.

Note that the output of the algorithm is sufficient for $\mathcal{S}$ to implement an optimal persuasive scheme. $\mathcal{S}$ looks at the realized k-type set $\mathcal{C}$, computes the Pareto frontier, and looks for slope s^*. If s^* is realized by a segment $\overline{ab}$, $\mathcal{S}$ recommends the action with type a with probability $\alpha_{ab}^{(s^*)}$ and the action with type b with probability $1-\alpha_{ab}^{(s^*)}$. If it is realized through a single type c, $\mathcal{S}$ recommends the corresponding action with probability 1.

Proposition 3.12

Given an efficient algorithm to compute the probability oracle, the Slope-Algorithm computes an optimal direct and persuasive scheme for symmetric instances in polynomial time.

Proof. Correctness follows from the characterization in the last section and the observations above. We denote the maximal running time of the probability oracle by T_o and the maximal time needed to solve the linear program (3.2) by T_{LP}. Let $m = |\Theta|$ denote the finite number of types. Then, finding the slopes can be done in time $O(m^2 \cdot T_o)$. Sorting the slopes requires time $O(m^2 \cdot \log m)$ as $O(m^2)$ slopes are identified in the first for-loop. For the second for-loop, we iterate through the $O(m^2)$ slopes, including the auxiliary ones. For each slope, we need at most m calls to the probability oracle and solve a polynomially-sized linear program. Overall, the running time is $O(m^3 \cdot T_o + m^2 \cdot T_{LP} + m^2 \cdot \log m)$. □

Note that the Basic Pareto Mechanism discussed above essentially is a special case of the Slope-Algorithm for symmetric instances with only a single k-type set and $k = n$. There exists only a single k-type set $\mathcal{C}$ so the set S will contain only the realized slopes on the Pareto frontier after the first for-loop. The auxiliary slopes can only correspond to individual types. Hence, checking the best possible value for $\mathcal{S}$ while ensuring that $u_{\mathcal{R}} \geq \varrho_E$ is equivalent to finding the intersection of the line $\varrho = \varrho_E$ and the Pareto frontier.

Using geometric properties of the utility pairs in prophet-secretary and d-random-order scenarios, we show how to design polynomial-time probability oracles in these scenarios. We prove the following result in the next subsection.

Theorem 3.13

An optimal signaling scheme with k signals can be computed in polynomial time for the prophet-secretary and the d-random-order scenarios.

3.1.3 Efficient Probability Oracles

We subdivide the proof of Theorem 3.13 in the two different classes, namely prophet-secretary and d-random-order.

Prophet-Secretary

We start with the prophet-secretary scenario in which there are n probability distributions $1, \dots, n$ over type spaces $\Theta^1, \dots, \Theta^n$. For distribution i, we denote the probability that type $\theta \in \Theta^i$ is drawn by q^i_θ. Recall that in the classic prophet-secretary scenario, a single type is drawn independently from each distribution and the drawn types are permuted uniformly at random. Note that it does not make a difference whether we first draw the types from their respective distributions and then permute the drawn types or first permute the distributions and then draw a type from each of them. Hence, we use the latter order of operations, namely we first apply a uniform random permutation on the distributions and then draw a single type independently from each of them.

For a simpler exposition, we use two assumptions. First, we assume that the n types spaces are mutually disjoint. This is without loss of generality as different types can have the same utility values for $\mathcal{S}$ and $\mathcal{R}$. Secondly, we assume that types are in general position, i.e., there are no more than two distinct types on any given straight line. We discuss in the end of the section how to deal with scenarios which do not satisfy this second assumption.

As input, there are n type spaces $\Theta^1, \dots, \Theta^n$ together with the probabilities q^i_θ for all $i \in [n]$ and $\theta \in \Theta^i$. Hence, the representation of the input has a size at least linear in n, $\max_{i\in[n]} |\Theta^i|$, and $\max_{i\in[n],\theta\in\Theta^i}(-\log q^i_\theta)$. In Algorithm 2, we use the following two queries Q1 and Q2.

Q1 For a pair of types a and b, return the probability p_{ab} that $\overline{ab}$ is in the Pareto-frontier of a k-type set $\mathcal{C}$. This query is performed in line 4.

Q2 For a type c and a slope s, return the probability $p^{(s)}_c$ that c is the unique point that corresponds to slope s on the Pareto-frontier of a k-type set $\mathcal{C}$. This query is performed in line 14.

We show how these two queries can be implemented in polynomial time for the prophet-secretary scenario.

Query Q1

If a and b are from the same distribution i, we obviously have $p_{ab} = 0$, as they cannot occur in the same k-type set or even state of nature. Otherwise, denote by i_a and i_b the distributions of a and b, respectively. Now, consider all types $c \in \Theta^i$ for every remaining distribution $i \neq i_a, i_b$. If a type c which lies below the line through a and b is in the same k-type set $\mathcal{C}$, c cannot prevent that a and b form a line segment on the Pareto frontier of $\mathcal{C}$. Hence, we call c an *allowed type*. If that is not the case, i.e., c lies above the line through a and b, the line segment between a and b cannot be on the Pareto frontier of $\mathcal{C}$. Hence, in this case, c cannot be an allowed type. Figure 7 illustrates the notion of allowed types. To capture the probability of drawing an allowed type from distribution $i \neq i_a, i_b$, we denote by Θ^i_{ab} the set of all allowed

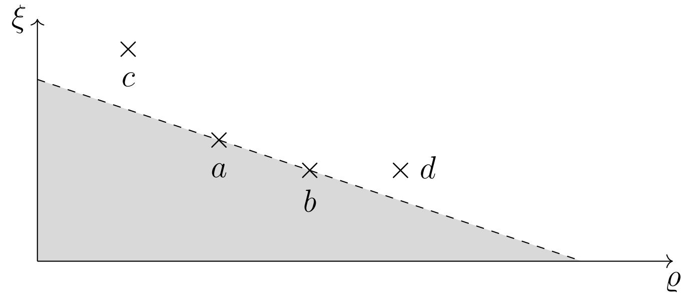


Figure 7: In order for a and b to be on the Pareto frontier of a k-type set, allowed types can only be in the gray area. If c is also in the k-type set, a could never be on the Pareto frontier. Similarly, if d is in the k-type set with a and b, b could not be on the Pareto frontier.

types of distribution i. Hence, the probability that an allowed type for types a and b is drawn from distribution $i \neq i_a, i_b$ is $q^i_{ab} = \sum_{c \in \Theta^i_{ab}} q^i_c$. For each pair of types a, b, the probabilities q^i_{ab} can be determined in time linear in the total number of types.

The following three conditions have to be satisfied for $\overline{ab}$ to be on the Pareto frontier of some k-type space.

1. Distributions i_a and i_b have to be permuted to the first k actions. This happens with probability $\frac{k}{n} \cdot \frac{k-1}{n-1}$.
2. Types a and b have to be drawn from their distributions, respectively. The types are drawn with probability $q^{i_a}_a$ and $q^{i_b}_b$, respectively. Hence, the combined probability is $q^{i_a}_a \cdot q^{i_b}_b$ as these events are independent.
3. An allowed type must be drawn from every other distribution $i \neq i_a, i_b$ which is permuted to the first k actions. For this condition, we consider every $(k-2)$-sized subset $A \subseteq [n] \setminus \{i_a, i_b\}$ and compute the probability that only allowed types are drawn from every distribution in A. Since the types from different distributions are drawn independently, this probability is

$$\frac{1}{\binom{n-2}{k-2}} \cdot \sum_{\substack{A \subseteq [n] \setminus \{i_a, i_b\} \\ |A| = k-2}} \prod_{i \in A} q^i_{ab} \ .$$

The probabilities for the first two conditions can obviously be computed in time polynomial in the input size. For the third probability, we need to compute the sum of products of all subsets of size $k-2$ of $n-2$ numbers. The following lemma shows how this can done in time $O(nk)$ using a dynamic program.

Lemma 3.14
Given a set $A = \{a_1, \dots, a_n\}$ of n numbers, the sum of products of all subsets of size k can be computed in time $O(nk)$.

Proof. We use a dynamic programming approach. Note that all subsets of size k can be iterated in the following way: For index $j = k$ to n, take the subsets of size $k-1$ of $\{a_1, \dots, a_{j-1}\}$ and add a_j to these sets. This way, no subset is skipped or iterated multiple times. Similarly, by multiplying a_j with the sum of products of subsets of size

$k-1$ of $\{a_1, \dots, a_{j-1}\}$, we can compute the sum of products of subsets of size k which have a_j as element with the highest index.

We use a table T of size $k \times n$. In $T[i][j]$, we store the sum of products of subsets of size i whose largest index is j. Clearly, for all $j < i$, we have $T[i][j] = 0$ since the set $\{a_1, \dots, a_j\}$ cannot have subsets of size $i > j$. The first row ($i = 1$) can easily be filled out using $T[1][j] = a_j$ for $j = 1, \dots, n$. There is only a single subset of size 1 which includes as element with largest index element a_j. The subsequent rows can then be filled successively using $T[i][j] = a_j \cdot \sum_{\ell=1}^{j-1} T[i-1][\ell]$ for all $j \geq i$. Using an additional variable to hold the current value of this summation means that this operation only takes constant time. In total, to fill out each entry of T, a constant number of steps is required. Hence, completely filling out T takes time $O(nk)$. Finding the overall sum is the matter of summing up $\sum_{j=1}^{n} T[k][j]$. $\square$

Overall, the probability that $\overline{ab}$ is on the Pareto frontier of some k-type set is

$$p_{ab} = \frac{k}{n} \cdot \frac{k-1}{n-1} \cdot q_a^{i_a} \cdot q_b^{i_b} \cdot \frac{1}{\binom{n-2}{k-2}} \cdot \sum_{\substack{A \subseteq [n] \setminus \{i_a, i_b\} \\ |A| = k-2}} \prod_{i \in A} q_{ab}^i$$

and we have shown how to compute this quantity in time polynomial in the input size for the prophet-secretary scenario.

Query Q2

For the second query, we consider a single type c which is the unique point corresponding to a slope s in a k-type set. We denote by i_c and Θ^{i_c} the corresponding distribution and type space, respectively. Obviously, no other type $c' \in \Theta^{i_c}$ can be drawn into the same k-type set. For every other distribution $i \neq i_c$, we again consider every type $d \in \Theta^i$. Similar to Q1, d is an allowed type if it lies below the line through c with slope s. We denote by Θ_c^i the set of allowed types from distribution i and by $q_c^i = \sum_{d \in \Theta_c^i} q_d^i$ the probability to draw an allowed type from that distribution, analogous to the notation for Q1. Clearly, the set Θ_c^i as well as the probability q_c^i can be determined in time linear in the number of types.

Now, for c to be the unique point corresponding to a slope s on the Pareto frontier of some k-type space, the following three conditions have to be satisfied.

1. Distribution i_c has to be permuted to the first k actions. This occurs with a probability of k/n.
2. Type c has to be drawn from distribution i_c. This event has probability q_c^i.
3. For every other distribution $i \neq i_c$ permuted to the first k actions, an allowed type must be drawn from that distribution. Again, we consider every $(k-1)$-sized subset $A \subseteq [n] \setminus \{i_c\}$ and compute the probability that only allowed types are drawn from every distribution in A. Since the types from different distributions are drawn independently, this probability is

$$\frac{1}{\binom{n-1}{k-1}} \cdot \sum_{\substack{A \subseteq \{1,\dots,n\} \setminus \{i_c\} \\ |A| = k-1}} \prod_{i \in A} q_c^i \ .$$

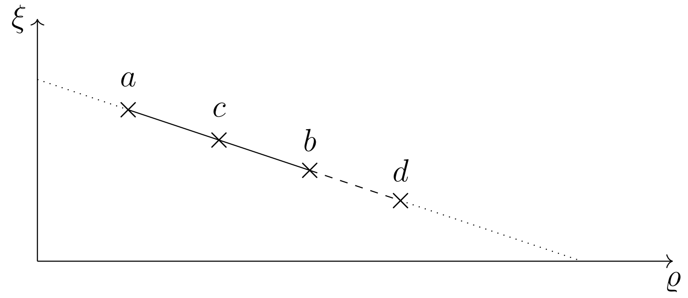


Figure 8: Illustration of allowed types when types are not in general position. For segment $\overline{ab}$ to be on the Pareto frontier of some k-type set $\mathcal{C}$, type c is allowed. If type d on the other hand is also in $\mathcal{C}$, this would make $\overline{ad}$ the longest segment on the Pareto frontier with the corresponding slope in $\mathcal{C}$, hence, type d is not an allowed type.

The computations for the first two conditions can clearly be completed in polynomial time with respect to the input size. Additionally, using Lemma 3.14, we can compute the probability for the third condition in time polynomial in the input size. Overall, the probability that type c is the unique point corresponding to a slope s on the Pareto frontier of some k-type set is

$$p_c^{(s)} = \frac{k}{n} \cdot q_c^{i_c} \cdot \frac{1}{\binom{n-1}{k-1}} \cdot \sum_{\substack{A \subseteq \{1,\ldots,n\} \setminus \{i_c\} \\ |A| = k-1}} \prod_{i \in A} q_c^i$$

and we have shown that this quantity can be computed in time polynomial in the input size for the prophet-secretary scenario.

On general position

Throughout this section, we have assumed that types are in general position, i.e., there are only two types on any straight line. Here, we describe in short how to handle cases where this assumption is not satisfied. If more than two types are on a single line, the events for them forming a line segment on the Pareto frontier with slope s are not disjoint. We can remedy this using the following observation.

A segment is not counted multiple times if we only consider the longest possible line segment on the Pareto frontier for any given slope. Hence, the following modification has to be made for query Q1 when determining the set of allowed types Θ^i_{ab}: All types of Θ^i that are on the segment $\overline{ab}$ are allowed since this does not prohibit $\overline{ab}$ to be the longest possible line segment. All types of Θ^i that are on the straight line going through a and b but not on $\overline{ab}$ must not be allowed. With this modification to compute the probabilities for p_{ab}, general position of types is no longer required. An illustration of this can be found in Figure 8.

This concludes the discussion of the prophet-secretary scenario. The following proposition summarizes the result of this section.

Proposition 3.15
For the prophet-secretary scenario we can implement a probability oracle for the Slope-Algorithm in polynomial time.

Together with Proposition 3.12, we obtain the main result of this section.

Corollary 3.16
An optimal signaling scheme for the prophet-secretary scenario with k signals can be computed in polynomial time.

d-Random-Order

The input for d-random-order instances consists of d type vectors $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^d$ of size n as well as a distribution $\mathcal{D}$ over these vectors. For $i \in [d]$, the probability that vector $\boldsymbol{\theta}^i$ is drawn is denoted by $q_{\boldsymbol{\theta}^i}$. The state of nature is generated as follows. First, a vector $\boldsymbol{\theta}$ is drawn according to distribution $\mathcal{D}$ and then a uniform random permutation is applied to the elements of $\boldsymbol{\theta}$. Hence, the input size of a d-random-order instance is at least linear in $d \cdot n$ and $\max_{i \in [n]}(-\log q_{\boldsymbol{\theta}^i})$. Again, we will have to show how to implement queries Q1 and Q2 in time polynomial in these quantities to show that a polynomial-time probability oracle exists. We use the same assumptions as in the prophet-secretary scenario, i.e., we assume that all $d \cdot n$ types are distinct and that types are in general position. As before, the first assumption is without loss of generality since different types are allowed to have the same utility for $\mathcal{S}$ and $\mathcal{R}$. The second assumption is not without loss but allows for a simpler exposition. We will discuss at the end of the section how to deal with instances that do not satisfy the second assumption of types in general position.

Query Q1

Recall that Q1 is the query for the probability p_{ab} that $\overline{ab}$ is on the Pareto frontier of a k-type set. Obviously, types a and b have to come from the same vector $\boldsymbol{\theta}^i$ to be in the same k-type set. Otherwise, the probability is $p_{ab} = 0$. The same holds for the other types in the k-type set – all of them have to come from $\boldsymbol{\theta}^i$. Hence, we consider each type c from $\boldsymbol{\theta}^i$ with $c \neq a, b$. We call c an *allowed type* if c lies below the line going through the types a and b. Otherwise, $\overline{ab}$ could not be on the Pareto frontier of a k-type set which includes a, b, and c. Figure 7 illustrates this. We denote the set of allowed types for types a and b from vector $\boldsymbol{\theta}^i$ by A^i_{ab}. To compute this set, only the types in the vector $\boldsymbol{\theta}^i$ have to be considered. Checking whether some type c is allowed for types a and b can be done in constant time. Hence, the set can be computed in time linear in n.

We again identify 3 conditions for $\overline{ab}$ to be on the Pareto frontier.

1. Vector $\boldsymbol{\theta}^i$ has to be drawn from $\mathcal{D}$. This event has probability $q_{\boldsymbol{\theta}^i}$.

2. Types a and b have to be permuted to the first k actions, which has a probability of $\frac{k}{n} \cdot \frac{k-1}{n-1}$.

3. Only allowed types can be permuted to the remaining $k-2$ slots of the k-type set. The probability for this is $\binom{|A^i_{ab}|}{k-2} \Big/ \binom{n-2}{k-2}$, where we assume that $\binom{|A^i_{ab}|}{k-2} = 0$ if $|A^i_{ab}| < k-2$.

All these computations can be performed in constant time once A^i_{ab} is determined – which can be done in time linear in n. The overall probability that $\overline{ab}$ is on the Pareto frontier of a k-type set is

$$p_{ab} = q_{\boldsymbol{\theta}^i} \cdot \frac{k}{n} \cdot \frac{k-1}{n-1} \cdot \frac{1}{\binom{n-2}{k-2}} \cdot \binom{|A^i_{ab}|}{k-2} \ .$$

This can clearly be computed in time polynomial in the size of the input representation of a d-random-order instance.

Query Q2

For query Q2, the oracle has to determine the probability that a type c is the unique point on the Pareto frontier of a k-type set which corresponds to a slope s. Hence, let c be a type from some vector $\boldsymbol{\theta}^i$. Again, we use the notion of *allowed types* where a type d from $\boldsymbol{\theta}^i$ is allowed if it lies below the line through c with slope s. We denote by A^i_c the set of allowed types from vector $\boldsymbol{\theta}^i$ for type c. Since only the single vector $\boldsymbol{\theta}^i$ has to be considered, A^i_c can be computed in time linear in n.

The 3 conditions for c to be the unique point corresponding to a slope s on the Pareto frontier are the following.

1. Vector $\boldsymbol{\theta}^i$ has to be drawn from $\mathcal{D}$. This event has probability $q_{\boldsymbol{\theta}^i}$.
2. Type c has to be permuted to the first k actions, which happens with probability k/n.
3. Only allowed types can be permuted to the remaining $k-1$ slots of the k-type set. The probability for this is $\binom{|A^i_c|}{k-1} \Big/ \binom{n-1}{k-1}$, where we assume that $\binom{|A^i_c|}{k-1} = 0$ if $|A^i_c| < k-1$.

Again, all quantities can be computed in time polynomial in the representation size of the input. Overall, the probability that a type c is the unique point corresponding to a slope s on the Pareto frontier of a k-type set is

$$p^{(s)}_c = q_{\boldsymbol{\theta}^i} \cdot \frac{k}{n} \cdot \frac{1}{\binom{n-1}{k-1}} \cdot \binom{|A^i_c|}{k-1} \ .$$

On General Position

We can use the same procedure as in the previous section on prophet-secretary instances to remove the assumption of general position of types, namely, to only consider the longest line segment with a particular slope s. Hence, we are able to state the following proposition summarizing the result of this section.

Proposition 3.17
For the d-random-order scenario we can implement a probability oracle for the Slope-Algorithm in polynomial time.

Together with Proposition 3.12, we obtain the main result of this section.

Corollary 3.18
An optimal signaling scheme for the d-random-order scenario with k signals can be computed in polynomial time.

Distribution	$\mathcal{D}_1$		$\mathcal{D}_2$		$\mathcal{D}_3$		
Type	θ_{11}	θ_{12}	θ_{21}	θ_{22}	θ_{31}	θ_{32}	θ_{33}
Value-Pair (ϱ, ξ)	(6,5)	(4,9)	(5,1)	(5,6)	(7,7)	(4,0)	(2,5)
Probability	0.3	0.7	0.8	0.2	0.4	0.3	0.3

Table 2: Example of a ϱ_E-optimal instance. Action 2 has a deterministic utility of 5 for $\mathcal{R}$, which is the highest a priori expected utility for $\mathcal{R}$ among the actions.

This concludes the section on symmetric instances. We have shown for two settings, the prophet-secretary as well as the d-random-order scenarios, how to implement an optimal signaling scheme in polynomial time. The algorithm we used offers a framework for other symmetric scenarios which allow for efficient probability oracles.

3.2 Independent Instances

In this section, we focus on actions with independent distributions, i.e., each action $i \in [n]$ has a fixed distribution $\mathcal{D}_i$ over a type space Θ_i with corresponding probabilities $\boldsymbol{q}_i$ from which its type is drawn. We again study a general scenario with $k \leq n$ signals. Even for the less general case of n signals for n actions, Dughmi and Xu [36] (cf. Theorem 3.3) showed that computing the optimal expected sender utility is $\#\mathcal{P}$-hard. Another hardness result for general Bayesian persuasion problems with a limited number of signals was given by Dughmi, Kempe and Qiang [34] (cf. Theorem 3.4). Contrasting these negative results, we study a subclass of independent instances which satisfy the following condition we call *ϱ_E-optimality*. We describe a persuasive scheme which extracts a constant-factor approximation for k signals in polynomial time for ϱ_E-optimal instances. An instance is ϱ_E-optimal if there exists an optimal signaling scheme such that each $\sigma \in \Sigma$ guarantees $\mathcal{R}$ an expected utility of at least ϱ_E conditional on the signal being σ. Recall that ϱ_E denotes the highest a priori expected utility for $\mathcal{R}$ of any of the n actions.

As an example class of ϱ_E-optimal instances, consider the following. There is an action i which has a deterministic value of ϱ_E for $\mathcal{R}$ among all types. Note that this does not mean that only a deterministic type exists for action i, the value for the sender could very well be randomized. This means that $\mathcal{R}$ can always guarantee a value of ϱ_E by choosing action i regardless of the signal. Hence, the scheme needs to ensure a conditional expectation of at least ϱ_E for $\mathcal{R}$ to be persuasive. A short example for an ϱ_E-optimal instance is given in Table 2 below.

For a different example, consider symmetric instances. In the previous section, we have seen that there always exists an optimal symmetric scheme for symmetric instances. Such a scheme ensures that every signal provides an expected utility for $\mathcal{R}$ of at least ϱ_E. Hence, all symmetric instances satisfy ϱ_E-optimality.

Our scheme proceeds as follows. First, an action with the highest a priori expectation for $\mathcal{R}$ is identified which essentially serves as a fallback or an outside option for $\mathcal{R}$. If there are several such actions, choose the one which has the highest expected utility for $\mathcal{S}$ among those with the highest expectation for $\mathcal{R}$. Without loss of generality, we relabel the actions such that this is action n. Then, the scheme identifies a set S of $k-1$ other actions and computes the recommendation probabilities for the actions from $S \cup \{n\}$. These steps are done separately and we discuss them in more detail below.

1. Choose a set S of $k-1$, and
2. compute the recommendation probabilities for the actions from $S \cup \{n\}$.

We give two variants for the first step. We first use a classic greedy approach to find a suitable set S in Section 3.2.1 in our *Independent Scheme* φ_{IS}. The approximation guarantee of φ_{IS} is given in the following Theorem 3.19.

Theorem 3.19
The Independent Scheme φ_{IS} is a direct and persuasive scheme for ϱ_E-optimal independent instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$,

$$u_{\mathcal{S}}(\varphi_{IS}) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot \left(1 - \left(1 - \frac{1}{k}\right)^{k-1}\right) \cdot u_{\mathcal{S}}(\varphi^*) \ .$$

This scheme already achieves a constant approximation ratio of at least $3/8 = 0.375$ for $k = 2$. For $k \to \infty$, the approximation ratio improves to $(1 - 1/e)^2 \approx 0.3996$. Still, we can improve this guarantee, especially for large values of k. In Section 3.2.2, we discuss the *Improved Independent Scheme* φ_{IIS} which utilizes an FPTAS to identify the set S. The approximation ratio for φ_{IIS} is given in Theorem 3.20.

Theorem 3.20
The Improved Independent Scheme φ_{IIS} is a direct and persuasive scheme for ϱ_E-optimal independent instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$ and every constant $\varepsilon > 0$

$$u_{\mathcal{S}}(\varphi_{IIS}) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot (1 - \varepsilon) \cdot \left(1 - \frac{1}{k}\right) \cdot u_{\mathcal{S}}(\varphi^*) \ .$$

For $k = 2$, φ_{IS} and φ_{IIS} provide (up to the factor $(1 - \varepsilon)$) the same approximation guarantees. For $k \to \infty$, φ_{IIS} achieves an approximation ratio of $(1 - 1/e) \cdot (1 - \varepsilon)$ compared to the ratio of $(1 - 1/e)^2$ of φ_{IS}. Additionally, we show that the approximation guarantee for this approach is asymptotically tight. Hence, a further improvement of the approximation ratio requires different techniques.

Finally, in Section 3.2.3, we discuss general independent instances and the limitations of our scheme concerning instances that do not satisfy ϱ_E-optimality.

3.2.1 Constant-Factor Approximation

In this section, we describe the Independent Scheme φ_{IS} and prove Theorem 3.19. Let $m = \max_{i \in [n]} |\Theta_i|$. For ease of exposition, we assume that $m = |\Theta_i|$ for all $i \in [n]$. This assumption is without loss of generality. For each $i \in [n]$ with $|\Theta_i| < m$, we can add a sufficient number of dummy types θ_i with $q_{i\theta_i} = 0$ to Θ_i to ensure $|\Theta_i| = m$. Hence, we are able to use $[m]$ to enumerate the possible types of each action i. To shorten notation, we use ϱ_{ij} and ξ_{ij} to denote the utility for $\mathcal{R}$ and $\mathcal{S}$ of action i with type j.

To identify a good subset $S \subseteq [n-1]$ for the first step of the scheme, we first consider the following parameterized linear programs for each individual action.

$$\begin{aligned} g_i(z) = \text{Max.} \quad & \sum_{j=1}^{m} x_{ij} \cdot \xi_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} \ \leq \ z \\ & \sum_{j=1}^{m} x_{ij} \cdot \varrho_{ij} \ \geq \ \varrho_E \cdot \sum_{j=1}^{m} x_{ij} \\ & x_{ij} \ \in \ [0, q_{ij}] \qquad \forall j \in [m] \end{aligned} \tag{3.3}$$

These linear programs have the following interpretation: For each action $i \in [n]$, the optimization problem (3.3) maximizes the expected utility for $\mathcal{S}$ conditional on the overall probability mass of (at most) z on action i. For each $j \in [m]$, the variable x_{ij} represents the probability mass for type j of action i. Hence, the first constraint limits the overall probability mass on action i to at most z. The second constraint ensures that the conditional expectation for $\mathcal{R}$ based on a signal for action i is at least ϱ_E. Clearly, the probability mass x_{ij} should be non-negative and at most the probability q_{ij} that type j is drawn. This is enforced by the last constraint.

We define the following set function $f : 2^{[n-1]} \to \mathbb{R}$ using these linear programs (3.3) for $i \in S \cup \{n\}$, limiting the probability mass on action i to z_i and $\sum_{i \in S\cup\{n\}} z_i$ to at most 1.

$$f(S) = \max \left\{ \sum_{i \in S\cup\{n\}} g_i(z_i) \;\middle|\; \sum_{i \in S\cup\{n\}} z_i \leq 1, \; z_i \geq 0 \;\; \forall i \in S \cup \{n\} \right\} \tag{3.4}$$

Consider a direct and persuasive scheme $\varphi_{S\cup\{n\}}$ for ϱ_E-optimal instances that recommends only actions from the set $S \cup \{n\}$. We denote the ex-post recommendation probabilities of $\varphi_{S\cup\{n\}}$ for type j from action i by by x_{ij}. Clearly, setting $z_i = \sum_{j=1}^{m} x_{ij}$ for all $i \in S\cup\{n\}$ means that all constraints in (3.3) and (3.4) are satisfied for the probabilities x_{ij} stemming from $\varphi_{S\cup\{n\}}$. Hence, the ex-post probabilities x^*_{ij} of an optimal scheme $\varphi^*_{S\cup\{n\}}$ give rise to a feasible solution for every linear program (3.3). Further, setting $z_i = \sum_{j=1}^{m} x^*_{ij}$ is feasible for (3.4). Hence, for every subset $S \subseteq [n-1]$, $f(S)$ is an upper bound on the optimal sender utility of a scheme recommending the actions from the set $S \cup \{n\}$, i.e.,

$$f(S) \geq u_{\mathcal{S}}\left(\varphi^*_{S\cup\{n\}}\right) \ . \tag{3.5}$$

Using the set function f, we are now able to choose a good set $S \subseteq [n-1]$ of $k-1$ actions to complete the first step of our Independent Scheme φ_{IS} using the algorithm *ActionsGreedy* (Algorithm 3). It starts out with an empty set $S = \emptyset$ and then greedily adds actions to S, maximizing the utility gain in every individual step, until $|S| = k-1$.

Having collected $k-1$ actions in the set S, our scheme φ_{IS} then uses *ComputeSignal* (Algorithm 4) to compute the signal based on the realized types of the actions in $S\cup\{n\}$.

We begin the analysis of the scheme's performance by observing that the procedure ActionsGreedy employs the classic greedy algorithm for maximization of a submodular function. The following lemma shows that indeed, f is a submodular function.

Lemma 3.21
The function f is non-negative, non-decreasing, and submodular.

Algorithm 3: ActionsGreedy

Input: Distributions $\mathcal{D}_1 = (\Theta_1, \boldsymbol{q}_1), \mathcal{D}_2 = (\Theta_2, \boldsymbol{q}_2), \ldots, \mathcal{D}_n = (\Theta_n, \boldsymbol{q}_n)$, s.t. $\sum_j q_{n,j} \cdot \varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j} \cdot \xi_{nj} = \max_{i \in [n]\,:\, \sum_j q_{i,j} \cdot \varrho_{ij} = \varrho_E} \sum_j q_{i,j} \cdot \xi_{ij}$, parameter $2 \leq k \leq n$

Set $S = \emptyset$.
for $\ell = 1, \ldots, k-1$ **do**
  Let i be an action maximizing $f(S \cup \{i\}) - f(S)$.
  Set $S = S \cup \{i\}$.
return S

Algorithm 4: ComputeSignal

Input: Distributions $\mathcal{D}_1 = (\Theta_1, \boldsymbol{q}_1), \mathcal{D}_2 = (\Theta_2, \boldsymbol{q}_2), \ldots, \mathcal{D}_n = (\Theta_n, \boldsymbol{q}_n)$, s.t. $\sum_j q_{n,j} \cdot \varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j} \cdot \xi_{nj} = \max_{i \in [n]\,:\, \sum_j q_{i,j} \cdot \varrho_{ij} = \varrho_E} \sum_j q_{i,j} \cdot \xi_{ij}$, parameter $2 \leq k \leq n$, set $S \subseteq [n-1]$ with $|S| = k-1$

for $i \in S \cup \{n\}$, let z_i^* and $\boldsymbol{x}_i^*$ be the values of the optimal solution in $f(S)$.
Order actions in $S \cup \{n\}$ s.t. $\frac{g_{i_1}\left(z_{i_1}^*\right)}{z_{i_1}^*} \geq \ldots \geq \frac{g_{i_k}\left(z_{i_k}^*\right)}{z_{i_k}^*}$, where we assume $\frac{0}{0} = 0$.
for $\ell = 1, \ldots, k$ **do**
  Observe type j of action i_ℓ.
  Draw $x \sim$ Unif[0,1].
  if $x \leq \frac{x_{i_\ell,j}^*}{q_{i_\ell,j}}$ **then return** *signal for action* i_ℓ
return *signal for action* n

Before we give the proof of Lemma 3.21, we give a short intuition on the functions g_i for $i \in [n]$. We illustrate g_i as defined in (3.3) in Figure 9. For each $i \in [n]$, g_i is a non-negative, piece-wise linear and concave function. As such, the slopes of the line segments of g_i have a decreasing non-negative value and an increase in z_i only yields diminishing returns for g_i. Hence, we employ a waterfilling approach to maximize $f(S)$, i.e., when increasing the total mass $\sum_{i \in S \cup \{n\}} z_i$, we maintain a common slope s of all functions g_i at their respective points z_i for all $i \in S \cup \{n\}$. Note that, similar to the discussion of slopes on the Pareto frontier in the previous section, a breakpoint between different line segments of g_i corresponds to all intermediate slopes. Hence, z_i will not be increased evenly among all $i \in S \cup \{n\}$. Rather, to guarantee consistent results without using some arbitrary tie-breaking rule, our approach is to evenly distribute excess mass among all $i \in S \cup \{n\}$ for which z_i can be increased while keeping the same common slope.

Proof of Lemma 3.21. We have already seen that every g_i is non-negative. Hence, f can only be non-negative. Further, f is clearly non-decreasing. For a subset $S \subsetneq [n-1]$ and $j \notin S$, consider the optimal allocation $z_i^{(S)}$ for $f(S)$. Note that $z_j = 0$ and $z_i = z_i^{(S)}$ for all $i \in S \cup \{n\}$ is a feasible allocation for $f(S \cup \{j\})$ and $f(S \cup \{j\}) \geq f(S)$ follows directly.

Without loss of generality, an optimal assignment of z_i for $i \in S \cup \{n\}$ distributes

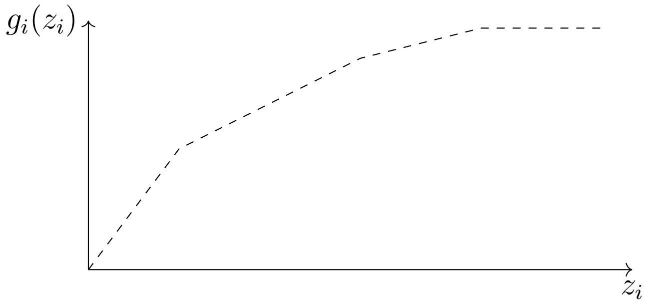


Figure 9: Schematic of a function g_i (cf. (3.3)) used for submodular function f in (3.4).

a unit of mass to the functions g_i. This can be reached using the waterfilling approach we described above, i.e., always keeping a common slope among all g_i for $i \in S \cup \{n\}$, increase z_i until $\sum_{i \in S \cup \{n\}} z_i = 1$. The slopes of the functions g_i are decreasing for increasing values of z_i. Hence, when going from S to $S \cup \{j\}$, the value of the common slope cannot decrease and the z_i have to be non-increasing.

To show that f is indeed submodular, consider $S \subseteq T \subsetneq [n-1]$ and $j \notin T$, $j \in [n-1]$. Let $z_j^{(S)}$ and $z_j^{(T)}$ denote the optimal choices in $f(S \cup \{j\})$ and $f(T \cup \{j\})$, respectively. Since z_i are non-increasing, we have $z_j^{(S)} \geq z_j^{(T)}$. We use the following auxiliary function f' to bound the marginal increase in f, where f' has the additional constraint that $z_j \leq z_j^{(T)}$, i.e.,

$$f'(Q) = \max \left\{ \sum_{i \in Q \cup \{n\}} g_i(z_i) \;\middle|\; \sum_{i \in Q \cup \{n\}} z_i \leq 1,\ z_i \geq 0 \ \ \forall i \in Q \cup \{n\},\ \text{and } z_j \leq z_j^{(T)} \right\} .$$

This means that $f'(S \cup \{j\}) \leq f(S \cup \{j\})$, since a non-negative mass of $z_j^{(S)} - z_j^{(T)}$ needs to be redistributed from j to a subset of $S \cup \{n\}$ which cannot have higher slopes.

Now, when comparing $f(S)$ to $f'(S \cup \{j\})$ and $f(T)$ to $f(T \cup \{j\})$, in both cases a mass of $z_j^{(T)}$ is assigned to g_j, leading to contribution of $g_j\left(z_j^{(T)}\right)$ from j to both $f'(S \cup \{j\})$ and $f(T \cup \{j\})$. Since the overall mass is 1, this reassignment of mass $z_j^{(T)}$ to g_j means that the same mass $z_j^{(T)}$ is removed from the remaining functions. We denote by $\tilde{f}(Q)$ another auxiliary function which only distributes a total mass of $1 - z_j^{(T)}$ to $Q \cup \{n\}$, i.e.,

$$\tilde{f}(Q) = \max \left\{ \sum_{i \in Q \cup \{n\}} g_i(z_i) \;\middle|\; \sum_{i \in Q \cup \{n\}} z_i \leq 1 - z_j^{(T)},\ z_i \geq 0 \ \ \forall i \in Q \cup \{n\} \right\} .$$

Thus, we have $g_j\left(z_j^{(T)}\right) = f(T \cup \{j\}) - \tilde{f}(T)$ and $g_j\left(z_j^{(T)}\right) = f'(S \cup \{j\}) - \tilde{f}(S)$. Additionally, we have $f(S) - \tilde{f}(S) \leq f(T) - \tilde{f}(T)$. This holds because the final mass of $z_j^{(T)}$ contributes at most as much value to $f(S)$ as it does to $f(T)$ since the common slope value for the set S cannot be higher than the one for the set T. Combining these (in)equalities, we get

$$f(T \cup \{j\}) - f(T) \ \leq\ f'(S \cup \{j\}) - f(S) \ \leq\ f(S \cup \{j\}) - f(S) \ ,$$

where the second inequality comes from the fact that $f'(S \cup \{j\}) \leq f(S \cup \{j\})$. This concludes the proof of submodularity for the function f. $\square_{\text{Lemma 3.21}}$

The following lemma compares the value $f(S)$ for the subset S with size $k-1$ computed by ActionsGreedy to the expected utility for the sender from the optimal scheme φ^*. Due to Lemma 2.1 we can assume that the optimal scheme φ^* directly recommends a set of actions of size k. In the following, we denote this set by K.

Lemma 3.22
For every $k \geq 2$, ActionsGreedy computes a subset S of $k-1$ actions such that

$$f(S) \geq \left(1 - \left(1 - \tfrac{1}{k}\right)^{k-1}\right) \cdot u_{\mathcal{S}}(\varphi^*) \ .$$

For the proof of Lemma 3.22, we use the following result from submodular optimization by Krause and Golovin [59] which itself is based on the analysis of submodular optimization by Nemhauser, Wolsey, and Fisher [66].

Lemma 3.23 ([59, Theorem 3.5])
Fix a nonnegative monotone submodular function $f : 2^{[n]} \to \mathbb{R}_+$ and let $\{S_i\}_{i \geq 0}$ be the greedily selected sets defined by

$$S_0 = \emptyset, \quad S_{i+1} = S_i \cup \left\{ \arg\max_{j \in [n]} f(S_i \cup \{j\}) - f(S_i) \right\} \ \textit{for } i \geq 0 \ .$$

Then for all positive integers t and ℓ,

$$f(S_\ell) \geq \left(1 - \left(1 - \frac{1}{t}\right)^{\ell}\right) \max_{S \,:\, |S| \,\leq\, t} f(S) \ .$$

Proof of Lemma 3.22. ActionsGreedy is a standard greedy algorithm for submodular maximization. It computes a set $S \subseteq [n-1]$ of $k-1$ actions. Our scheme will then recommend an action from the action set $S \cup \{n\}$. Note that, in contrast to the set computed by ActionsGreedy, the set K of actions recommended by φ^* does not have to include n. Hence, an optimal scheme $\varphi^*_{K \cup \{n\}}$ might be able to recommend $k+1$ actions and $u_{\mathcal{S}}(\varphi^*_{K \cup \{n\}}) \geq u_{\mathcal{S}}(\varphi^*)$. We denote by S^*_k an optimal set $S \subseteq [n-1]$ for function f of size k, i.e., $S^*_k \in \arg\max\{f(S) \mid S \subseteq [n-1], |S| = k\}$. Then, the following holds.

$$u_{\mathcal{S}}(\varphi^*) \ \leq \ u_{\mathcal{S}}(\varphi^*_{K \cup \{n\}}) \ \leq \ f(K) \ \leq \ f(S^*_k) \tag{3.6}$$

This allows us to overestimate the optimal value $u_{\mathcal{S}}(\varphi^*)$ by $f(S^*_k)$. Note that this does include $k+1$ actions, one of which has to be action n.

Plugging $t = k$ as well as $\ell = k-1$ into Lemma 3.23, and using our notation of $S = S_{k-1}$, we get $f(S) \geq (1-(1-1/k)^{k-1}) \cdot f(S^*_k)$, and the lemma follows using (3.6). □

Now consider the second step of φ_{IS}, i.e., the computation of a signal using ComputeSignal. The following lemma bounds from below the expected utility our scheme achieves for any set $S \cup \{n\}$ of k actions.

Lemma 3.24
For every $k \geq 2$, let $S \cup \{n\}$ be any set of k actions. Given the set $S \cup \{n\}$ of actions, ComputeSignal computes a signaling scheme φ such that

$$u_{\mathcal{S}}(\varphi) \geq \left(1 - \left(1 - \tfrac{1}{k}\right)^{k}\right) \cdot f(S) \ .$$

The algorithm decides for each action $i \in S \cup \{n\}$ independently whether to recommend i, stopping the process after the first recommendation. Hence, we can split the probability that an action is recommended into two parts. The actions are considered consecutively, ordered by descending order of utility for $\mathcal{S}$ per unit of mass. This obviously means that action i can only be the single recommended action if no other action has been recommended prior to i. Secondly, the coin flip for action i needs to turn up as "recommend". If all coins come up as "don't recommend", a recommendation for action n is given – which is an action with the a priori highest expectation for $\mathcal{R}$ and therefore serves as a "backup".

For the proof of the lemma, we use the generalized mediant inequality which is stated in the following Lemma 3.25. The proof of Lemma 3.24 follows below.

Lemma 3.25 (Generalized Mediant Inequality)
Let $\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \geq \frac{a_\ell}{b_\ell} > 0$ denote ℓ positive fractions and $w_1, \ldots, w_\ell > 0$ positive weights. Then, it holds that

$$\frac{a_1}{b_1} \geq \frac{\sum_{i=1}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{\ell} w_i \cdot b_i} \geq \frac{a_\ell}{b_\ell} \ . \tag{3.7}$$

Additionally, if $w_1 \geq w_2 \geq \cdots \geq w_\ell$,

$$\frac{\sum_{i=1}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{\ell} w_i \cdot b_i} \geq \frac{\sum_{i=1}^{\ell} a_i}{\sum_{i=1}^{\ell} b_i} \ . \tag{3.8}$$

Proof of Lemma 3.25. For every $i \in [\ell]$, we have $\frac{a_1}{b_1} \geq \frac{a_i}{b_i} \geq \frac{a_\ell}{b_\ell}$. Multiplying by $w_i \cdot b_i > 0$ gives us $\frac{a_1}{b_1} \cdot w_i \cdot b_i \geq w_i \cdot a_i \geq \frac{a_\ell}{b_\ell} \cdot w_i \cdot b_i$ for every $i \in [\ell]$. Hence,

$$\frac{a_1}{b_1} \sum_{i=1}^{\ell} w_i \cdot b_i \geq \sum_{i=1}^{\ell} w_i \cdot a_i \geq \frac{a_\ell}{b_\ell} \cdot \sum_{i=1}^{\ell} w_i \cdot b_i$$

and (3.7) follows by dividing every term by $\sum_{i=1}^{\ell} w_i \cdot b_i > 0$.

For the second statement, we repeatedly apply the result of the first inequality. In each step $j = 1, \ldots, \ell - 1$, we replace w_j by $w_{j+1} \leq w_j$ and show that

$$\frac{\sum_{i=1}^{j-1} w_j \cdot a_i + \sum_{i=j}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{j-1} w_j \cdot b_i + \sum_{i=j}^{\ell} w_i \cdot b_i} \geq \frac{\sum_{i=1}^{j} w_j \cdot a_i + \sum_{i=j+1}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{j} w_j \cdot b_i + \sum_{i=j+1}^{\ell} w_i \cdot b_i} \ .$$

Hence, after all $\ell - 1$ steps, we have

$$\frac{\sum_{i=1}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{\ell} w_i \cdot b_i} \geq \frac{\sum_{i=1}^{\ell} w_\ell \cdot a_i}{\sum_{i=1}^{\ell} w_\ell \cdot b_i} = \frac{\sum_{i=1}^{\ell} a_i}{\sum_{i=1}^{\ell} b_i} \ .$$

Consider a single step j. We assume that $w_j > w_{j+1}$, otherwise replacing w_j by w_{j+1} does not have any effect.

$$\frac{\sum_{i=1}^{j-1} w_j \cdot a_i + \sum_{i=j}^{\ell} w_i \cdot a_i}{\sum_{i=1}^{j-1} w_j \cdot b_i + \sum_{i=j}^{\ell} w_i \cdot b_i} = \frac{w_j \cdot \overbrace{\sum_{i=1}^{j-1} a_i}^{=:a} + \overbrace{\sum_{i=j}^{\ell} w_i \cdot a_i}^{=:c}}{w_j \cdot \underbrace{\sum_{i=1}^{j-1} b_i}_{=:b} + \underbrace{\sum_{i=j}^{\ell} w_i \cdot b_i}_{=:d}}$$

Using (3.7), we know that $\frac{w_j \cdot a}{w_j \cdot b} \geq \frac{a_{j-1}}{b_{j-1}} \geq \frac{a_j}{b_j} \geq \frac{c}{d}$ as $\frac{a_i}{b_i} \geq \frac{a_{i+1}}{b_{i+1}}$. It remains to show that $\frac{w_j \cdot a + c}{w_j \cdot b + d} \geq \frac{w_{j+1} \cdot a + c}{w_{j+1} \cdot b + d}$, or equivalently

$$(w_j \cdot a + c) \cdot (w_{j+1} \cdot b + d) \geq (w_{j+1} \cdot a + c) \cdot (w_j \cdot b + d) \ ,$$

to complete the proof of (3.8). It holds that

$$\begin{aligned}(w_j \cdot a + c) \cdot (w_{j+1} \cdot b + d) &= w_j \cdot w_{j+1} \cdot a \cdot b + w_j \cdot a \cdot d + w_{j+1} \cdot b \cdot c + c \cdot d \\ &\geq w_j \cdot w_{j+1} \cdot a \cdot b + w_{j+1} \cdot a \cdot d + w_j \cdot b \cdot c + c \cdot d \\ &= (w_{j+1} \cdot a + c) \cdot (w_j \cdot b + d) \ .\end{aligned}$$

The inequality holds as $\underbrace{(w_j - w_{j+1})}_{>0} \cdot a \cdot d \geq \underbrace{(w_j - w_{j+1})}_{>0} \cdot b \cdot c$ since $\frac{a}{b} \geq \frac{c}{d}$.

$\square_{\text{Lemma 3.25}}$

Proof of Lemma 3.24. Given the chosen set S of actions, we consider these actions one-by-one in non-increasing order of $\frac{g_i(z_i^*)}{z_i^*}$. We use the for-loop in line 3 to do this. Note that we assume $0/0 = 0$ when sorting the actions in line 2 of ComputeSignal. This assumption makes sense in this context as no value can be generated from action i if $z_i^* = 0$ and hence no mass is put on action i. In iteration ℓ of the loop, the type j of the action i_ℓ currently under consideration is observed. A recommendation is issued independently with a probability of $\frac{x^*_{i_\ell j}}{q_{i_\ell j}}$. Since type j is realized for action i_ℓ with probability $q_{i_\ell j}$, the combined probability for issuing a recommendation and having type j is $x_{i_\ell j}$. This means that overall, a recommendation in iteration ℓ, conditioned on reaching this iteration is sent with probability $\sum_{j=1}^m x_{i_\ell j} \leq z^*_{i_\ell}$ due to (3.3). We now want to argue that we can assume equality $\sum_{j=1}^m x_{i_\ell j} = z^*_{i_\ell}$ without loss of generality.

We have already seen that the optimal assignment $\boldsymbol{z}^*$ can be reached through a waterfilling approach, where $z^*_{i_\ell}$ are increased until $\sum_{\ell=1}^k z^*_{i_\ell} = 1$ while keeping a common slope for g_{i_ℓ}. Note that for every $z_n \in [0,1]$, we can assume without loss of generality that the first constraint in (3.3) holds with tightness without violating the second constraint as the a priori receiver expectation for action n is ϱ_E. Now, consider $i \in [n-1]$. The slopes of $g_i(z)$ are non-increasing and there must exist at most one breakpoint $\hat{z}_i \in [0,1]$ such that the slope of $g_i(z)$ is 0 for all $\hat{z}_i \leq z \leq 1$. Otherwise, we can set $\hat{z}_i = 1$. Combining these two observations, we can assume without loss of generality that in an optimal solution $\boldsymbol{z}^*$ of (3.4), the first constraint of every linear program (3.3) is tight, i.e., $\sum_{j=1}^m x^*_{ij} = z^*_i$ for all $i \in [n]$.

The expectation for $\mathcal{S}$ from iteration ℓ, conditioned on reaching this iteration, is $\sum_{j=1}^m x^*_{i_\ell j} \cdot \xi_{i_\ell j} = g_{i_\ell}(z^*_{i_\ell})$. Overall, this means that iteration $\ell > 1$ of the for-loop is reached with probability $p_\ell := \prod_{\ell'=1}^{\ell-1}(1 - z^*_{\ell'})$. Note that even in the final iteration $\ell = k$, the recommendation is not sent deterministically. Rather, a coin is flipped whether to send a recommendation or not. This means that no signal could be sent during the for-loop. In this case, action n, the a priori best action for $\mathcal{R}$, is recommended. Hence, we refer to this as the “backup”. To lower bound the performance of the algorithm, we assume that a backup signal has a value of 0 for $\mathcal{S}$.

Thus, we can bound the overall performance ratio of our scheme by

$$\frac{u_{\mathcal{S}}(\varphi)}{f(S)} \geq \frac{\sum_{\ell=1}^{k} g_{i_\ell}(z^*_{i_\ell}) \cdot p_\ell}{\sum_{\ell=1}^{k} g_{i_\ell}(z^*_{i_\ell})} = \frac{\sum_{\ell=1}^{k} u_{i_\ell} \cdot z^*_{i_\ell} \cdot p_\ell}{\sum_{\ell=1}^{k} u_{i_\ell} \cdot z^*_{i_\ell}} \quad , \tag{3.9}$$

where we use the notation $u_{i_\ell} = \frac{g_{i_\ell}(z^*_{i_\ell})}{z^*_{i_\ell}}$. If $g_{i_\ell}(z^*_{i_\ell}) = 0$ for some ℓ, no value can be extracted from the corresponding action. Thus, we can exclude that action and only consider the remaining $k-1$ actions for the ratio in (3.9). Hence, we assume without loss of generality that $g_{i_\ell}(z^*_{i_\ell}) > 0$ and therefore $u_{i_\ell} > 0$ for all $\ell \in [k]$. Since the actions are ordered by non-increasing ratio $u_{i_\ell} = \frac{g_{i_\ell}(z^*_{i_\ell})}{z^*_{i_\ell}}$, we have $u_{i_1} \geq u_{i_2} \geq \cdots \geq u_{i_k}$ as well as

$$\frac{z^*_{i_1} p_1}{z^*_{i_1}} \geq \frac{z^*_{i_2} p_2}{z^*_{i_2}} \geq \cdots \geq \frac{z^*_{i_k} p_k}{z^*_{i_k}} \quad .$$

Hence, we can use Lemma 3.25 to see that

$$\begin{aligned}
\frac{u_{\mathcal{S}}(\varphi)}{f(S)} &\geq \frac{\sum_{\ell=1}^{k} u_{i_\ell} \cdot z^*_{i_\ell} \cdot p_\ell}{\sum_{\ell=1}^{k} u_{i_\ell} \cdot z^*_{i_\ell}} \geq \frac{\sum_{\ell=1}^{k} z^*_{i_\ell} \cdot p_\ell}{\underbrace{\sum_{\ell=1}^{k} z^*_{i_\ell}}_{=1}} = \sum_{\ell=1}^{k} z^*_{i_\ell} \cdot p_\ell \\
&= 1 - \prod_{i=1}^{k} (1 - z^*_{i_\ell}) \\
&\geq 1 - \left(1 - \frac{1}{k}\right)^k \quad .
\end{aligned}$$

For the second line, observe that $\sum_{\ell=1}^{k} z^*_{i_\ell} \cdot p_\ell$ is the probability that a recommendation is sent during the for-loop in line 3 of ComputeSignal and $\prod_{\ell=1}^{k}(1 - z^*_{i_\ell})$ is the complementary probability, i.e., that no recommendation is given. For the third line, observe that $1 - \prod_{\ell=1}^{k}(1 - z^*_{i_\ell})$ is symmetric and convex in every variable $z^*_{i_\ell}$. As such, it has a global minimum at $z^*_{i_1} = \ldots = z^*_{i_k} = 1/k$. $\square_{\text{Lemma 3.24}}$

Combining the previous lemmas allows to bound the approximation ratio. To complete the proof of Theorem 3.19, we need to show persuasiveness of φ_{IS} and bound the running time of the scheme. We proceed to show persuasiveness of the scheme.

Lemma 3.26
ComputeSignal returns a direct and persuasive signaling scheme for independent instances with k signals.

To prove persuasiveness of the scheme, we show that for every recommended action, the expected value for $\mathcal{R}$ is at least ϱ_E and further argue why this is sufficient for persuasiveness.

Proof. To show persuasiveness, we again use that without loss of generality, we can assume $\sum_{j=1}^m x_{ij} = z_i$ for all $i \in S \cup \{n\}$ as argued in the proof of Lemma 3.24 above.

Using this insight, we prove persuasiveness. In particular, for every choice of the set S of actions with $S \subseteq [n-1]$ and $|S| = k-1$, we show that ComputeSignal computes a direct and persuasive signal.

For each action $i \in S \cup \{n\}$, ComputeSignal observes the type realization and uses the optimal solution $\boldsymbol{x}^*$ for LP (3.3) to flip an independent coin whether or not to recommend action i. First, condition on the event that the scheme returns the signal for action $i_\ell \in S$ in the for-loop in line 3. The probability that the signal is sent in iteration ℓ is $\sum_{j=1}^m q_{i_\ell j} \cdot \frac{x^*_{i_\ell j}}{q_{i_\ell j}} = z^*_{i_\ell}$. We again use $p_\ell = \prod_{\ell'=1}^{\ell-1}(1 - z^*_{i_{\ell'}})$ to denote the probability that the scheme arrives in that iteration. A signal for action $i_\ell \neq n$ yields a conditional expected utility for $\mathcal{R}$ of

$$\frac{1}{p_\ell \cdot z^*_{i_\ell}} \cdot p_\ell \cdot \sum_{j=1}^m q_{i_\ell j} \cdot \frac{x^*_{i_\ell j}}{q_{i_\ell j}} \cdot \varrho_{i_\ell j} = \frac{1}{z^*_{i_\ell}} \cdot \sum_{j=1}^m x^*_{i_\ell j} \cdot \varrho_{i_\ell j} \geq \varrho_E \ ,$$

where the inequality follows from $\sum_{j=1}^m x^*_{i_\ell j} = z^*_{i_\ell}$ and the second constraint in (3.3).

Now, consider the case that ComputeSignal recommends action n. There are two scenarios for this to occur. First, during iteration ℓ with $i_\ell = n$, the coin flip for action n can be to recommend that action, and second, if no recommendation for an action was sent during the for-loop in line 3, action n is recommended in line 7 of ComputeSignal.

In the first scenario, the expected utility for $\mathcal{R}$ is

$$p_\ell \cdot \sum_{j=1}^m q_{i_\ell,j} \cdot \frac{x^*_{i_\ell,j}}{q_{i_\ell,j}} \cdot \varrho_{i_\ell,j} = p_\ell \cdot \sum_{j=1}^m x^*_{i_\ell,j} \cdot \varrho_{i_\ell,j} \ .$$

In the second scenario, no recommendation was sent during the for-loop. Again, we assume that $i_\ell = n$ and denote the probability that no signal was sent in an iteration $\ell' \neq \ell$ by $p_{-\ell} = \prod_{\ell' \neq \ell}(1 - z_{i_{\ell'}})$. Additionally, during iteration ℓ, no signal must have been sent, either. This means that $\mathcal{R}$ obtains an expected utility of

$$p_{-\ell} \cdot \sum_{j=1}^m q_{nj} \cdot \left(1 - \frac{x^*_{nj}}{q_{nj}}\right) \cdot \varrho_{nj} = p_{-\ell} \cdot \left(\varrho_E - \sum_{j=1}^m x^*_{nj} \varrho_{nj}\right)$$

since $\sum_{j=1}^m q_{nj} \cdot \varrho_{nj} = \varrho_E$. Overall, conditional on a signal for action n, $\mathcal{R}$ gets an expected utility of

$$\begin{aligned}
\frac{p_\ell \cdot \sum_{j=1}^m x^*_{nj} \cdot \varrho_{nj} + p_{-\ell}\left(\varrho_E - \sum_{j=1}^m x^*_{nj} \cdot \varrho_{nj}\right)}{p_\ell \cdot z^*_n + p_{-\ell} \cdot (1 - z^*_n)} &= \frac{p_{-\ell} \cdot \varrho_E + (p_\ell - p_{-\ell}) \cdot \sum_{j=1}^m x^*_{nj} \cdot \varrho_{nj}}{p_{-\ell} + (p_\ell - p_{-\ell}) \cdot z^*_n} \\
&\geq \frac{\varrho_E \cdot (p_{-\ell} + (p_\ell - p_{-\ell}) \cdot z^*_n)}{p_{-\ell} + (p_\ell - p_{-\ell}) \cdot z^*_n} \\
&= \varrho_E \ ,
\end{aligned}$$

where the inequality follows from the equality $z^*_n = \sum_{j=1}^m x^*_{nj}$ and the second constraint in (3.3).

Hence, for every recommended action $i \in S \cup \{n\}$, the expected utility for $\mathcal{R}$ is at least ϱ_E. Thus, deviating to any action $i' \notin S \cup \{n\}$ is not profitable for $\mathcal{R}$, since the type of action i' is independent of the signal, and every action a priori has an expected utility of at most ϱ_E for $\mathcal{R}$. It remains to show that a deviation to a different action $\hat{i} \in S \cup \{n\}$ is also not profitable for $\mathcal{R}$.

Assume that a recommendation is given for action $i_\ell \neq n$ during iteration ℓ of the for-loop in line 3 of ComputeSignal. We have already seen that conditional on that signal, the expected utility for $\mathcal{R}$ from action i_ℓ is at least ϱ_E. Consider an action $i_{\ell'}$ from an earlier iteration $\ell' < \ell$. ComputeSignal did not issue a recommendation for $i_{\ell'}$. The a priori expectation of action $i_{\ell'}$ is at most ϱ_E by assumption. Further, a recommendation for action $i_{\ell'}$ means that the conditional expectation is at least ϱ_E as we have seen above. Hence, no recommendation for action $i_{\ell'}$ cannot mean an expected utility of more than ϱ_E. Similarly, a deviation to an action $i_{\ell'} \in S \cup \{n\}$ in a later iteration $\ell' > \ell$ cannot be profitable, either. The scheme has not yet considered the type of that action, hence, the only information available on that action is the prior and the action has an expected utility for $\mathcal{R}$ of at most ϱ_E.

Similarly, consider the case that a recommendation for action n is signaled. Obviously, the receiver cannot know whether it was sent during or after the for-loop. Yet, we have already seen that a signal for action n gives a conditional expectation of at least ϱ_E to the receiver. All other actions $i_{\ell'} \neq n$ that have been considered (which might be all of them) only provide an expected value of at most ϱ_E to $\mathcal{R}$, just as all actions that have not yet been considered by the scheme.

Hence, it is not profitable for $\mathcal{R}$ to deviate from a recommendation and the scheme is persuasive. □

The final step to proving Theorem 3.19 is the running time of the algorithms. ActionsGreedy solves (3.4) an $O(nk)$ number of times to compute the set S. ComputeSignal has to solve (3.4) only once. Afterwards, at most k independent coin flips are computed. Clearly, both algorithms can be implemented to run in time polynomial in the representation size of the input. This concludes the proof of Theorem 3.19.

3.2.2 Improved Approximation and Tightness

In this section, we describe a more elaborate scheme to improve the approximation ratio of the scheme φ_{IS} from the previous section. The scheme will aptly be named Improved Independent Scheme φ_{IIS}. It still uses the same two-step approach of first identifying a good subset S of size $k-1$ and then computing the recommendation scheme for the subset $S \cup \{n\}$. Note that the second step cannot be improved upon using our approach, there are instances in which $\mathcal{S}$ can recover at most a fraction of $1 - (1 - 1/k)^k$ of $f(S)$.

Consider the following ϱ_E-optimal instance with IID actions, where every action $i \in [n]$ only has two possible types $\{\theta_1, \theta_2\}$. θ_1 has a probability of $1/k$ and provides a utility-pair of $(\varrho(\theta_1), \xi(\theta_1)) = (1, 1)$ and θ_2 has a probability of $1 - 1/k$ and gives no utility to $\mathcal{R}$ or $\mathcal{S}$, i.e., $(\varrho(\theta_2), \xi(\theta_2)) = (0, 0)$. Observe that any set $S \subseteq [n-1]$ of $k-1$ actions will have $f(S) = 1$ with $z_i = x_{i1} = 1/k$ for all $i \in S \cup \{n\}$. An optimal scheme will always recommend an action of type θ_1 if such an action exists – but this will only happen with probability $1 - (1 - 1/k)^k$. Hence, even an optimal scheme can only recover a fraction of $1 - (1 - 1/k)^k$ of $f(S)$. Thus, Lemma 3.24 is tight and we will have to consider the first step of the approach to improve the approximation ratio of the scheme.

The main insight of this section is summarized in the following proposition.

Proposition 3.27
For every $k \geq 2$ and every constant $\varepsilon > 0$, there is a polynomial-time algorithm to compute a subset S of $k-1$ actions such that

$$f(S) \geq (1-\varepsilon) \cdot \left(1 - \frac{1}{k}\right) \cdot u_{\mathcal{S}}(\varphi^*) \ .$$

Before going into details on the algorithm, we give a short overview of our approach. Rather than using the standard greedy algorithm for submodular maximization as in the previous section, we use an FPTAS to compute, for every given constant $\varepsilon > 0$, a set $S \subseteq [n-1]$ of $k-1$ actions. Compared to the set $S^* \subseteq [n-1]$ of size $k-1$ which maximizes f, this gives us an approximation ratio of $\frac{f(S)}{f(S^*)} \geq 1 - \varepsilon$, a notable improvement to the ratio of $(1-(1-{}^{1}/_{k})^{k-1})$ of Lemma 3.22 achieved by ActionsGreedy.

The Improved Independent Scheme uses a discretized version $\hat{f}$ of the submodular function f. For all $i \in [n]$, we only allow discretized values for z_i, i.e., we use the additional constraint that $z_i \in \{0, \tau, 2\tau, \ldots, ({}^{1}/_{\tau} - 1) \cdot \tau, 1\}$, where $\tau = \frac{1}{\lceil k/\delta \rceil}$ for $\delta = {}^{\varepsilon}/_{2}$. This discretization yields a $(1-\delta)$-approximation to the original function f, i.e., $\hat{f}(S) \geq (1-\delta) \cdot f(S)$ for each subset $S \subseteq [n-1]$. We then use a knapsack-style FPTAS to find a subset S of size $k-1$ such that $\hat{f}(S) \geq (1-\delta) \cdot \hat{f}(S^*)$ for every constant $\delta > 0$. Here, S^* is the subset of size $k-1$ maximizing f. This means that $\hat{f}(S) \geq (1-\delta) \cdot \hat{f}(S^*) \geq (1-\delta)^2 \cdot f(S^*) \geq (1-\varepsilon) \cdot f(S^*)$ since we chose $\delta = {}^{\varepsilon}/_{2}$. Using the same notation as in the proof of Lemma 3.22, we denote by K the set of actions of size k being recommended by the optimal scheme φ^* as well as by $S_k^* \subseteq [n-1]$ a subset of size k maximizing function f. By submodularity of f, we have $f(S^*) \geq \frac{k-1}{k} f(S_k^*)$. As we have seen before, it holds that $f(S_k^*) \geq f(K) \geq u_{\mathcal{S}}(\varphi^*)$ which concludes the high level overview of Proposition 3.27.

The second step of our approach, namely computing the final signal using ComputeSignal, remains unchanged. Since the analysis of ComputeSignal was done for every subset S, it carries over. In terms of running time, we need to solve LP (3.3) at most $O\left(\frac{n \cdot k}{\varepsilon}\right)$ times for the discretization. The FPTAS then requires a time of at most $O\left(\frac{n^2 \cdot k^6}{\varepsilon^3}\right)$ and we get the result of Theorem 3.20 by combining Proposition 3.27 with Lemmas 3.24 and 3.26.

In the following, we describe and analyze the FPTAS in detail.

Discretization of f

Rather than considering the original submodular function f, we consider a discretized version $\hat{f}$. For a given constant $\delta > 0$, we define $\tau = \frac{1}{\lceil k/\delta \rceil}$ and add the following constraint to the definition of f in (3.4). For every action $i \in [n]$, the total mass z_i can only take one of ${}^{1}/_{\tau} + 1$ values in $[0,1]$, namely, $z_i \in \{0, \tau, 2\tau, 3\tau, \ldots, \frac{1-\tau}{\tau} \cdot \tau, 1\}$.

Lemma 3.28
Consider the subset $S^* \subseteq [n-1]$ that maximizes $f(S^*)$. It holds that

$$\hat{f}(S^*) \geq (1-\delta) \cdot f(S^*) \ .$$

Proof. We can assume without loss of generality that the optimal set S^* maximizing $f(S)$ has a size of $k-1$. Otherwise, adding additional actions to a set of smaller size $k' < k$ and setting their respective masses to 0 does not change the value of $f(S)$. Let $\boldsymbol{z}^*$ denote a corresponding optimal assignment of mass to the actions of S^* for $f(S^*)$. This implies that $\sum_{i \in S^* \cup \{n\}} z_i^* \leq 1$ by the first constraint of (3.4). Now, for each $i \in S^* \cup \{n\}$, we define $z_i' = (1-\delta) \cdot z_i^*$. Since the functions g_i are monotone and concave, this means that $g_i(z_i') \geq (1-\delta) \cdot g_i(z_i^*)$ for all $i \in S^* \cup \{n\}$. We round z_i' up to the next multiple of τ, i.e., $\hat{z}_i = \tau \cdot \lceil z_i'/\tau \rceil$ for all $i \in S \cup \{n\}$. Then, $\hat{\boldsymbol{z}}$ is a feasible solution for (3.4). Obviously, $\hat{z}_i \geq 0$ which satisfies the second constraint. For the first constraint, we have

$$\sum_{i \in S^* \cup \{n\}} \hat{z}_i \leq \sum_{i \in S^* \cup \{n\}} z_i' + \tau = \sum_{i \in S^* \cup \{n\}} (1-\delta) \cdot z_i^* + \tau \leq (1-\delta) + k \cdot \frac{1}{\lceil k/\delta \rceil} \leq 1 \ .$$

For the value of $\hat{f}(S^*)$, this means

$$\hat{f}(S^*) \geq \sum_{i \in S^* \cup \{n\}} g_i(\hat{z}_i) \geq \sum_{i \in S^* \cup \{n\}} g_i(z_i') \geq (1-\delta) \cdot \sum_{i \in S^* \cup \{n\}} g_i(z_i^*) = (1-\delta) \cdot f(S^*) \ ,$$

which concludes the proof of the lemma. □

The discretization of $\hat{f}$ allows us to rephrase the optimization problem. Instead of a continuous mass of 1, we want to distribute $1/\tau$ *particles* to the actions $S \cup \{n\}$. We denote the marginal profit of the ℓ-th particle assigned to action i by $\mu_i^\ell = g_i(\ell \cdot \tau) - g_i((\ell-1) \cdot \tau)$. Since the functions g_i are monotone and concave, we observe that $\mu_i^\ell \geq 0$ and $\mu_i^\ell \geq \mu_i^{\ell+1}$ for all $i \in [n]$ and all $\ell \geq 1$. For a given set S, the optimal assignment of particles can clearly be computed in a greedy fashion: Sort the marginal profits of all actions $i \in S \cup \{n\}$ in a non-increasing order and assign them in that order until $1/\tau$ particles have been assigned. In order to find the set $\hat{S}^*$ optimizing $\hat{f}(S)$ among all subsets $S \subseteq [n-1]$ of size $k-1$, we consider the marginal profit μ^* of the last particle assigned to an action $i \in \hat{S}^* \cup \{n\}$ by the above greedy algorithm. Clearly, we do not know $\hat{S}^*$ and hence do not know μ^* in advance. Our approach will be to guess μ^* by considering all possible values μ. Since there are n functions g_i, and each function is allocated at most $1/\tau$ particles, we have to test at most $O(n \cdot k/\delta)$ many different values of marginal profit μ.

For a given marginal profit value μ, we denote by $\ell_i(\mu)$ the largest number of a particle with marginal profit strictly larger than μ. This means that if $i \in \hat{S}^*$ and $\mu = \mu^*$, a mass of $\hat{z}_i \geq \tau \cdot \ell_i(\mu)$ will be assigned to action i in an optimal assignment achieving $\hat{f}(\hat{S}^*)$. Depending on the remaining particles for action i and their respective marginal profit, more than $\ell_i(\mu)$ particles can be distributed to action i – but a mass of at least $\tau \cdot \ell_i(\mu)$ is *required* for μ to be the marginal profit of the final particle assigned to an action. Hence, we denote this mass by $w_i^r(\mu) = \tau \cdot \ell_i(\mu)$ and the corresponding value by $p_i^r(\mu) = g_i(\tau \cdot \ell_i(\mu))$. This allows us to express $\hat{f}(\hat{S}^*)$ by

$$\begin{aligned}
\hat{f}(\hat{S}^*) &= \sum_{i \in \hat{S}^* \cup \{n\}} g_i(\hat{z}_i) \\
&= \sum_{i \in \hat{S}^* \cup \{n\}} g_i(\tau \cdot \ell_i(\mu^*)) + \mu^* \cdot \sum_{i \in \hat{S}^* \cup \{n\}} (\hat{z}_i - \tau \cdot \ell_i(\mu^*))
\end{aligned}$$

$$= \sum_{i \in \hat{S}^* \cup \{n\}} p_i^r(\mu^*) + \mu^* \cdot \left(1 - \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(\mu^*)\right) \; .$$

In the second and third line, we have split up g_i in terms of particles that provide a marginal profit of strictly more than μ^* as well as the remaining particles which provide a marginal profit of exactly μ^*. We now consider these particles more closely.

Suppose there are $t_i(\mu) \geq 0$ particles for action i with marginal profit μ, namely the particles $\ell_i(\mu) + 1, \ell_i(\mu) + 2, \dots, \ell_i(\mu) + t_i(\mu)$. Since $\mu^* = \mu$, this means that $\hat{z}_i \in [\tau \cdot \ell_i(\mu), \tau \cdot \ell_i(\mu) + t_i(\mu)]$. Particles with a marginal profit of μ are somewhat *optional* as not necessarily all particles with marginal profit μ will be assigned. Hence, we denote the maximum additional mass from these particles by $w_i^o(\mu) = \tau \cdot t_i(\mu)$ and their maximum additional profit by $p_i^o(\mu) = \mu \cdot w_i^o(\mu) = \mu \cdot \tau \cdot t_i(\mu)$. We observe the following relationship between $\sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(\mu^*)$ and $\sum_{i \in \hat{S}^* \cup \{n\}} w_i^o(\mu^*)$.

$$0 \; \leq \; 1 - \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(\mu^*) \; \leq \; \sum_{i \in \hat{S}^* \cup \{n\}} w_i^o(\mu^*) \; . \tag{3.10}$$

In the following, we state the optimization problem for a given value μ of marginal profit as parameterized knapsack problem.

Knapsack Problem

We use the following integer optimization problem to find a subset $S \subseteq [n-1]$ of at most $k-1$ actions such that $1/\tau$ particles with marginal profit at least μ from the set of actions $S \cup \{n\}$ can be assigned and the overall profit from the assigned particles is maximized.

$$\begin{aligned} h(\mu) = \quad \text{Max.} \quad & \sum_{i=1}^{n} y_i \cdot p_i^r(\mu) + \min\left(\mu - \mu \cdot \sum_{i=1}^{n} y_i \cdot w_i^r(\mu),\; \sum_{i=1}^{n} y_i \cdot p_i^o(\mu)\right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i \cdot w_i^r(\mu) \;\leq\; 1 \\ & \sum_{i=1}^{n-1} y_i \;\leq\; k-1 \\ & y_n \;=\; 1 \\ & y_i \;\in\; \{0,1\} \quad \forall i \in [n-1] \end{aligned} \tag{3.11}$$

The variable $y_i \in \{0, 1\}$ for $i \in [n]$ denotes whether action i is included in the set $S \cup \{n\}$. Hence, the penultimate constraint, $y_n = 1$, is trivial. The second constraint states that at most $k - 1$ actions from $[n - 1]$ can be chosen for the set S. The first constraint bounds the total mass that is distributed to the actions $i \in S \cup \{n\}$ through the required particles for a given value μ of marginal profit. The objective function maximizes the total profit from particles of marginal profit at least μ, where the first term captures the profit from the required particles. The second term of the objective function either fills the remaining mass with optional particles of marginal profit μ or adds all remaining particles with marginal profit μ. For a given value μ, we denote the optimal solution for $h(\mu)$ by $\boldsymbol{y}^*$ and the action set optimizing $h(\mu)$ by $S_\mu^* = \{i \in [n-1] \mid y_i^* = 1\}$.

Lemma 3.29
For every marginal profit μ, the following holds:

(a) If $h(\mu)$ is feasible, then $h(\mu) \leq \hat{f}(\hat{S}^*)$.

(b) If $\mu = \mu^*$, then $h(\mu^*)$ is feasible and $h(\mu^*) = \hat{f}(\hat{S}^*)$.

(c) If $h(\mu)$ is infeasible, then $\mu \neq \mu^*$.

Proof. We start by proving (c). If $h(\mu)$ is infeasible then for every subset $S \subseteq [n-1]$ with at most $k-1$ actions, the first constraint is violated, i.e., $\sum_{i \in S \cup \{n\}} w_i^r(\mu) > 1$. This violates (3.10) which implies $\mu \neq \mu^*$, thereby showing (c).

To show (a) and (b), consider a marginal profit value μ and a feasible solution $\boldsymbol{y}$ for $h(\mu)$ with corresponding action set $S = \{i \in [n-1] \mid y_i = 1\}$. If S and μ satisfy (3.10), i.e., at most $1/\tau$ particles have a marginal profit greater than μ and a total of $1/\tau$ particles can be assigned using particles with marginal profit at least μ, then $h(\mu) = \hat{f}(S)$. Clearly, the resulting assignment of particles for set S is the same as in the optimal greedy strategy described above – assigning particles by non-increasing marginal profit. If a feasible solution $\boldsymbol{y}$ for a marginal profit value μ does not satisfy (3.10), it holds that

$$\sum_{i \in S \cup \{n\}} w_i^r(\mu) + \sum_{i \in S \cup \{n\}} w_i^o(\mu) < 1 \ .$$

This means that in h, the set S only yields a value of

$$\begin{aligned} &\sum_{i \in S \cup \{n\}} p_i^r(\mu) + \min\left(\mu - \mu \sum_{i \in S \cup \{n\}} w_i^r(\mu), \sum_{i \in S \cup \{n\}} p_i^o(\mu)\right) \\ &\quad = \sum_{i \in S \cup \{n\}} p_i^r(\mu) + \sum_{i \in S \cup \{n\}} p_i^o(\mu) \ , \end{aligned}$$

whereas in $\hat{f}(S)$ the remaining mass would be filled using particles that provide a marginal profit less than μ. This clearly holds for $S = S_\mu^*$ as well, which implies $h(\mu) \leq \hat{f}(S_\mu^*) \leq \hat{f}(\hat{S}^*)$, proving (a).

To show (b), it remains to show that $\hat{S}^*$ is indeed feasible for $h(\mu^*)$. This is sufficient since we have already seen above that $h(\mu) = \hat{f}(S)$ if S is feasible for $h(\mu)$ and μ and S satisfy (3.10). It is straightforward to see that (3.10) holds for μ^* and $\hat{S}^*$. Furthermore, $\hat{S}^*$ is clearly feasible for $h(\mu^*)$ since $|\hat{S}^*| = k-1$ by definition and (3.10) implies the first constraint of (3.11). This concludes the proof. □

Lemma 3.29 allows us to approximate $\hat{f}(\hat{S}^*)$ by approximating $h(\mu)$ for every possible marginal profit μ. In the following, we describe an FPTAS which guarantees, for every given constant $\delta > 0$, an approximation ratio of $(1-\delta)$.

Knapsack Approximation via Dynamic Program

Note that the constraints of (3.11) – except for $y_n = 1$ – exactly represent the constraints of the "1.5-dimensional" or "cardinality constrained" knapsack problem [56, Section 9.7]. To fit the knapsack terminology, we now consider items rather than individual particles. For every marginal profit μ and every action $i \in [n]$, there is a required item with weight $w_i^r(\mu)$ and profit $p_i^r(\mu)$ as well as an optional item with weight $w_i^o(\mu)$ and profit $p_i^r(\mu)$. The constraints mandate that for all i with $y_i = 1$, required items are included completely while there is no constraint on the inclusion of optional items. The objective function limits the profit from optional items of the chosen actions by the remaining capacity in the knapsack. Hence, arbitrary fractions of optional items can be included. Note that all optional items provide profit of a rate of μ per unit of weight by definition. Therefore, it does not make a difference which optional item is (partially) included for a given marginal profit value μ. Clearly, if the chosen required items leave enough capacity in the knapsack to include another required item from some action $i \notin S$ (and $|S| < k - 1$), including that item increases the overall profit since all required items have a profit rate of strictly more than μ.

Our approach resembles the classic dynamic programming approach for the knapsack problem. If $w_n^r(\mu) > 1$ for some μ, we can immediately move on to the next value of μ since the constraints of (3.11) cannot be satisfied. Similarly, if $w_i^r(\mu) + w_n^r(\mu) > 1$ for some $i \in [n-1]$, we can drop action i from consideration for that particular value of μ. Hence, we can assume without loss of generality that $w_i^r(\mu) + w_n^r(\mu) \leq 1$ for all $i \in [n-1]$. For the dynamic program, we scale the profit values. We denote by $p_{max}(\mu) = \max\{p_i^r(\mu), \min(\mu, p_i^o(\mu)) \mid i \in [n]\}$ the maximum profit obtainable from a single item and use $\kappa = \frac{\delta \cdot p_{max}(\mu)}{2k}$ to scale the profits of all items, namely $\bar{p}_i^r = \left\lfloor \frac{p_i^r(\mu)}{\kappa} \right\rfloor$ and $\bar{p}_i^o = \left\lfloor \frac{p_i^o(\mu)}{\kappa} \right\rfloor$. This means that the maximum scaled profit of a single item – required or optional – is of size $O(k/\delta)$. The dynamic program uses a 4-dimensional table A which has the following interpretation. In each entry $A(i, j, \bar{p}^r, \bar{p}^o)$, the minimum weight of the packed set of required items satisfying the following conditions is stored.

1. All packed required items are from the set $[i] \cup \{n\}$,
2. the set of packed required items includes action n and exactly j other actions,
3. the scaled profit of the packed required items is exactly $\bar{p}^r$, and
4. the scaled profit of the optional items corresponding to the packed required items sums to $\bar{p}^o$.

Now, consider an entry $A(i, j, \bar{p}^r, \bar{p}^o)$. There might be different sets which allow for j packed required items from the set $[i]$ with cumulative scaled profit from required items of $\bar{p}^r$ and optional items of $\bar{p}^o$. By taking a set with minimal weight from the packed required items, the highest possible capacity is left unused to be filled by optional items, thereby increasing the overall scaled profit.

The table has $O\left(n \cdot k \cdot \frac{k^2}{\delta} \cdot \frac{k^2}{\delta}\right) = O\left(\frac{n \cdot k^5}{\delta^2}\right)$ many entries. There are n different choices for i, namely $i \in \{0, \ldots, n-1\}$, between 0 and $k-1$ actions beside n can be taken, and the scaled profit from packed items, required as well as optional, cannot exceed k times the scaled profit of a single item.

We fill in the table starting with scenarios with $i = 0$ and $j = 0$, i.e., only considering action n. Clearly, when only using action n, the only available required item has weight w_n^r and the optional items provide a scaled profit of $\bar{p}_n^o$. Hence, we can initialize

$A(0,0,\bar{p}_n^r,\bar{p}_n^o) = w_n^r$. The remaining entries with $i = 0$ and $j = 0$ are initialized with ∞, i.e., the base of the recursion is

$$\begin{aligned} A(0,0,\bar{p}_n^r,\bar{p}_n^o) &= w_n^r, && \text{and} \\ A(0,0,\ x\ ,\ y\ ) &= \infty, && \text{for every } x,y \in \{0,1,\ldots,k\cdot\lfloor k/\delta \rfloor\}, (x,y) \neq (\bar{p}_n^r,\bar{p}_n^o)\ . \end{aligned}$$

Clearly, when considering all subsets of $[i] \cup \{n\}$ which use j actions from $[i]$ to achieve scaled profits $\bar{p}^r$ and $\bar{p}^o$, either action i is included in the best subset or not. In the latter case, the best weight and the achieved profits still are the same as it was for the entry in $A(i-1,j,\bar{p}^r,\bar{p}^o)$, so the entry can just be kept. In the former case, only $j-1$ actions from the set $[i-1]$ can be considered and the weight as well as the scaled profits of action i have to be accounted for. In that case, the new value for $A(i,j,\bar{p}^r,\bar{p}^o)$ would be $w_i^r + A(i-1,j-1,\bar{p}^r-\bar{p}_i^r,\bar{p}^o-\bar{p}_i^o)$. Hence, we can fill in the table in increasing order of the parameters by setting

$$A(i,j,\bar{p}^r,\bar{p}^o) = \min\left\{\begin{matrix} A(i-1,j,\bar{p}^r,\bar{p}^o), \\ w_i^r + A(i-1,j-1,\bar{p}^r-\bar{p}_i^r,\bar{p}^o-\bar{p}_i^o) \end{matrix}\right\}\ ,$$

where we assume the entry is ∞ whenever the arguments become negative. Clearly, no entry is written multiple times and updating an entry only requires a constant number of steps, so the overall time to fill the table is linear in its size.

All entries with $A(i,j,\bar{p}^r,\bar{p}^o) \leq 1$ correspond to feasible solutions since the packed required items do not surpass the weight limitation. From these entries, we pick one that maximizes $\kappa\cdot\bar{p}^r + \min(\mu - \mu\cdot A(i,j,\bar{p}^r,\bar{p}^o), \kappa\cdot\bar{p}^o)$. This way, the total scaled profit from required items and optional items is maximized while not violating the capacity constraint of the knapsack.

Finding the best entry in the table can thus be achieved in time linear in its size, and the overall running time for a single value μ of marginal profit is $O\left(\frac{n\cdot k^5}{\delta^2}\right)$. Since there are at most $O(n\cdot k/\delta)$ many values of μ to check, the overall running time can be bounded by $O\left(\frac{n^2\cdot k^6}{\delta^3}\right)$, which is polynomial in n and k for every constant $\delta > 0$. It remains to show that this approach guarantees an approximation ratio of $1-\varepsilon$ of the optimum solution. We argue why this holds in the following.

Approximation Ratio of the FPTAS Approach

Recall that S_μ^* is an optimal set of actions for $h(\mu)$. For the scaled profit from S_μ^*, it holds that

$$\begin{aligned} &\sum_{i\in S_\mu^*\cup\{n\}} \kappa\cdot\bar{p}_i^r + \min\left(\mu - \mu\cdot\sum_{i\in S_\mu^*\cup\{n\}} w_i^r(\mu), \sum_{i\in S_\mu^*\cup\{n\}} \kappa\cdot\bar{p}_i^o\right) \\ &\qquad \geq \sum_{i\in S_\mu^*\cup\{n\}} (p_i^r(\mu)-\kappa) + \min\left(\mu - \mu\cdot\sum_{i\in S_\mu^*\cup\{n\}} w_i^r(\mu), \sum_{i\in S_\mu^*\cup\{n\}} (p_i^o(\mu)-\kappa)\right) \\ &\qquad \geq\ h(\mu) - 2k\cdot\kappa\ =\ h(\mu) - \delta\cdot p_{max}\ . \end{aligned}$$

Recall that $p_{max} = \max\{p_i^r(\mu), \min(\mu, p_i^o(\mu)) \mid i \in [n]\}$ is the maximum profit obtainable from a single item without overfilling the knapsack. We now want to argue that $h(\mu) \geq p_{max}$, thereby finishing the proof that an optimal solution from the dynamic program guarantees at least a $(1-\delta)$-approximation of $h(\mu^*)$.

If $p_i^r(\mu) = p_{max}$ for some $i \in [n-1]$, this clearly means that $h(\mu) \geq p_{max}$ since $w_i^r(\mu) + w_n^r(\mu) \leq 1$. The choice $S = \{i\}$ provides a feasible solution for (3.11) which implies $h(\mu) \geq p_i^r(\mu)$ since $p_n^r(\mu) \geq 0$. Adding parts of the optional items can only increase the overall value of the objective function. Similarly, if $p_n^r(\mu) = p_{max}$, consider $S = \emptyset$. This is a feasible solution for (3.11). Filling the remaining capacity $1 - w_n^r(\mu)$ of the knapsack with a fraction of the optional item of action n can only increase the overall value of the objective function. If $p_{max} = \min(\mu, p_i^o(\mu))$ for some $i \in [n-1]$, then consider the case that the knapsack is filled using only the optional item of action i. Here, μ is an upper bound to the profit since the knapsack only has a total capacity of 1 – which might be less than the weight of the optional item of action i. The following step increases the overall profit without violating the capacity constraint of the knapsack. Add the required items of actions n and i to the knapsack, replacing parts of the optional item if necessary. This is clearly possible since $w_i^r(\mu) + w_n^r(\mu) \leq 1$. Since the rate of profit per unit of weight for required items is higher than μ, the rate for optional items, this leads to an improvement of overall profit. Additionally, filling the remaining capacity in the knapsack using the optional item of action n yields the objective value for the feasible set $S = \{i\}$. Hence, $h(\mu) \geq p_{max}$. Similarly, if $p_{max} = \min(\mu, p_n^o(\mu))$, using the above strategy for the feasible set $S = \emptyset$ shows that $h(\mu) \geq p_{max}$.

Overall, the dynamic program computes a solution set S' for a marginal profit value μ which has the highest scaled profit. The unscaled profit of S' is at least the scaled profit of the same set. Since S' has the highest scaled profit of any set for marginal profit μ, its scaled profit is at least that of the set S_μ^* for any marginal profit μ, including the optimal marginal profit μ^* and corresponding optimal set S^*. Above, we have shown that the scaled profit of S_μ^* is at least $(1-\delta) \cdot h(\mu)$ for any μ. Hence, we have shown that the unscaled profit of S' is at least $(1-\delta) \cdot h(\mu^*)$. In combination with Lemmas 3.28 and 3.29, we get

$$f(S') \geq \hat{f}(S') \geq (1-\delta) \cdot h(\mu^*) = (1-\delta) \cdot \hat{f}(\hat{S}^*) \geq (1-\delta)^2 \cdot f(S^*) \geq (1-\varepsilon) f(S^*) \ .$$

3.2.3 Beyond ϱ_E-Optimality

In this section, we show that our approach for ϱ_E-optimal instances does not achieve a good approximation ratio for general independent instances. Finding a scheme which guarantees a constant-factor approximation for such instances remains an interesting open problem for future work.

Consider the following example with $n = 2$ actions and $k = 2$ signals, i.e., there can be a signal for both actions. The state space of action 1 is a singleton, $\Theta_1 = \{\theta_1\}$ with $(\varrho(\theta_1), \xi(\theta_1)) = (0, 1)$. Action 2 has two different types, $\Theta_2 = \{\theta_{21}, \theta_{22}\}$ with $(\varrho(\theta_{21}), \xi(\theta_{21})) = (1, 0)$ and $(\varrho(\theta_{22}), \xi(\theta_{22})) = (0, 0)$. Both types have a probability of $q_{21} = q_{22} = 1/2$. This means that $\varrho_E = 1/2$, realized by action 2.

An optimal direct scheme φ^* has the following form. In the first state of nature (θ_1, θ_{21}), action 2 is recommended. In the second state of nature (θ_1, θ_{22}), a signal for action 1 is sent. This way, $\mathcal{R}$ can enjoy full information and in return allow $\mathcal{S}$ to obtain some utility when there is no utility for $\mathcal{R}$ to be had. Hence, following the recommendation is a best response for $\mathcal{R}$ and the scheme is persuasive. The expected utility for $\mathcal{S}$ is $u_{\mathcal{S}}(\varphi^*) = 1/2$.

Now, consider our approach of solving (3.4). The second constraint of the linear program (3.3) requires the conditional expectation behind a signal for each action individually to be at least ϱ_E. Since the signals for action 1 can never provide a positive conditional expectation for $\mathcal{R}$, the optimal solution to (3.3) is $x_1^* = 0, x_{21}^* = x_{22}^* = 1/2$. Thus, $z_1^* = 0$, $z_2^* = 1$, and $g_1(z_1^*) = g_2(z_2^*) = 0$. This implies that the optimal value for the linear program (3.4) is 0. Clearly, the optimal value of (3.4) is not an upper bound for the expected sender utility of an optimal scheme.

On a more fundamental level, we can see using the example above that there are independent instances in which the sender's ability to extract value from one action depends on the state of another action. Our approach focuses on the individual states of the independent actions and does not take the possible correlation between different actions into account.

3.3 Guarantees for Limited Signals

In this final section of our discussion of offline Bayesian persuasion, we compare the expected utility $\mathcal{S}$ can extract when using a scheme with a restricted signal space with k signals to the expected utility of a scheme with at least n signals in the same instance.

We denote by OPT_k the expected sender utility of an optimal scheme with k signals. Since more than n signals do not provide additional value for $\mathcal{S}$, we will compare OPT_k to OPT_n.

We again discuss symmetric instances and independent, ϱ_E-optimal instances separately.

3.3.1 Symmetric Instances

In order to show a tight approximation ratio of k/n for symmetric instances, we define the following Imitation Scheme φ_{Imi} for instances with n actions and k signals. First, it runs an optimal symmetric and persuasive scheme φ_n^* for n signals and considers the resulting signal $i \in [n]$. If $i \in [k]$, φ_{Imi} forwards the signal to $\mathcal{R}$. Otherwise, φ_{Imi} draws $i' \in [k]$ uniformly at random and sends i' to $\mathcal{R}$. Hence, the scheme φ_{Imi} is clearly direct and symmetric – φ_n^* is symmetric and whenever the recommendation is not directly forwarded, a uniform random action from $[k]$ is recommended.

In the following proposition, we show that φ_{Imi} is persuasive and guarantees the approximation ratio of k/n for symmetric instances. The running time for the scheme depends on the running time of φ_n^*. An obvious candidate for φ_n^* would be the Slope-Algorithm discussed in Section 3.1.2. Hence, the existence of an efficient probability oracle for queries Q1 and Q2 (cf. Section 3.1.3) implies a polynomial running time in the input size of the instance.

Proposition 3.30
The Imitation Scheme is symmetric, direct, and persuasive in symmetric instances. For every $k \geq 1$ it holds that $u_{\mathcal{S}}(\varphi_{Imi}) \geq k/n \cdot \mathrm{OPT}_n$. There exists a random-order instance such that $\mathrm{OPT}_k \leq k/n \cdot \mathrm{OPT}_n$.

Proof. Before showing the upper and lower bounds on the approximation ratio, let us prove the remaining properties of φ_{Imi} claimed in the proposition. Clearly, φ_{Imi} is a direct and symmetric scheme. Let us now show that φ_{Imi} is persuasive. Consider a recommendation for action $i \in [k]$. With probability k/n, i was recommended by φ_n^* and with probability $1-k/n$, a different action $i' \notin [k]$ was recommended. Hence, conditional on a recommendation for i, the type distribution of this action is $\mathcal{D}_{yes}$ with probability k/n and $\mathcal{D}_{no}$ with probability $1-k/n$, where $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ are the distributions resulting from φ_n^*. If action i is not recommended, it was not recommended by φ_n^*, either. Hence, the underlying type distribution is $\mathcal{D}_{no}$. Since φ_{Imi} is a symmetric scheme, each action is recommended with a probability of $1/k$. Additionally, the a priori expectation for $\mathcal{R}$ for any action is ϱ_E. Hence, the following holds for every action $i \in [k]$

$$\frac{1}{k} \cdot \left(\frac{k}{n} \cdot \varrho_{yes} + \left(1 - \frac{k}{n}\right) \cdot \varrho_{no}\right) + \frac{k-1}{k} \cdot \varrho_{no} = \frac{1}{n} \cdot \varrho_{yes} + \left(1 - \frac{1}{n}\right) \cdot \varrho_{no} = \varrho_E \ ,$$

where ϱ_{yes} and ϱ_{no} denote the expected utility for $\mathcal{R}$ corresponding to distributions $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$, respectively. Since φ_n^* is symmetric and persuasive, $\varrho_{yes} \geq \varrho_E$. This in turn implies that $k/n \cdot \varrho_{yes} + (1 - k/n) \cdot \varrho_{no} \geq \varrho_E$. Since φ_{Imi} is a symmetric scheme, Lemma 3.10 shows that φ_{Imi} is a persuasive scheme.

For the approximation guarantee achieved by φ_{Imi}, we can see that with probability k/n, φ_{Imi} recommends the same action as φ_n^*. Since φ_n^* is symmetric, each action is recommended with equal probability. This means that $u_{\mathcal{S}}(\varphi_{Imi}) \geq k/n \cdot \mathrm{OPT}_n$ and φ_{Imi} guarantees an approximation ratio of k/n.

We finish the proof by showing the upper bound on OPT_k. To this end, we use a random-order instance with n different types $(\theta_1, \theta_2, \ldots, \theta_n)$, where $(\varrho(\theta_1), \xi(\theta_1)) = (1, 1)$ and $(\varrho(\theta_i), \xi(\theta_i)) = (0, 0)$ for all $i > 1$. Clearly, both $\mathcal{S}$ and $\mathcal{R}$ prefer θ_1 over all other states. Hence, if $\mathcal{S}$ has access to n signals, the expected utility for $\mathcal{S}$ and $\mathcal{R}$ is 1. If there are only k signals, Lemma 3.5 tells us that there exists an optimal signaling scheme which recommends the actions from $[k]$. Hence, if θ_1 is not among the first k actions, a type with utility 0 has to be recommended. This means that the probability that θ_1 can be recommended and thus the expected utility for $\mathcal{S}$ is only k/n, yielding the upper bound of $\mathrm{OPT}_k \leq k/n \cdot \mathrm{OPT}_n$. Together with the lower bound above, this yields a tight approximation. □

3.3.2 Independent Instances

We use an IID instance to show an upper bound of $O(k/n)$ on the approximation ratio for independent instances. Since IID instances are symmetric, they satisfy ϱ_E-optimality. Hence, we are able to show an asymptotically tight approximation ratio of $\Theta(k/n)$ for ϱ_E-optimal independent instances.

Lemma 3.31
There exists an IID instance such that $\mathrm{OPT}_k \leq \frac{e}{e-1} \cdot \frac{k}{n} \cdot \mathrm{OPT}_n$.

Proof. For every action, there exist only 2 different types in the distribution. A good type θ_1 with $q_{\theta_1} = 1/n$ and $(\varrho(\theta_1), \xi(\theta_1)) = (1, 1)$ as well as a bad type θ_0 with $q_{\theta_0} = 1 - 1/n$ and $(\varrho(\theta_0), \xi(\theta_0)) = (0, 0)$. If there is an action $i \in [k]$ with type θ_1, it is clearly optimal for $\mathcal{S}$ and $\mathcal{R}$ that action i is recommended. To bound the approximation ratio of $\frac{\mathrm{OPT}_k}{\mathrm{OPT}_n}$, first observe that

$$\mathrm{OPT}_k \geq \frac{k}{k+1} \mathrm{OPT}_{k+1} \quad \text{for all } k \geq 1. \tag{3.12}$$

Algorithm 5: ActionsReduce

Input: Distributions $\mathcal{D}_1 = (\Theta_1, \boldsymbol{q}_1), \mathcal{D}_2 = (\Theta_2, \boldsymbol{q}_2), \dots, \mathcal{D}_n = (\Theta_n, \boldsymbol{q}_n)$, s.t. $\sum_j q_{n,j} \cdot \varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j} \cdot \xi_{nj} = \max_{i \in [n] : \sum_j q_{i,j} \cdot \varrho_{ij} = \varrho_E} \sum_j q_{i,j} \cdot \xi_{ij}$, parameter $2 \leq k \leq n$

Compute $f([n-1])$.
For every $i \in [n]$, let z_i^* be the values of the optimal solution in $f([n-1])$.
Let S be the set of the $k-1$ actions from $[n-1]$ with largest values $g_i(z_i^*)$
return S

This holds because we can use φ_{Imi} to approximate the expected sender utility for k signals in the IID instance with $k+1$ actions. Since $u_\mathcal{S}(\varphi_{Imi})$ with k actions clearly is a lower bound to OPT_k, Proposition 3.30 shows (3.12).

It is straightforward to see that $\mathrm{OPT}_1 = 1/n$. Clearly, (3.12) implies $\frac{\mathrm{OPT}_k}{k} \geq \frac{\mathrm{OPT}_{k+1}}{k+1}$ and repeated application of this inequality shows that $\frac{1}{n} = \frac{\mathrm{OPT}_1}{1} \geq \frac{\mathrm{OPT}_k}{k}$ for all $k \geq 1$. Hence, we have

$$\frac{\mathrm{OPT}_k}{\mathrm{OPT}_n} \Big/ \frac{k}{n} = \frac{1}{\mathrm{OPT}_n} \cdot n \cdot \frac{\mathrm{OPT}_k}{k} \leq \frac{1}{\mathrm{OPT}_n} = \frac{1}{1 - \left(1 - \frac{1}{n}\right)^n} \quad .$$

Observe that $\frac{1}{1-\left(1-\frac{1}{n}\right)^n}$ monotonically increases with n and approaches $\frac{1}{1-\frac{1}{e}} = \frac{e}{e-1} \approx 1.582$ for $n \to \infty$. Hence, for every k and n, the approximation ratio of $\frac{\mathrm{OPT}_k}{\mathrm{OPT}_n}$ is at most $\frac{k}{n} \cdot \frac{e}{e-1}$. □

For the lower bound of $\Omega(k/n)$ for ϱ_E-optimal independent instances, we consider the following Independent-Imitation Scheme φ_{ImiIS}. Similar to the schemes in Sections 3.2.1 and 3.2.2, we use a two-step approach of first choosing a subset $S \subseteq [n-1]$ of $k-1$ actions and then, in the second step, computing the signaling scheme for $S \cup \{n\}$. For the latter step, we again use ComputeSignal (Algorithm 4). For the former step, we use the procedure ActionsReduce (Algorithm 5). In this algorithm, $f([n-1])$ (cf. (3.4)) is computed, which distributes a mass of 1 to all n actions. Then, the $k-1$ most profitable actions from $[n-1]$, i.e., the actions $i \in [n-1]$ with highest value $g_i(z_i^*)$, form the set S. Here, $\boldsymbol{z}^*$ constitutes the assignment of mass to the actions in the optimal solution of $f([n-1])$.

In both ComputeSignal and ActionsReduce, the running time is dominated by solving a linear program. Hence, the scheme can be implemented in time polynomial in the input size.

Theorem 3.32
The Independent-Imitation Scheme is direct and persuasive for ϱ_E-optimal independent instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$,

$$u_\mathcal{S}(\varphi_{ImiIS}) \geq \left(1 - \left(1 - \tfrac{1}{k}\right)^k\right) \cdot \left(1 - \tfrac{1}{k}\right) \cdot \frac{k}{n} \cdot \mathrm{OPT}_n \quad .$$

Proof. Observe that $f([n-1]) \geq \mathrm{OPT}_n$. This holds due to (3.5), i.e., $f(S) \geq u_\mathcal{S}\left(\varphi^*_{S \cup \{n\}}\right)$. We denote by z_i^* and $\boldsymbol{x}_i^*$ the optimal solutions for action $i \in [n]$ in

$f([n-1])$. Clearly, considering only the actions $i \in S$, z_i^* and $\boldsymbol{x}_i^*$ constitute a feasible solution for $f([n-1])$, and since the $k-1$ most profitable actions have been chosen, it holds that

$$\sum_{i \in S \cup \{n\}} g_i(z_i^*) \geq \frac{k-1}{n} \cdot f([n-1]) \geq \left(1 - \frac{1}{k}\right) \cdot \frac{k}{n} \cdot \mathrm{OPT}_n \ .$$

From Lemma 3.24, we know that the scheme resulting from ComputeSignal for a set S achieves at least a $(1-(1-{}^1\!/_k)^k)$-approximation of $f(S)$. Hence, the overall approximation follows. By Lemma 3.26, we know that ComputeSignal produces a direct and persuasive scheme for any set $S \subseteq [n-1]$, so φ_{ImiIS} is direct and persuasive. □

With this result, we conclude our discussion of offline Bayesian persuasion.

Chapter 4

Online Bayesian Persuasion

In this chapter, we discuss the Bayesian persuasion game with a dynamic component, namely with online arrival. In contrast to the setting in Chapter 3, the types of the actions will be revealed sequentially in a round-wise fashion and $\mathcal{S}$ has to send a signal in every round. The receiver – only knowing the signal – then has to immediately decide whether to take the current action or wait for the next one. If $\mathcal{R}$ decides to take the action, the process ends and $\mathcal{R}$ and $\mathcal{S}$ get their respective utilities. If, on the other hand, $\mathcal{R}$ decides to forego the action, it is lost forever and the next action is revealed to $\mathcal{S}$.

We discuss this main setting in two variants. In the first section, we consider a scenario with distributional information on the action types, similar to the independent setting in the previous chapter. Here, the instances are reminiscent of the classic prophet inequality problem [60]. In this setting, a gambler sees a sequence of n boxes which contain some a priori unknown prizes. The gambler is trying to find the best prize among the sequence. Each prize is drawn according to a known box-specific prior distribution. When opening a box, the gambler sees the prize inside but then has to immediately and irrevocably decide whether to take the current prize or forego it and be allowed to open the next box. The performance of the gambler is measured against an all-knowing prophet – who always picks the best prize. In the realm of Bayesian persuasion, we model this as follows. In every round, the sender observes the action type of the current action and sends a signal to the receiver, who then has to make a decision. We compare the performance of our online schemes to that of an optimal persuasive "prophet" sender, i.e., an optimal persuasive offline scheme.

In the second part, we consider instances that are more in line with the classic secretary problem [38], in which a decision maker tries to find the best candidate for a position. While the decision maker knows the total number of candidates n, there is no information on the quality of the candidates. The n candidates arrive one-by-one in uniform random order. The decision maker performs an interview upon the arrival of each candidate and uses this information to rank the interviewed candidates. Immediately after seeing a candidate, an irrevocable decision whether to hire the current candidate, thereby filling the position, or to go on the the next interview has to be made. The objective of the decision maker is to maximize the probability of hiring the best of the n candidates.

Similar to the previous setting, the sender again observes the type of the current action and sends a signal to $\mathcal{R}$ who has to decide whether to take the action or wait for the next round. Since the random order of actions provides symmetry for this setting, it is somewhat reminiscent of the symmetric instances in Chapter 3. While the online arrival certainly provides a new aspect, the possible types and hence the utility values of actions are completely unknown – not only drawn from known distributions.

Hence, the key difference between the variants discussed in this chapter lies in the information available a priori to $\mathcal{S}$ and $\mathcal{R}$. In both scenarios n, the total number of actions, is known. While the first variant offers a fixed order of distributions for the actions' types, the second one offers almost no information but the guarantee that actions are revealed in uniform random order.

4.1 Prophet Inequalities for Persuasion

In this section, we discuss our first variant of online Bayesian persuasion. The contents are based on [47]. The model we consider is somewhat similar to the model of Section 3.2 on independent instances in the previous chapter. There are n commonly known distributions $\mathcal{D}_1 = (\Theta_1, \boldsymbol{q}_1), \ldots, \mathcal{D}_n = (\Theta_n, \boldsymbol{q}_n)$. In each round $i = 1, \ldots, n$, the type θ_i of action i is independently drawn according to $\mathcal{D}_i$. Again, we assume that $|\Theta_i| = m$ for all $i \in [n]$, possibly using dummy types j' with $q_{ij'} = 0$ to fill up the type spaces.

The process is the following.

1. Both $\mathcal{S}$ and $\mathcal{R}$ know the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of types of the n actions.
2. $\mathcal{S}$ commits to a signaling scheme φ.
3. $\mathcal{R}$ learns the scheme φ.
4. In each round $i = 1, \ldots, n$:
    - 4.1. $\mathcal{S}$ learns the type θ_i of action i.
    - 4.2. $\mathcal{S}$ sends a signal $\sigma_i \in \Sigma$ according to φ to $\mathcal{R}$.
    - 4.3. $\mathcal{R}$ decides whether to take action i based on signals $\sigma_1, \ldots, \sigma_i$. If $\mathcal{R}$ takes action i, the process ends.

Again, we focus on direct and persuasive schemes. Due to the online nature of the problem, Lemma 2.1 is not directly applicable. Yet, one can see that it is still without loss of generality to only consider direct and persuasive schemes. A decision has to be made by $\mathcal{R}$ in every round. Either $\mathcal{R}$ takes the current action or foregoes it and waits for the next action. Hence, this can be reflected on the sender's side by a recommendation to take or not to take the current action. More intricate messages from $\mathcal{S}$ can still only lead to a decision whether to take the current action or not. Thus, replacing a message by signal YES whenever $\mathcal{R}$ would have taken the action and NO otherwise is without loss of generality. In combination, this means that $\sigma_i \in \{\text{YES}, \text{NO}\}$ for every round $i \in [n]$. Additionally, exactly one YES-signal is sent, making the sender's signal a direct recommendation for a single action. If $\mathcal{R}$ takes the corresponding action, the process ends. Otherwise, if $\mathcal{R}$ deviates, $\mathcal{S}$ will only send NO-signals, thereby not giving $\mathcal{R}$ further information. Thus, we can assume that $\mathcal{S}$ uses a direct scheme.

Additionally, if a direct scheme φ is not persuasive, $\mathcal{R}$ is not inclined to follow the recommendation given by φ and rather takes an action i without a recommendation. Hence, $\mathcal{S}$ can adapt φ and give a recommendation for action i such that $\mathcal{R}$ does not want to deviate from the recommendations. This changes neither the sender's nor the receiver's expected utility.

With these preliminaries, we begin our discussion of optimal online signaling. We start by discussing a simple scheme for IID instances.

4.1.1 A Simple Scheme for the IID-Case

In the previous chapter, we discussed the offline case of Bayesian persuasion and noted Theorem 3.2, a result by Dughmi and Xu [36]. They solve optimal signaling for IID actions in polynomial time using a linear program, leveraging the symmetry of the instances. They additionally give the following simpler linear program (4.1) for IID instances which uses fewer constraints.

$$\begin{array}{lrcll} \text{Max.} & n \cdot \sum_{j=1}^{m} x_j \cdot \xi_j & & & \\ \text{s.t.} & n \cdot \sum_{j=1}^{m} x_j & = & 1 & \\ & x_j + (n-1) \cdot y_j & = & q_j & \forall\, j \in [m] \\ & \sum_{j=1}^{m} x_j \cdot \varrho_j & \geq & \sum_{j=1}^{m} y_j \cdot \varrho_j & \\ & x_j, y_j & \geq & 0 & \forall\, j \in [m] \end{array} \tag{4.1}$$

The intuition for this program is that x_j is the ex-post probability of receiving a recommendation for type j of action i, where the symmetry of the instance allows us not to discriminate different actions $i \neq i'$ but rather consider a single probability x_j for every type $j \in [m]$. Additionally, this leads to the first constraint of $n \cdot \sum_{j=1}^{m} x_j = 1$, since all n actions are recommended with equal probability $1/n$. Similar to x_j, y_j is the ex-post probability of not getting a recommendation on type j. The third constraint relaxes the persuasiveness constraints of the original linear program by only requiring that taking a recommended action is at least as profitable as taking a non-recommended one for $\mathcal{R}$. Finally, the objective function represents the expected utility for $\mathcal{S}$. Note that the second constraint implies $y_j = \frac{q_j - x_j}{n-1}$ with $y_j \geq 0$ whenever $x_j \leq q_j$. Hence, the constraints $y_j \geq 0$, $x_j + (n-1) \cdot y_j = q_j$, and $\sum_{j=1}^{m} x_j \cdot \varrho_j \geq \sum_{j=1}^{m} y_j \cdot \varrho_j$ can be replaced by $x_j \leq q_j$ and $\sum_{j=1}^{m} x_j \cdot \varrho_j \geq \varrho_E \cdot \sum_{j=1}^{m} x_j$. To see this, observe that

$$\sum_{j=1}^{m} x_j \cdot \varrho_j \geq \sum_{j=1}^{m} y_j \cdot \varrho_j = \sum_{j=1}^{m} \frac{q_j - x_j}{n-1} \cdot \varrho_j = \sum_{j=1}^{m} \frac{q_j \cdot \varrho_j}{n-1} - \sum_{j=1}^{m} \frac{x_j \cdot \varrho_j}{n-1} \ ,$$

which implies $\sum_{j=1}^{m} x_j \cdot \varrho_j \geq \frac{1}{n} \sum_{j=1}^{m} q_j \cdot \varrho_j = \varrho_E \cdot \sum_{j=1}^{m} x_j$ due to $\sum_{j=1}^{m} x_j = 1/n$ and $\varrho_E = \sum_{j=1}^{m} q_j \cdot \varrho_j$.

This allows us to use the following equivalent formulation of linear program (4.1).

$$\begin{array}{llcll} \text{Max.} & n \cdot \sum_{j=1}^{m} x_j \cdot \xi_j & & & \\ \text{s.t.} & n \cdot \sum_{j=1}^{m} x_j & = & 1 & \\ & \sum_{j=1}^{m} x_j \cdot \varrho_j & \geq & \varrho_E \cdot \sum_{j=1}^{m} x_j & \\ & x_j & \in & [0, q_j] & \forall\, j \in [m] \end{array} \tag{4.2}$$

Observe that this linear program is very similar to (3.3) with $z = 1/n$. Since (4.2) is not only for independent but also identical distributions, symmetry can be used here. This allows us to get the overall expected utility from the objective function of a single linear program rather than having to solve a linear program for every action. Note that symmetry is enforced by the first constraint $\sum_{j=1}^{m} x_j = 1/n$ as well as the usage of a single variable x_j instead of individual variables x_{ij} for the different actions $i = 1, \ldots, n$.

Consider an optimal persuasive scheme φ^*. In Section 3.1.1, we have seen that φ^* can be assumed to be symmetric without loss of generality. Consider the corresponding ex-post distribution $\boldsymbol{x}^{\varphi^*}$. Clearly, it is a feasible solution for (4.2). Hence, an optimal solution $\boldsymbol{x}^*$ for (4.2) provides an upper bound on the sender's expected utility from a persuasive scheme. Since $\boldsymbol{x}^*$ only satisfies a relaxed set of constraints, it might not directly correspond to a persuasive signaling scheme. Dughmi and Xu turn $\boldsymbol{x}^*$ into the ex-post distribution of a persuasive scheme by using independent coin flips with probability $\frac{x^*_{\theta_i}}{q_{\theta_i}}$ for each action i with type θ_i. Out of all actions whose coins came up heads, their scheme randomly proposes one of these. If none of the coins came up heads, the scheme proposes a random action. Recall that Dughmi and Xu discuss the offline model, which means that the scheme has access to all realizations at the same time.

We adopt this approach as our *Simple Scheme for IID* (Algorithm 6) for the online model. The scheme ensures that exactly one YES-recommendation is sent. In each round $i \in [n-1]$, if no YES-signal has been sent, such a signal is issued independently with probability $\frac{x^*_{\theta_i}}{q_{\theta_i}}$. Otherwise, a NO-signal is sent. Finally, in the last round, the signal is YES if such a signal has not been sent, and otherwise the signal is NO. Clearly, the scheme only requires information available in round i to give a recommendation. Hence, it can be used online. It is persuasive and guarantees a $(1 - 1/e)$-approximation of the optimal persuasive scheme for the offline case.

Proposition 4.1
The simple scheme for IID is persuasive in the online setting and yields a $(1 - 1/e)$-approximation.

Proof. For persuasiveness, we need to show that it is in the receiver's interest to take action i if $\sigma_i =$ YES. By definition of the algorithm, $\sigma_i =$ YES for exactly one $i \in [n]$. Let us assume that $\sigma_i =$ YES for $i < n$. Then, $\mathcal{R}$ gets a conditional expected utility of

$$\frac{\sum_{j=1}^{m} q_j \cdot \frac{x_j}{q_j} \cdot \varrho_j}{\sum_{j=1}^{m} q_j \cdot \frac{x_j}{q_j}} = \frac{\sum_{j=1}^{m} x_j \cdot \varrho_j}{\sum_{j=1}^{m} x_j} \geq \varrho_E$$

Algorithm 6: Simple Scheme for IID

```
Input: Distribution D = (Θ, q), number of rounds n
Compute an optimal solution x* for (4.2).
Set recSent = False.
for round i = 1, ..., n − 1 do
    Observe type θ_i of the action in round i.
    if recSent = True then Signal NO.
    else
        Draw x ~ Unif[0,1].
        if x ≤ x*_{θ_i} / q_{θ_i} then Send signal YES and set recSent = True.
        else Send signal NO.
for round n do
    if recSent = False then Send signal YES.
    else Send Signal NO.
```

by following the recommendation. Not taking action i and waiting for some round $i' > i$ only gives $\mathcal{R}$ an expected utility of ϱ_E since $\mathcal{S}$ will not provide any information after the first signal YES. Additionally, each action has an a priori expectation of ϱ_E for $\mathcal{R}$. Since the conditional expectation for a YES-signal is at least ϱ_E, this means that the conditional expectation for a NO-signal can be at most ϱ_E. Hence, for rounds $i = 1, \ldots, n-1$, the scheme is persuasive. Finally, if round n is reached without a previous YES-signal, the signal is σ_n = YES regardless of the type of action n. This means that the expectation for $\mathcal{R}$ is ϱ_E. Clearly, this shows that the scheme is persuasive which means that $\mathcal{R}$ will follow the recommendations given by $\mathcal{S}$.

To compute the expected utility for $\mathcal{S}$, first consider round n. By construction of the scheme, the signal is always YES if no previous YES-signal was sent. For the sake of analysis, let us consider a more elaborate approach. Either, the signal is a "regular" YES-signal due to (4.2), or, if the mechanism were to decide to signal NO in round n based on (4.2), an auxiliary signal YES′. We lower bound the expected utility for $\mathcal{S}$ by 0 when the signal YES′ is sent. This means that in every round $i \in [n]$, the expected utility for $\mathcal{S}$ conditional on a signal YES is $n \cdot \sum_{j=1}^{m} x_j \cdot \xi_j$ since $\sum_{j=1}^{m} x_j = 1/n$. This is exactly the objective function of (4.2), which upper bounds the expected utility of a persuasive signaling scheme in the offline case. Since $\sum_{j=1}^{m} x_j = 1/n$, a "regular" signal YES is sent with probability $1 - (1 - 1/n)^n \geq 1 - 1/e$ in one of the n rounds. □

4.1.2 Beyond IID

In the following, we move away from IID instances and discuss general independent distributions. Theorem 3.3 tells us that the offline scenario proves to be hard to solve. Interestingly, an optimal online scheme can be computed in polynomial time, where we use backwards induction and solve a linear program for each round $i = n-1, \ldots, 1$. Clearly, if no previous action has been recommended, a recommendation to take the action in the final round regardless of the type θ_n cannot decrease the expected utility for $\mathcal{R}$. The same holds for $\mathcal{S}$, hence, in an optimal persuasive scheme, the last action will be recommended if no action has previously been recommended. This means that the expected utility for $\mathcal{S}$ and $\mathcal{R}$ when reaching round n can be computed directly. Using the expected utility for round $i+1$, we can then compute the optimal recommendation probabilities for round i, for all $i = n-1, \ldots, 1$. The result is summarized in the following theorem.

Theorem 4.2
An optimal persuasive signaling scheme in the online setting can be computed in polynomial time.

Proof. We focus on direct signaling schemes. In such a scheme, at most a single signal YES will be sent in all n rounds. Hence, we can assume that the online process stops after the first YES signal. Either, $\mathcal{R}$ takes the current action, or $\mathcal{S}$ does not provide any additional information in the upcoming rounds.

Consider the case that round $i \in [n]$ is reached with $\sigma_1, \ldots, \sigma_{i-1} = \text{NO}$. At the beginning of round i, some type j is revealed. By x_{ij} we denote the probability that $\sigma_i = \text{YES}$ conditioned on that particular type. The optimal value for x_{ij} can be computed using the optimal choices for the subsequent rounds. To this end, we denote the expected utility for $\mathcal{R}$ and $\mathcal{S}$ from an optimal mechanism in rounds $i, i+1, \ldots, n$ by ϱ^i and ξ^i, respectively. Similarly, we use ϱ_E^i to denote the maximum expected utility for $\mathcal{R}$ from a single round, i.e., $\varrho_E^i = \max_{\ell \in \{i,\ldots,n\}} \sum_{j=1}^m q_{\ell j} \cdot \varrho_{\ell j}$.

Since $\varrho_{ij}, \xi_{ij} \geq 0$ for all $i \in [n], j \in [m]$, it is without loss of generality to set $x_{nj} = 1$ for all $j \in [m]$. Both $\mathcal{R}$ and $\mathcal{S}$ (weakly) prefer taking any action rather than no action. Hence, if the last round is reached without a previous recommendation for an action, a signal $\sigma_n = \text{YES}$ should be sent. Thus, $\varrho^n = \sum_{j=1}^m q_{nj} \cdot \varrho_{nj}$ and $\xi^n = \sum_{j=1}^m q_{nj} \cdot \xi_{nj}$.

Now, assume we know the optimal persuasive scheme for rounds $i+1, \ldots, n$. Then, in round i, some type j is revealed with probability q_{ij}. Hence, if the signal is YES, the expected utility for $\mathcal{S}$ is $q_{ij} \cdot \xi_{ij}$, and if it is NO, the expected utility for $\mathcal{S}$ is $q_{ij} \cdot \xi^{i+1}$ – as long as the resulting scheme is persuasive. Hence, $\mathcal{S}$ tries to maximize $\sum_{j=1}^m q_{ij} \cdot x_{ij} \cdot \xi_{ij} + \sum_{j=1}^m q_{ij} \cdot (1 - x_{ij}) \cdot \xi^{i+1}$ under persuasiveness constraints. Note that this is equal to

$$\xi^{i+1} + \sum_{j=1}^{m} q_{ij} \cdot x_{ij} \cdot \left(\xi_{ij} - \xi^{i+1}\right) \tag{4.3}$$

since $\sum_{j=1}^m q_{ij} = 1$.

The receiver's expected utility for taking an action after having received a YES signal is $\frac{\sum_{j=1}^m q_{ij} \cdot x_{ij} \cdot \varrho_{ij}}{\sum_{j=1}^m q_{ij} \cdot x_{ij}}$. If $\mathcal{R}$ dismisses the action in round i, $\mathcal{S}$ will not send additional information and hence $\mathcal{R}$ can only rely on the information available a priori. Thus, $\mathcal{R}$ would choose a subsequent round providing the a priori best expected utility which is ϱ_E^{i+1}. This means that $\mathcal{R}$ would follow a signal YES in round i if $\sum_{j=1}^m q_{ij} \cdot x_{ij} \cdot \varrho_{ij} \geq \sum_{j=1}^m q_{ij} \cdot x_{ij} \cdot \varrho_E^{i+1}$, or, equivalently,

$$\sum_{j=1}^{m} q_{ij} \cdot x_{ij} \cdot \left(\varrho_{ij} - \varrho_E^{i+1}\right) \geq 0 \ . \tag{4.4}$$

If, on the other hand, the signal $\sigma_i = \text{NO}$, the receiver gets an expected utility of $\frac{\sum_{j=1}^m q_{ij} \cdot (1-x_{ij}) \cdot \varrho_{ij}}{\sum_{j=1}^m q_{ij} \cdot (1-x_{ij})}$ when taking the action despite the signal. Otherwise, if following the recommendation, $\mathcal{R}$ gets an expected utility of $\frac{\sum_{j=1}^m q_{ij}\cdot(1-x_{ij})\cdot\varrho^{i+1}}{\sum_{j=1}^m q_{ij}\cdot(1-x_{ij})}$ by continuously following the recommendations given by the optimal persuasive mechanism. This means that $\mathcal{R}$ would dismiss the action in round i when receiving a signal $\sigma_i = \text{NO}$ if $\sum_{j=1}^m q_{ij}\cdot(1-x_{ij})\cdot\varrho^{i+1} \geq \sum_{j=1}^m q_{ij}\cdot(1-x_{ij})\cdot\varrho_{ij}$. Since $\sum_{j=1}^m q_{ij} = 1$ for all $i \in [n]$, this can be equivalently expressed as

$$\sum_{j=1}^m q_{ij}\cdot x_{ij}\cdot\left(\varrho_{ij} - \varrho^{i+1}\right) \geq \sum_{j=1}^m q_{ij}\cdot\varrho_{ij} - \varrho^{i+1} \ . \tag{4.5}$$

Given the values ϱ_E^{i+1}, ϱ^{i+1}, and ξ^{i+1} for some $i \in [n-1]$, the objective (4.3) as well as the constraints (4.4) and (4.5) are all linear functions. Hence, the optimal mechanism for round i can be determined using the following linear program, which, in turn, produces the values ϱ^i and ξ^i.

$$\begin{array}{lrcl} \text{Max.} & \xi^{i+1} + \displaystyle\sum_{j=1}^m q_{ij}\cdot x_{ij}\cdot(\xi_{ij} - \xi^{i+1}) & & \\ \text{s.t.} & \displaystyle\sum_{j=1}^m q_{ij}\cdot x_{ij}\cdot(\varrho_{ij} - \bar{\varrho}_E^{i+1}) & \geq & 0 \\ & \displaystyle\sum_{j=1}^m q_{ij}\cdot x_{ij}\cdot(\varrho_{ij} - \varrho^{i+1}) & \geq & \displaystyle\sum_j q_{ij}\cdot\varrho_{ij} - \varrho^{i+1} \\ & x_{ij} & \in & [0,1] \ \ \forall\, j \in [m] \ . \end{array} \tag{4.6}$$

Since ϱ^n and ξ^n are the a priori expectation for $\mathcal{R}$ and $\mathcal{S}$, respectively, and ϱ_E^i can be precomputed for any $i \in [n]$ just using the known distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$, we are able to find the optimal mechanism using backwards induction and a sequence of $n-1$ linear programs.

We have yet to show that the linear programs indeed produce a persuasive mechanism. By the inductive assumption, the mechanism obtained from the linear programs for rounds $i+1, \ldots, n$ is persuasive. Hence, $\varrho^{i+1} \geq \varrho_E^{i+1}$. Otherwise, $\mathcal{R}$ would certainly deviate to the most profitable round. This observation implies that (4.6) is feasible. If the a priori receiver expectation of round i is higher than the highest receiver expectation from the remaining rounds, i.e., $\sum_{j=1}^m q_{ij}\cdot\varrho_{ij} \geq \varrho_E^{i+1}$, then, setting $x_{ij} = 1$ for all $j \in [m]$ satisfies all constraints. This makes sense for $\mathcal{R}$, as this means that $\mathcal{R}$ a priori prefers round i to rounds $i+1, \ldots, n$.

Similarly, if the current round is a priori less profitable than a subsequent round, i.e., $\sum_{j=1}^m q_{ij}\cdot\varrho_{ij} < \varrho_E^{i+1}$, then $0 > \sum_j q_{ij}\cdot\varrho_{ij} - \varrho_E^{i+1} \geq \sum_j q_{ij}\cdot\varrho_{ij} - \varrho^{i+1}$, and setting $x_{ij} = 0$ for all $j \in [m]$ satisfies all constraints. Without additional information by $\mathcal{S}$, a rational $\mathcal{R}$ would wait for the a priori more profitable round. □

As we will show below, there are instances for which an online sender cannot obtain a positive approximation to an optimal prophet sender. The following theorem encapsulates this.

Theorem 4.3
There are instances in which no online signaling scheme yields a positive approximation ratio.

Before we begin with the proof, we give a short intuition why this holds. Since the sender's scheme is public knowledge, $\mathcal{R}$ knows whether $\mathcal{S}$ uses information from future rounds for the signal in the current round i. An offline or prophet $\mathcal{S}$ is able to use the complete state of nature $\theta_1, \dots, \theta_n$, whereas an online $\mathcal{S}$ is restricted to the first i actions and their types $\theta_1, \dots, \theta_i$. Hence, an offline sender can credibly convince the receiver to wait for an action with a good type in a round with an a priori low expectation whereas in the online case, a save option in an early round is always preferred by $\mathcal{R}$.

Proof of Theorem 4.3. Consider the following instance with $n = 2$ actions. The first action has a deterministic type θ_1 with $\varrho_1 = 1, \xi_1 = 0$. The second action has two types, each is drawn with probability $1/2$. Type θ_{21} has values $\varrho_{21} = 2 - 2\varepsilon$ for some $\varepsilon \in (0, 1/2)$ and $\xi_{21} = 1$. The other type has values $\varrho_{22} = \xi_{22} = 0$.

Action 1 yields an a priori expectation of 1 for $\mathcal{R}$, whereas action 2 only provides an a priori expectation of $1-\varepsilon < 1$. Hence, an online sender cannot persuade $\mathcal{R}$ to wait for the second round. Thus, for any persuasive online scheme φ_{on}, we have $u_{\mathcal{S}}(\varphi_{on}) = 0$.

An offline sender on the other hand knows which type the second action has. Since θ_{21} provides the highest profit to both $\mathcal{S}$ and $\mathcal{R}$, an optimal offline scheme φ^*_{off} recommends action 2 if and only if θ_{21} is realized and action 1 otherwise. Thus, $\mathcal{R}$ takes the best action in both cases and the scheme is persuasive. The sender gets an expected utility of $u_{\mathcal{S}}(\varphi^*_{off}) = 1/2$. This shows that no persuasive online scheme can achieve a positive approximation ratio. $\square_{\text{Theorem 4.3}}$

To contrast this negative result, we identify instances for which $\mathcal{S}$ can extract a constant approximation of the optimal offline signaling scheme in the following section.

4.1.3 Satisfactory Status Quo

In this section, we describe our assumption which we term “Satisfactory Status Quo” or SSQ for short. Under this assumption, we are able to design a persuasive online scheme which guarantees a constant approximation of the optimal offline scheme.

The motivation behind SSQ is that there exists a canonical deviation option for $\mathcal{R}$, i.e., the status quo which is satisfactory. Consider the example from the introduction of a customer shopping for a car. Whenever the customer has a vehicle that is still working, clearly, the new one should be a significant upgrade. Surely, the customer will only buy a new car if it is an improvement over the old one justifying a substantial financial investment. More formally, the assumption requires two conditions.

1. There exists an external option $X \notin [n]$ which has the best expectation for $\mathcal{R}$, i.e., $\varrho_E = \sum_{j=1}^m q_{Xj} \cdot \varrho_{Xj} \geq \max_{i\in[n]} \sum_{j=1}^m q_{ij} \cdot \varrho_{ij}$ that can be chosen by $\mathcal{R}$ at any time in the online process.

2. The instance is ϱ_E-optimal (cf. Section 3.2).

The first condition can equivalently be replaced by the condition that action n in the final round has the highest a priori expectation for $\mathcal{R}$. Clearly, if n has the highest a priori expectation for $\mathcal{R}$, waiting for the final round and dismissing the actions in other rounds is a viable strategy for $\mathcal{R}$. Similarly, if there exists an external option $X \notin [n]$, relabeling it as action $n + 1$ and thereby making it the final action satisfies the condition. In the following, we will assume that the there is an external option $X \notin [n]$ to simplify the exposition.

We have already seen how to get an upper bound on the sender's expected utility in the offline case for ϱ_E-optimal instances with n signals through (3.5), i.e.,

Algorithm 7: Simple Scheme for SSQ

Input: Distributions $(\mathcal{D}_i)_{i\in[n]}$, factors $\boldsymbol{d} = (d_i)_{i\in[n]}$, online sequence of types $\theta_1, \dots, \theta_n$, outside option X

Compute an optimal solution $\boldsymbol{x}^*$ for (4.7).
Set $recSent = False$.
for *round* $i = 1, \dots, n$ **do**
  Observe type θ_i of the action in round i.
  if $recSent = True$ **then** Signal NO.
  else
    Draw $x \sim$ Unif[0,1].
    if $x \leq 1 - d_i$ **then** Signal NO.
    else
      Draw $x \sim$ Unif[0,1].
      if $x \leq \frac{x^*_{\theta_i}}{q_{\theta_i}}$ **then** Send signal YES and set $recSent = True$.
      else Send signal NO.

$f([n-1]) \geq u_{\mathcal{S}}\left(\varphi^*_{[n]}\right)$. The corresponding linear program is a natural extension of the linear program (4.2) for the IID case and directly corresponds to (3.4) and (3.3) for $S = [n-1]$.

$$
\begin{array}{lrcll}
\text{Max.} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot \xi_{ij} & & & \\
\text{s.t.} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} & \leq & 1 & \\
 & \displaystyle\sum_{j=1}^{m} x_{ij} \cdot \varrho_{ij} & \geq & \varrho_E \cdot \displaystyle\sum_{j=1}^{m} x_{ij} & \forall\, i \in [n] \\
 & x_{ij} & \in & [0, q_{ij}] & \forall\, i \in [n], j \in [m]
\end{array}
\tag{4.7}
$$

Similar to the simple scheme for IID, we define our *simple scheme for SSQ* based on an optimal solution $\boldsymbol{x}^*$ of linear program (4.7). This allows us to prove the main result of this section, namely a tight $1/2$-approximation for instances satisfying SSQ. In round i of the mechanism, with probability $1-d_i$, signal $\sigma_i = \text{NO}$ is sent without considering the type of action i. This ensures that earlier rounds do not get a disproportionate amount of recommendations just because they are early. Note that for the similar mechanism ComputeSignal (Algorithm 4), we were able to choose the order in which actions are considered. In this online version on the other hand, the order is predetermined. Due to these damping factors $\boldsymbol{d}$, it can very well happen that no signal YES is sent during the online process. This can be interpreted as a recommendation to stick with the status quo or take the external action.

Theorem 4.4
For a suitable choice of parameters $\mathbf{d}$*, the simple scheme for SSQ is persuasive in the online setting satisfying SSQ and yields a* $1/2$*-approximation. Further, there are instances satisfying SSQ for which no better approximation is possible.*

Proof. We begin by showing that for every $\boldsymbol{d} \in [0,1]^n$, the resulting scheme is persuasive. Consider some round $i \in [n]$. With probability $d_i \cdot \frac{x_{ij}}{q_{ij}}$, a signal $\sigma_i = \text{YES}$ is sent for type j. First, assume that $\sigma_k = \text{NO}$ for all rounds $k = 1, \ldots, i-1$ and $\sigma_i = \text{YES}$. Following that recommendation and taking the action yields an expected utility for $\mathcal{R}$ of

$$\frac{\sum_{j=1}^{m} q_{ij} \cdot d_i \cdot \frac{x_{ij}}{q_{ij}} \cdot \varrho_{ij}}{\sum_{j=1}^{m} q_{ij} \cdot d_i \cdot \frac{x_{ij}}{q_{ij}}} = \frac{d_i \cdot \sum_{j=1}^{m} x_{ij} \cdot \varrho_{ij}}{d_i \cdot \sum_{j=1}^{m} x_{ij}} \geq \varrho_E \quad ,$$

where the inequality follows from the second constraint of (4.7). Not taking the action in round i means that $\mathcal{S}$ will not reveal any more information. Hence, the expectation for $\mathcal{R}$ cannot exceed ϱ_E, the value which is guaranteed by the outside option, and, by assumption, upper bounds the expectation of the current action. Thus, it is in the receiver's interest to take the action in round i on a YES-signal.

If $\sigma_k = \text{NO}$ for $k = 1, \ldots, i$, the receiver's expected utility for taking the action regardless is

$$\begin{aligned} \frac{\sum_{j=1}^{m} q_{ij} \cdot \left(1 - \frac{x_{ij} \cdot d_i}{q_{ij}}\right) \cdot \varrho_{ij}}{\sum_{j=1}^{m} q_{ij} \cdot \left(1 - \frac{x_{ij} \cdot d_i}{q_{ij}}\right)} &= \frac{\sum_{j=1}^{m} (q_{ij} - x_{ij} \cdot d_i) \cdot \varrho_{ij}}{\sum_{j=1}^{m} q_{ij} - x_{ij} \cdot d_i} = \frac{\sum_{j=1}^{m} q_{ij} \cdot \varrho_{ij} - d_i \cdot \sum_{j=1}^{m} x_{ij} \cdot \varrho_{ij}}{1 - d_i \cdot \sum_{j=1}^{m} x_{ij}} \\ &\leq \frac{\varrho_E - d_i \cdot \varrho_E \cdot \sum_{j=1}^{m} x_{ij}}{1 - d_i \cdot \sum_{j=1}^{m} x_{ij}} = \varrho_E \quad . \end{aligned}$$

The inequality comes from the fact that $\sum_{j=1}^{m} q_{ij} \cdot \varrho_{ij} \leq \varrho_E$ due to our assumption SSQ and the second constraint of (4.7). Since $\mathcal{R}$ can guarantee an expected utility of at least ϱ_E by following the signal – either because there is a signal YES in a later round $i' > i$ or by taking the outside option – it is in the receiver's interest to follow the recommendation of the scheme. Combining these two cases shows persuasiveness of the scheme for any $\boldsymbol{d} \in [0,1]^n$.

To show that the simple scheme for SSQ guarantees an approximation ratio of at least $1/2$ compared to the expected utility of an optimal offline scheme, it suffices to show that the simple scheme provides an expected utility of at least $1/2$ of the optimal value of the linear program (4.7).

To this end, we define *damping factors* d_i such that for each round, the probability for the scheme to consider sending a signal is exactly $1/2$. We follow an approach by Chawla et al. [28] as well as Alaei [2]. We define $r_i = \Pr[\text{reaching round } i]$. Clearly, $r_1 = 1$. For $i \geq 1$, the following recursion holds

$$r_{i+1} = r_i \cdot \left(1 - d_i \cdot \sum_{j=1}^{m} x_{ij}\right) \quad ,$$

since the probability to reach round $i+1$ is exactly the probability to reach round i and not send a signal $\sigma_i = \text{YES}$. Setting $d_i = \frac{1}{2 \cdot r_i}$ guarantees $r_i \cdot d_i = 1/2$ as required.

Note that d_i is well defined as $r_i \geq 1/2$ for all i. It holds that

$$\begin{aligned} r_{i+1} &= r_i \cdot (1 - d_i \cdot \sum_{j=1}^{m} x_{ij}) = r_i - \frac{1}{2} \cdot \sum_{j=1}^{m} x_{ij} \\ &= r_{i-1} - \frac{1}{2} \cdot \left(\sum_{j=1}^{m} x_{(i-1)j} + \sum_{j=1}^{m} x_{ij} \right) \\ &= \dots \\ &= r_1 - \frac{1}{2} \cdot \sum_{\ell=1}^{i} \sum_{j=1}^{m} x_{\ell j} \end{aligned}$$

and $\sum_{i,j} x_{ij} \leq 1$ due to the first constraint of (4.7).

This means that the expected utility for $\mathcal{S}$ is

$$\sum_{i=1}^{n} r_i \cdot d_i \cdot \sum_{j=1}^{m} q_{ij} \cdot \frac{x^*_{ij}}{q_{ij}} \cdot \xi_{ij} = \sum_{i=1}^{n} \frac{1}{2} \cdot \sum_{j=1}^{m} x^*_{ij} \cdot \xi_{ij} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} x^*_{ij} \cdot \xi_{ij} \ ,$$

i.e., exactly $1/2$ times the optimal value of (4.7).

To complete the proof, we now give a sequence of instances where an online scheme cannot guarantee more than a factor of $1/2$. There are $n = 2$ rounds, where $\mathcal{D}_1$ is a deterministic distribution with $\varrho_1 = \xi_1 = 1$. $\mathcal{D}_2$ has two types, θ_{21} and θ_{22}. Type θ_{21} is drawn with probability $1/t$ and provides utility $\varrho_{21} = \xi_{21} = t$. The other type, θ_{22}, which is drawn with probability $1 - 1/t$, does not provide any utility, i.e., $\varrho_{22} = \xi_{22} = 0$. Additionally, there is an outside option X with utility $\varrho_X = 1$ for $\mathcal{R}$ and utility $\xi_X = 0$ for $\mathcal{S}$.

An optimal offline scheme clearly recommends action 2 with type θ_{21} if it is realized and action 1 otherwise. This gives $\mathcal{S}$ an expected utility of $2 - 1/t$. A persuasive online scheme can only guarantee an expected utility of 1 to $\mathcal{S}$ as there is no information on the type of action 2 in round 1. □

Interestingly, in the classic one-dimensional prophet inequality problem, this bound of $1/2$ is the best possible for general distributions. Note that the above instance represents a worst-case instance for the one-dimensional problem when disregarding the outside option – the interests of $\mathcal{S}$ and $\mathcal{R}$ are perfectly aligned, hence, it is essentially a one-dimensional instance.

Crucially, the benchmarks for Bayesian persuasion instances and prophet inequality instances differ. For more general instances satisfying SSQ with misaligned interests, $\mathcal{S}$ will only achieve a $1/2$-approximation to the optimal offline $\mathcal{S}$, whereas in the one-dimensional problem, the gambler achieves a $1/2$-approximation to the expected maximum. These values can be very different.

This observation concludes our discussion of prophet inequalities for Bayesian persuasion. In the next section, we continue in the realm of online persuasion but consider a different model which is more in line with the classic secretary problem.

4.2 Secretary Recommendation

The online model nicely represents dynamic arrival and departure of choice options, yet it is not always the case that distributional information on the actions' types is available.

In this section, which is based on [48], we study a variant of online Bayesian persuasion where there are no known underlying distributions from which the actions' types are drawn. Rather, the only prior knowledge $\mathcal{S}$ and $\mathcal{R}$ have is that the actions and their types are revealed in a uniform random order. The process is very similar to the one in the previous section.

1. $\mathcal{S}$ commits to a signaling scheme φ.
2. $\mathcal{R}$ learns the scheme φ.
3. In each round $i = 1, \dots, n$:
   - 3.1. $\mathcal{S}$ learns the type θ_i of action i.
   - 3.2. $\mathcal{S}$ sends a signal $\sigma_i \in \Sigma$ according to φ to $\mathcal{R}$.
   - 3.3. $\mathcal{R}$ decides whether to take action i based on signals $\sigma_1, \dots, \sigma_i$. If $\mathcal{R}$ takes action i, the process ends.

The model setup is reminiscent of the d-random order scenario for $d = 1$ in the previous chapter in that there exists a vector $\boldsymbol{\theta}$ of types and the types are permuted uniformly at random. Crucially, that vector is unknown to $\mathcal{R}$ and $\mathcal{S}$. This leads to an interesting question regarding the model:

Without knowing what is to come, how can $\mathcal{R}$ evaluate the signaling scheme φ employed by $\mathcal{S}$ when the only information is a YES or a NO in every round? To solve this, we require the sender's schemes to be persuasive even if $\mathcal{R}$ were to know the type-vector a priori. Using this approach, we can assume without loss of generality that φ is a direct and persuasive scheme via the same arguments we used in the previous Section 4.1.

For a simpler exposition, we denote by $\boldsymbol{\varrho}$ and $\boldsymbol{\xi}$ the utility values corresponding to a type vector $\boldsymbol{\theta}$. Note that for all $t \in [n]$, ϱ_t and ξ_t directly correspond to type θ_t and a single type is realized for each action. Further, we denote by $\varrho_{max} = \max_{t \in [n]} \varrho_t$ and $\xi_{max} = \max_{t \in [n]} \xi_t$ the respective highest utility values and by $c_{\mathcal{R}} = \arg\max_{\theta \in \boldsymbol{\theta}} \varrho(\theta)$ and $c_{\mathcal{S}} = \arg\max_{\theta \in \boldsymbol{\theta}} \xi(\theta)$ the types corresponding to these values. For simplicity, we assume without loss of generality that the utility values are mutually distinct, which means that there is a unique type $c_{\mathcal{R}}$ and a potentially different but also unique type $c_{\mathcal{S}}$. Using a slight perturbation of equal values, this assumption can be satisfied without (significantly) changing the instances.

To study the effects of informational advantage of $\mathcal{S}$ over $\mathcal{R}$, we consider two different variants. The first scenario is *without disclosure*, which means that $\mathcal{R}$ only learns the signals $\sigma_1, \dots, \sigma_i$ before making a decision in round i. The second scenario is *with disclosure*, where $\mathcal{R}$ is informed about the types of dismissed actions. Hence, at the beginning of round i, the sender's informational advantage is only the knowledge of θ_i as $\mathcal{R}$ has learned $\theta_1, \dots, \theta_{i-1}$.

Further, we consider two variants of objectives for both $\mathcal{S}$ and $\mathcal{R}$. For the *ordinal* objective, $\mathcal{S}$ (or $\mathcal{R}$) aims to maximize the *success probability*, i.e., the probability that the action which is eventually taken has type $c_{\mathcal{S}}$ (or $c_{\mathcal{R}}$, respectively). Not securing their respective best type is considered as failure and does not provide any utility. For the second objective, the *cardinal* objective, the respective agent strives to maximize the expected utility of the action which is taken. Hence, in this case, not being able

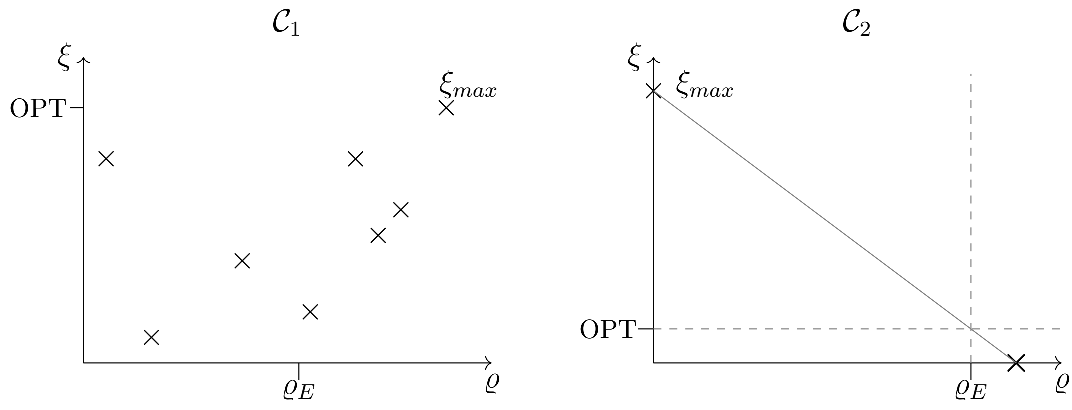


Figure 10: Dependence of OPT on the set of types

to secure the individual best type can still provide a good utility. As a benchmark, we consider the *basic scenario* in which $\boldsymbol{\theta}$ is known to both $\mathcal{S}$ and $\mathcal{R}$, while the order of the types remains unknown. In the *secretary scenario*, the types are a priori unknown.

We study all sixteen different combinations of these variants, i.e., cardinal and ordinal $\mathcal{R}$ and $\mathcal{S}$ both with and without disclosure for the basic as well as the secretary scenario. We summarize the approximation guarantees of our signaling schemes in Tables 3 and 4 below. All results are without lower-order terms and asymptotics are with respect to n, the number of actions. Results in bold represent matching upper and lower bounds.

In Table 3, we show the results for the *cardinal receiver objective.* The entries in the upper left, i.e., for the basic scenario without disclosure, represent the benchmark for the remaining entries. We use the so-called *Pareto Procedure* which uses the geometric ideas of Algorithm 1 to determine a type to recommend for any set of types. This allows us to use the idea of our Basic Pareto Mechanism in an online fashion. The resulting mechanism is optimal for $\mathcal{S}$ and persuasive. Hence, we can use its objective value (expected utility and success probability, respectively) as benchmark for the other scenarios. We denote the objective value by OPT.

Note that there is no concise closed-form expression for the sender's expected utility with respect to ξ_{max}, it depends on the complete set of types $\mathcal{C}$. In Figure 10, we give a short illustration of this phenomenon. In both type sets, ξ_{max} has the same value. For $\mathcal{C}_1$ on the left side, $c_{\mathcal{S}} = c_{\mathcal{R}}$, hence, in an optimal mechanism for the basic scenario, waiting for and recommending $c_{\mathcal{S}}$ is a persuasive strategy. For $\mathcal{C}_2$, on the other hand, every type besides $c_{\mathcal{S}}$ is clustered around $c_{\mathcal{R}}$, denoted by the bold marker. Since a persuasive mechanism must guarantee a utility of at least ϱ_E to $\mathcal{R}$, OPT cannot be more than $1/n \cdot \xi_{max}$.

In the left column of Table 3, the lower entries represent the secretary scenario without disclosure. Clearly, this setting represents a generalization of the classic secretary problem and hence, an upper bound of $1/e$ for perfectly aligned $\mathcal{S}$ and $\mathcal{R}$ utility values follows directly. Interestingly, we are able to repeatedly run the Pareto Procedure on the growing set of types revealed up to and including round i to achieve a constant-factor approximation of the optimal value in the corresponding benchmark scenario. The lower bounds for ordinal and cardinal sender utility are both obtained using this mechanism.

For the basic scenario with disclosure, we first describe an optimal mechanism using

Cardinal $\mathcal{R}$	without Disclosure		with Disclosure	
	Ordinal $\mathcal{S}$	Cardinal $\mathcal{S}$	Ordinal $\mathcal{S}$	Cardinal $\mathcal{S}$
Basic	Optimal mechanism Cor. 4.5		$\mathbf{\frac{1}{2}}$ Thm. 4.16, 4.19	$\frac{1}{3}$ Thm. 4.18
Secretary	$\frac{1}{4}$ Thm. 4.11	$\frac{1}{3\sqrt{3}}$ Thm. 4.8	$\mathbf{\Theta\left(\frac{1}{n}\right)}$ Thm. 4.20	$\mathbf{\Theta\left(\frac{1}{n}\right)}$ Cor. 4.21

Table 3: Approximation guarantees of persuasive mechanisms for cardinal receiver utility. All bounds stated without lower-order terms. Results indicated in bold have asymptotically matching upper bounds.

an exponentially-sized family of nested linear programs. Additionally, we are able to repeatedly use the Pareto Procedure on the shrinking set of remaining types. The resulting mechanism requires only polynomial time and guarantees a constant-factor approximation regardless of the sender's objective. For the ordinal sender objective, we show a matching upper bound of $1/2$.

The final setting with cardinal receiver utility, namely the secretary scenario with disclosure, proves to be the only setting without a constant-factor approximation – even including the case of ordinal receiver utility described below. While a trivial mechanism of recommending action 1 deterministically is persuasive and provides a success probability of $1/n$, there are instances for which no persuasive mechanism can improve upon this trivial lower bound. For cardinal sender utility, the same tight approximation factor of $\Theta(1/n)$ holds.

In Table 4, we summarize the results for the settings with *ordinal receiver objective*. Notably, for all scenarios, the approximation guarantees of our mechanisms have small constant factors and asymptotically matching upper bounds. In the basic scenario, $\mathcal{S}$ leverages the following two facts. First, $\mathcal{R}$ only has a success probability of $1/n$ without any additional information, and second, $\mathcal{R}$ only cares about securing $c_{\mathcal{R}}$ in the ordinal setting. Hence, $\mathcal{S}$ is able to secure $c_{\mathcal{S}}$ with a probability of $1 - 1/n$. This holds even if the types of dismissed actions are disclosed to $\mathcal{R}$.

In the secretary scenario without disclosure, our mechanism flips a weighted coin and then runs either the classic secretary algorithm for $\mathcal{S}$, disregarding the types' values for $\mathcal{R}$ or the other way around, i.e., only considering the values for $\mathcal{R}$ and not those for $\mathcal{S}$. Optimizing the probabilities with which the respective algorithms are run gives a tight lower bound of $1/e$. For the secretary scenario with disclosure, this approach is not feasible. Assume some action i gets a NO-recommendation and is subsequently dismissed by $\mathcal{R}$. If round i is after the initial sample phase and action i has a higher utility for $\mathcal{R}$ than the previous actions, it would reveal that the mechanism which is being used optimizes for $\mathcal{S}$. This in turn might incentivize $\mathcal{R}$ to ignore any upcoming YES-signals. Instead, our mechanism proposes the first action which has a higher utility for $\mathcal{S}$ or for $\mathcal{R}$ than all previous ones. This leads to a tight approximation ratio of $1/4$. Interestingly, this scenario allows $\mathcal{R}$ to enjoy a success probability of $1/4$ as well, a massive improvement compared to $1/n$ without the involvement of $\mathcal{S}$.

Ordinal $\mathcal{R}$	without Disclosure		with Disclosure	
	Ordinal $\mathcal{S}$	Cardinal $\mathcal{S}$	Ordinal $\mathcal{S}$	Cardinal $\mathcal{S}$
Basic	$\mathbf{1}$ Prop. 4.23	$\mathbf{1}$ Prop. 4.23	$\mathbf{1}$ Prop. 4.26	$\mathbf{1}$ Prop. 4.26
Secretary	$\mathbf{\frac{1}{e}}$ Thm. 4.24	$\mathbf{\frac{1}{e}}$ Thm. 4.24	$\mathbf{\frac{1}{4}}$ Thm. 4.28, 4.29	$\mathbf{\frac{1}{4}}$ Thm. 4.28, 4.29

Table 4: Approximation guarantees of persuasive mechanisms for ordinal receiver utility discussed. All bounds stated without lower-order terms. Results indicated in bold have asymptotically matching upper bounds.

We start our discussion with the case of cardinal receiver utility.

4.2.1 Cardinal Utility for $\mathcal{R}$

Throughout this section, we consider the case of cardinal utility for $\mathcal{R}$ as summarized in Table 3 above. When introducing Bayesian persuasion in Chapter 3, we showed that the Basic Pareto Mechanism (Algorithm 1) is an optimal persuasive mechanism for random-order instances in the offline setting. Recall that the algorithm identified two types a and b on the Pareto frontier of $\boldsymbol{\theta}$ and proceeded to send a recommendation for either a or b with a certain probability, satisfying persuasiveness-constraints.

With this in mind, we start the discussion of our results with the basic scenario without disclosure of dismissed actions.

Benchmark: Basic Scenario without Disclosure

In the basic scenario without disclosure of dismissed actions, $\mathcal{S}$ essentially faces an offline random order problem. This is due to the following. We assume that the set of types $\mathcal{C}$ is public knowledge, only the order of the types is unknown. Hence, $\mathcal{S}$ is able to determine the type to recommend prior to the start of the process. Then, $\mathcal{S}$ only needs to identify the position of the chosen type. Clearly, this can be done in an online fashion. Since $\mathcal{R}$ does not learn the types of dismissed actions, the distribution behind a YES-signal is the same no matter in which round i it is sent. The same holds for the distribution behind the NO-signals in the other rounds. We use our *Pareto Procedure* (Algorithm 8) which computes a type to recommend based on the set of valuation pairs for all our constant-factor approximation algorithms. Hence, it requires a few additions compared to the Basic Pareto Mechanism in Chapter 3.

As a first step, the Pareto Procedure adjusts the input. For all $t \in [n]$, the procedure scales $\xi_t = \frac{\xi_t}{\xi_{max}}$ such that all values for $\mathcal{S}$ are between 0 and 1. Similarly, the new receiver values are $\varrho_t = \frac{\varrho_t}{\varrho_{max}}$ for all $t \in [n]$. Note that $\xi_{max} = 0$ or $\varrho_{max} = 0$ can only occur if $n = 1$ since we assumed mutually distinct values. Thus, besides this corner case, in which the only persuasive mechanism is to signal YES for the single action, this adjustment step is well-defined.

For ordinal sender utility, an additional normalizing step is performed. For every type $t \neq c_{\mathcal{S}}$, the procedure sets $\xi_t = 0$. This way, the expected utility for $\mathcal{S}$ is equal to the success probability in such settings. This clearly transforms the Pareto frontier – only $c_{\mathcal{S}}$ and $c_{\mathcal{R}}$ remain on the Pareto frontier. Since the values for $\mathcal{R}$ remain the same, this does not change the outcome for $\mathcal{R}$ from the Pareto Procedure. In any case, a utility of at least ϱ_E for $\mathcal{R}$ is guaranteed and an expectation strictly higher than ϱ_E is achieved if and only if $\varrho_{c_{\mathcal{S}}} > \varrho_E$. Note that we only apply the Pareto Procedure to the case of cardinal receiver objective and hence such a step for the receiver values is not performed. The mechanisms for the setting with disclosure or the secretary scenario utilize the procedure for a shrinking or growing subset of types, respectively. For each of these calls to the procedure, the adjustments are performed for the current set of types.

The Pareto Procedure then computes a single type c such that $\mathbb{E}[\varrho(c)] \geq \varrho_E$ while maximizing the expectation for $\mathcal{S}$ under that constraint as we have seen in the Basic Pareto Mechanism. The Online Pareto Mechanism (Algorithm 9) considers the input sequence and sends a recommendation YES exactly when the current type is c, where c is the type computed by the Pareto Procedure. This way, the Online Pareto Mechanism functions as an online version of the Basic Pareto Mechanism. Hence, our first result is a corollary of Proposition 3.1.

Corollary 4.5
For cardinal and ordinal sender and cardinal receiver utility, the Online Pareto Mechanism is an optimal persuasive mechanism in the basic scenario without disclosure.

Since both the Basic Pareto Mechanism and the Online Pareto Mechanism essentially function in the same way, i.e., identifying a type c by using the Pareto Procedure and recommending only this type, we will drop the “Basic” or “Online” and refer to the mechanism as the Pareto Mechanism.

Hence, using the Pareto Mechanism gives us our benchmark for the remaining scenarios. We continue with the discussion of the secretary scenario without disclosure, i.e., the bottom left entry of Table 3.

Secretary Scenario without Disclosure

In this scenario, neither $\mathcal{S}$ nor $\mathcal{R}$ know the possible types a priori and dismissed types are not revealed to $\mathcal{R}$. We use the following approach. In rounds $1, \ldots, s$, our mechanism $\varphi(s)$ merely observes the types and only sends signals $\sigma_1, \ldots, \sigma_s = \text{NO}$. We denote the set of types observed up to round i by A_i. In rounds $i = s+1, \ldots, n-1$, $\varphi(s)$ updates the set A_i with the current type θ_i and calls the Pareto Procedure with input A_i. If the type computed by the Pareto Procedure is equal to θ_i, a signal $\sigma_i = \text{YES}$ is sent, otherwise, $\sigma_i = \text{NO}$ is issued. Once the first recommendation is given, only NO-signals will be sent. In the final round n, unless a YES-recommendation has been sent in a previous round, the current action is recommended regardless of its type θ_n. Since a growing set of types is considered in each round, we term our mechanism the *Growing Pareto Mechanism* (Algorithm 10).

With Lemma 4.7, we show that the Growing Pareto Mechanism is persuasive before showing the approximation guarantees of $\frac{1}{3\sqrt{3}} - o(1)$ and $1/4 - o(1)$ for cardinal and ordinal sender utility in Theorem 4.8 and 4.11, respectively.

Algorithm 8: Pareto Procedure

Input: A set of valuation pairs $(\varrho_t, \xi_t)_{t\in[n]}$
Output: A type c

Set $c_{\mathcal{S}} = \arg\max_{t\in[n]} \xi_t$.
Scale and normalize the type values, set $\varrho_E = {}^1\!/_n \cdot \sum_{t=1}^{n} \varrho_t$.
Let $\mathcal{C} = \{(\varrho_t, \xi_t) \mid t \in [n]\}$ and $conv(\mathcal{C})$ be the convex hull of $\mathcal{C}$.
Let $PC(\mathcal{C})$ be the Pareto frontier of $conv(\mathcal{C})$.
if $\varrho_{c_{\mathcal{S}}} \geq \varrho_E$ **then** Set $a = b = c_{\mathcal{S}}$.
else
  Find types $a, b \in [n]$ s.t. (a, b) is the segment of $PC(\mathcal{C})$ that intersects with line $\varrho = \varrho_E$.
Determine probability for type a:
if $\varrho_a = \varrho_b$ **then** Set $\alpha = 1$.
else if $\varrho_a \neq \varrho_b$ *and* $\xi_a = \xi_b$ **then** Set $\alpha = 0$.
else Set $\alpha = \frac{\varrho_E - \varrho_b}{\varrho_a - \varrho_b}$.
Draw $x \sim \mathrm{Unif}[0, 1]$.
if $x \leq \alpha$ **then** Set $c = a$.
else Set $c = b$.
return *Type* c

Algorithm 9: Online Pareto Mechanism

Input: Set of valuation pairs $(\varrho_t, \xi_t)_{t\in[n]}$, online sequence of types $\theta_1, \dots, \theta_n$

Let c denote the output of the Pareto Procedure for $(\varrho_t, \xi_t)_{t\in[n]}$.
for *round* $i = 1, \dots, n$ **do**
  if $\theta_i = c$ **then** Send $\sigma_i =$ YES.
  else Send $\sigma_i =$ NO.

To show persuasiveness, we consider round i and prove that it is in the interest of $\mathcal{R}$ to follow the signal given by the mechanism. Following the ideas of [57], we adopt a different perspective on the process of drawing the type in round i. Rather than iteratively drawing a type uniformly at random from the remaining set of types, we first draw the set A_i of i types uniformly at random. Then, the type to be observed in round i is drawn uniformly at random from the set A_i. This allows us to show the following lemma which we use to prove Lemma 4.7.

Lemma 4.6
Consider a given round i and a given subset of types A_i that arrived up to round i. In the Growing Pareto Mechanism

$$\Pr\left[\bigwedge_{\ell=1}^{i-1} \sigma_\ell = NO \;\middle|\; A_{i-1}\right] = \begin{cases} 1 & i = 2, \dots, s+1 \\ \frac{s}{i-1} & i = s+2, \dots, n \end{cases}$$

and

Algorithm 10: Growing Pareto Mechanism

Input: Number of rounds n, sample size s, online sequence of types $\theta_1, \ldots, \theta_n$

Set $A_0 = \emptyset$ and $recSent = False$.

for *round* $i = 1, \ldots, n-1$ **do**

  Set $A_i = A_{i-1} \cup \{\theta_i\}$. `// Observe type of action` i.

  if $i \leq s$ *or* $recSent = True$ **then** Signal NO.

  else

    Let c_i be the type chosen by Pareto Procedure on set A_i.

    if $c_i = \theta_i$ **then** Signal YES and set $recSent = True$.

    else Signal NO.

for *round* n **do**

  if $recSent = False$ **then** Signal YES.

  else Signal NO.

$$\Pr[\sigma_i = \mathit{YES} \mid A_i] = \begin{cases} 0 & i = 1, \ldots, s \\ \frac{1}{i} \cdot \frac{s}{i-1} & t = s+1, \ldots, n-1 \\ \frac{s}{n-1} & i = n \end{cases} .$$

Proof. Given a set A_i of types observed in rounds $1, \ldots, i$, we can draw the order of the types in a reverse fashion. We denote the type observed in round i by t_i. For each type $t \in A_i$, the probability to be in position i is $1/i$. The Pareto Procedure computes a single type $c \in A_i$ for input set A_i. Hence, the probability that $c = t_i$ is $1/i$. However, the mechanism never sends more than a single YES-signal. Thus, in all previous rounds $\ell = 1, \ldots, i-1$, the signal $\sigma_\ell =$ NO had to be sent for a signal $\sigma_i =$ YES.

Hence, consider the signal in round $i-1$ for given set A_i with $t_i \in A_i$ observed in round i. For the set $A_{i-1} = A_i \setminus \{t_i\}$, the Pareto Procedure computed some type $c \in A_{i-1}$. Again, the type in round $i-1$ is chosen uniformly at random from A_{i-1}, hence, with probability $1 - \frac{1}{i-1} = \frac{i-2}{i-1}$, the signal σ_{i-1} in round $i-1$ was NO. We continue the same argumentation for rounds $\ell \in \{i-2, i-3, \ldots, s+1\}$, where in each round ℓ the probability for a signal NO is $\frac{\ell-1}{\ell}$. Since the Growing Pareto Mechanism always sends a signal NO in the first s rounds, we get

$$\Pr\left[\bigwedge_{\ell=1}^{i-1} \sigma_\ell = \mathrm{NO} \,\middle|\, A_{i-1}\right] = \prod_{\ell=s+1}^{i-1} \frac{\ell-1}{\ell} = \frac{s}{s+1} \cdot \frac{s+1}{s+2} \cdots \frac{i-3}{i-2} \cdot \frac{i-2}{i-1} = \frac{s}{i-1}$$

for every round $i = s+1, \ldots, n$. Thus, for rounds $i = s+1, \ldots, n-1$, we get

$$\Pr[\sigma_i = \mathrm{YES} \mid A_i] = \mathbb{E}_{t_i}\left[\Pr[\sigma_i = \mathrm{YES} \mid A_i, t_i] \cdot \Pr\left[\bigwedge_{\ell=1}^{i-1} \sigma_\ell = \mathrm{NO} \,\middle|\, A_i \setminus \{t_i\}\right]\right] = \frac{1}{i} \cdot \frac{s}{i-1} \; .$$

Since $\sigma_n =$ YES with probability 1 whenever $\sigma_1, \ldots, \sigma_{n-1} =$ NO, for round n we get

$$\Pr[\sigma_n = \mathrm{YES} \mid A_n] = \Pr[\sigma_n = \mathrm{YES}] = 1 \cdot \frac{s}{n-1}$$

using an analogous argumentation, concluding the proof of the lemma. □

Lemma 4.7
For cardinal receiver utility, the Growing Pareto Mechanism is persuasive in the secretary scenario without disclosure.

Proof. Consider some round $i \in \{s+1, \dots, n\}$ with signal $\sigma_i = \text{YES}$. This implies that $\sigma_\ell = \text{NO}$ for all previous rounds $\ell \in [i-1]$. Observe that A_i, the set of types revealed up to and including round i is chosen uniformly at random and for every type $t \in A_i$ observed in round i, the probability that the Growing Pareto Mechanism only sends signals $\sigma_1, \dots, \sigma_{i-1} = \text{NO}$ is the same and does not depend on t.

We denote by t_i the random type observed in round i. Then, $\mathcal{R}$ gets an expected utility of

$$\mathbb{E}\left[\varrho_{t_i} \mid \sigma_i = \text{YES}\right] \quad \geq \quad \mathbb{E}_{A_i}\left[\sum_{t \in A_i} \frac{\varrho_t}{i}\right] \quad = \quad \varrho_E$$

when following the signal and taking action i.

By Lemma 4.6, we know that the signal $\sigma_i = \text{YES}$ in round i has the same probability for every subset A_i. Hence, there is no information on the remaining types to be observed in the upcoming rounds $r > i$ to be gathered. This means that they are distributed uniformly at random and yield an expected utility of ϱ_E for $\mathcal{R}$. Hence, dismissing the current action and waiting for a future round is not a profitable deviation for $\mathcal{R}$.

Similarly, assume that $\sigma_1, \dots, \sigma_{i-1} = \text{NO}$. Clearly, this means that no information on the type t_i of round i can be gathered and the type is distributed uniformly at random. For the expectation of $\mathcal{R}$, this implies $\mathbb{E}\left[\varrho_{t_i} \mid \wedge_{\ell=1}^{i-1} \sigma_\ell = \text{NO}\right] = \varrho_E$. Since a YES-signal in round i produces an expectation of at least ϱ_E for $\mathcal{R}$, a NO-signal can only yield an expectation of at most ϱ_E for $\mathcal{R}$ as either one or the other is always issued and the overall expectation is ϱ_E.

Overall, for every round $i \in [n]$, it is profitable for $\mathcal{R}$ to follow the signal σ_i computed by the Growing Pareto Mechanism. □

As the first approximation result, we show the constant-factor approximation to the optimum in the basic scenario for cardinal sender utility.

Theorem 4.8
For cardinal sender and receiver utility in the secretary scenario without disclosure, the Growing Pareto Mechanism with $s = \left\lfloor \frac{n}{\sqrt{3}} \right\rfloor$ yields a $\left(\frac{1}{3\sqrt{3}} - o(1)\right)$-approximation of the optimal expected utility in the corresponding basic scenario.

For the proof of the theorem, we use the fact that on average, removing a single type from the set of types only decreases the value of OPT, the value of the optimal expected utility for $\mathcal{S}$ in the basic scenario, by a small factor. To that end, we subdivide the set of types into the set $L = \{t \in [n] \mid \varrho_t \leq \varrho_E\}$ and $H = \{t \in [n] \mid \varrho_t > \varrho_E\} = [n] \setminus L$ of low and high receiver utility, respectively. By $d = |L|$, we denote the number of types with a low utility for $\mathcal{R}$. Recall that the Pareto Procedure scales all values such that $\varrho_{max} = \xi_{max} = 1$. Hence, we will assume that this holds. For any subset $M \subseteq [n]$, we will use OPT_{-M} to denote the optimal expected utility for $\mathcal{S}$ in the basic scenario with type set $[n] \setminus M$. Similar to $\text{OPT}_{\{-t\}}$, we define $\varrho_{E-t} = \frac{1}{n-1}\left(\sum_{\ell=1}^{n} \varrho_\ell - \varrho_t\right)$ to be the receiver's a priori expectation of a random type from type set $[n] \setminus \{t\}$.

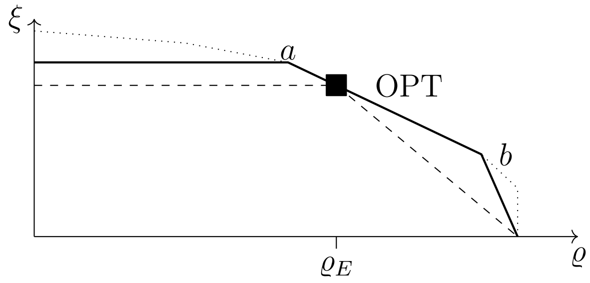


Figure 11: Adaptations as performed in the proof of Lemma 4.9. Dotted: Original Pareto frontier. Solid: Pareto frontier in the adapted instance with $\xi_t = 0$ for $t \neq a, b$. Dashed: Lower bound on the Pareto frontier when a and b remain in the candidate set

Recall that the Pareto Procedure identifies two types a and b on the Pareto frontier and uses a convex combination of the two to optimize the expectation for $\mathcal{S}$ while satisfying the constraint that $\mathcal{R}$ gets an expected utility of at least ϱ_E. If $\varrho_{c_\mathcal{S}} \geq \varrho_E$, it holds that $a = b$, otherwise $a \neq b$.

Lemma 4.9
Consider an instance of the basic scenario with type set $[n]$*. Let* a *and* b *be the types as determined in the Pareto Procedure. It holds that*

$$\sum_{t \neq a,b} \mathrm{OPT}_{-\{t\}} \geq (n-3) \cdot \mathrm{OPT} \quad .$$

Proof. Consider the Pareto Mechanism and the following adjustment to the instance. For all $t \neq a, b$, we set $\xi_t = 0$. This does not change the value of OPT, since OPT only depends on a, b, and the value of ϱ_E, neither of which was changed in this adjustment. Additionally, this step cannot increase the values of $\mathrm{OPT}_{-\{t\}}$ of the type set $[n] \setminus \{t\}$.

The adjusted Pareto frontier now only consists of (at most) three types, namely the types a and b and $(\varrho_{c_\mathcal{R}}, 0)$. Clearly, if $a = b$ or $b = c_\mathcal{R}$, fewer types are on the Pareto frontier and the adjusted Pareto frontier is a lower bound to the original one. In fact, we use an additional step to lower bound the utility of $\mathrm{OPT}_{-\{t\}}$ and assume that the Pareto frontier only consists of the points $(\varrho_E, \mathrm{OPT})$ and $(\varrho_{max}, 0)$. In Figure 11, we illustrate these adaptations. The dotted line represents the original Pareto frontier, the solid line the first adjustment and finally, the dashed line is used to represent the lower bound to the Pareto frontier without using a and b.

Using these steps and the partition of the type set into L and H described above, we are now able to bound the expected utility of $\mathrm{OPT}_{-\{t\}}$. If $t \in H$, we get that $\varrho_{E-t} \leq \varrho_E$, hence, there is no loss and $\mathrm{OPT}_{-\{t\}} \geq \mathrm{OPT}$. Clearly, this implies

$$\sum_{t \in H \setminus \{b\}} \mathrm{OPT} - \mathrm{OPT}_{-\{t\}} \leq 0. \quad .$$

If $t \in L$ on the other hand, $\varrho_{E-t} \geq \varrho_E$ and we use the (negative) slope $\frac{-\mathrm{OPT}}{\varrho_{\max} - \varrho_E}$ from $(\varrho_E, \mathrm{OPT})$ to $(\varrho_{max}, 0)$ and the change $\varrho_E - \varrho_{E-t}$ in the expected value for $\mathcal{R}$ to lower bound $\mathrm{OPT}_{-\{t\}}$. This gives us

$$\sum_{t \in L \setminus \{a\}} \mathrm{OPT} - \mathrm{OPT}_{-\{t\}} \leq \frac{\mathrm{OPT}}{\varrho_{\max} - \varrho_E} \cdot \sum_{t \in L} \left[\varrho_{E-t} - \varrho_E \right]$$

$$
\begin{aligned}
&= \frac{\mathrm{OPT}}{\varrho_{\max} - \varrho_E} \cdot \sum_{t \in L} \left[\sum_{\ell \in [n] \setminus \{t\}} \frac{\varrho_\ell}{n-1} - \varrho_E \right] \\
&= \frac{1}{n-1} \cdot \frac{\mathrm{OPT}}{\varrho_{\max} - \varrho_E} \cdot \sum_{t \in L} \left[\sum_{\ell \in L \setminus \{t\}} \varrho_\ell + \sum_{\ell \in H} \varrho_\ell - (n-1) \cdot \varrho_E \right] \\
&= \frac{\mathrm{OPT}}{(n-1) \cdot (\varrho_{\max} - \varrho_E)} \cdot \left[(d-1) \cdot \sum_{\ell \in L} \varrho_\ell + d \cdot \sum_{\ell \in H} \varrho_\ell - d \cdot (n-1) \cdot \varrho_E \right] \\
&= \frac{\mathrm{OPT}}{(n-1) \cdot (\varrho_{\max} - \varrho_E)} \cdot \left[(d-1) \cdot \left(\sum_{\ell=1}^{n} \varrho_\ell - n \cdot \varrho_E \right) + \sum_{\ell \in H} (\varrho_\ell - \varrho_E) \right] \\
&= \frac{\mathrm{OPT}}{(n-1) \cdot (\varrho_{\max} - \varrho_E)} \cdot \sum_{\ell \in H} (\varrho_\ell - \varrho_E) \\
&\le \frac{\mathrm{OPT}}{(n-1) \cdot (\varrho_{\max} - \varrho_E)} \cdot \sum_{\ell \in H} (\varrho_{\max} - \varrho_E) \\
&= \frac{\mathrm{OPT}}{(n-1) \cdot (\varrho_{\max} - \varrho_E)} \cdot (n-d) \cdot (\varrho_{\max} - \varrho_E) \\
&= \frac{n-d}{n-1} \cdot \mathrm{OPT} \quad \le \quad \mathrm{OPT} \ .
\end{aligned}
$$

Overall, this gives us

$$
\sum_{t \in [n] \setminus \{a,b\}} \mathrm{OPT} - \mathrm{OPT}_{-\{t\}} \le (n-2) \cdot \mathrm{OPT} - \sum_{t \in [n] \setminus \{a,b\}} \mathrm{OPT}_{-\{t\}} \le \mathrm{OPT}
$$

which implies the lemma. □

By OPT_i, we denote the expected value for $\mathcal{S}$ achieved by the Pareto Mechanism when applied to the basic scenario for the random subset of types A_i of size i. Note that $\mathrm{OPT}_n = \mathrm{OPT}$, the optimum in the basic scenario.

Corollary 4.10
For $i \ge 3$ it holds that

$$
\mathrm{OPT}_i \ge \mathrm{OPT} \cdot \prod_{\ell=i+1}^{n} \left(1 - \frac{3}{\ell} \right) = \frac{i \cdot (i-1) \cdot (i-2)}{n \cdot (n-1) \cdot (n-2)} \cdot \mathrm{OPT} \ .
$$

Proof. To generate A_i, we use $[n]$, the complete set of types. Then, we iteratively remove a random type until there are i types left. Note that for $i = n-1$ we have by Lemma 4.9

$$
\mathrm{OPT}_{n-1} = \frac{1}{n} \sum_{t \in [n]} \mathrm{OPT}_{-\{t\}} \ge \frac{1}{n} \sum_{t \in [n] \setminus \{a,b\}} \mathrm{OPT}_{-\{t\}} \ge \frac{n-3}{n} \mathrm{OPT} \ .
$$

For $i < n-1$, the result follows by repeated application. □

These results allow us to prove the approximation guarantee for cardinal sender utility, i.e., Theorem 4.8.

Proof of Theorem 4.8. In a given round $i = s+1, \dots, n-1$, the Growing Pareto Mechanism guarantees an expected utility of at least $\Pr[\sigma_i = \text{YES}] \cdot \text{OPT}_i$ to $\mathcal{S}$. For a simpler exposition, we underestimate the overall expected utility by assuming that round $i = n$ does not provide any utility. Note that here, a YES-signal is sent with probability 1 to satisfy persuasiveness-constraints regardless of the values of the final action's type. We optimize for the value $s = \lfloor \gamma \cdot n \rfloor$ for a constant $\gamma \in [0,1]$. Using Lemma 4.6 and Corollary 4.10, we get

$$\begin{aligned}
&\sum_{i=s+1}^{n-1} \frac{1}{i} \cdot \frac{s}{i-1} \cdot \frac{i \cdot (i-1) \cdot (i-2)}{n \cdot (n-1) \cdot (n-2)} \cdot \text{OPT} \\
&\qquad = \text{OPT} \cdot \frac{s}{n \cdot (n-1) \cdot (n-2)} \cdot \sum_{i=s+1}^{n-1} (i-2) \\
&\qquad = \text{OPT} \cdot \frac{s}{n \cdot (n-1) \cdot (n-2)} \cdot \left(\frac{n \cdot (n-1)}{2} - \frac{s \cdot (s+1)}{2} - 2(n-1-s) \right) \\
&\qquad = \text{OPT} \cdot \left(\frac{s}{2(n-2)} - \frac{s^2 \cdot (s+1)}{2n \cdot (n-1) \cdot (n-2)} - \frac{2s \cdot (n-1-s)}{n \cdot (n-1) \cdot (n-2)} \right) \\
&\qquad = \text{OPT} \cdot \frac{1}{2} \cdot \left(\gamma - \gamma^3 - o(1) \right) \ .
\end{aligned}$$

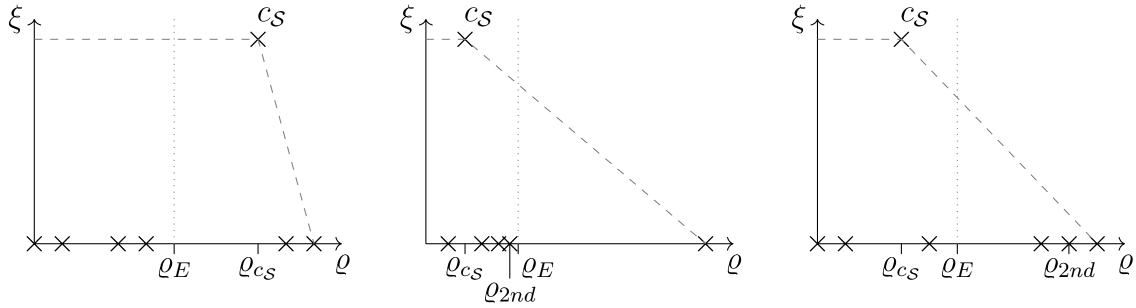
The last expression is maximized at $\gamma = 1/\sqrt{3}$, so we set the length of the sample phase $s = \lfloor n/\sqrt{3} \rfloor$. Hence, $\mathcal{S}$ gets an expected utility of at least $\left(\frac{1}{3\sqrt{3}} - o(1) \right) \cdot \text{OPT}$ when using the Growing Pareto Mechanism. □

This concludes the first part of the secretary scenario without disclosure for cardinal receiver utility. We continue with the setting of ordinal utility for $\mathcal{S}$. Compared to the case of cardinal sender utility, $\mathcal{S}$ is able to achieve a better approximation ratio in this setting while still using the Growing Pareto Mechanism, albeit with a different sample size.

Theorem 4.11
For ordinal sender and cardinal receiver utility in the secretary scenario without disclosure, the Growing Pareto Mechanism with $s = \lfloor n/2 \rfloor$ yields a success probability of at least $\left(\frac{1}{4} - o(1) \right)$ times the optimal success probability in the corresponding basic scenario.

We prove the theorem basically in the same way we proved the result for the cardinal sender utility. Crucially, we are able to improve the intermediate steps. The following Lemma 4.12 is an improved version of Lemma 4.9, the counterpart for the cardinal case, which makes Corollary 4.13 an improved version of Corollary 4.10.

In addition to the notation established for the cardinal case above, we denote by ϱ_{2nd} the second highest utility for $\mathcal{R}$ among all types. Observe that for Lemma 4.9, we lower bounded the expected utility for $\mathcal{S}$ in the case that b showed up in the current round by 0, where b is one of the two types a, b identified by the Pareto Procedure. In the ordinal case below, $b = c_{\mathcal{R}}$ and we are able to include the amount of $\alpha \cdot \text{OPT}_{-\{c_{\mathcal{R}}\}}$ for our bound. Here, α is the probability that the Pareto Procedure does not pick b to be the chosen type. In the ordinal case, it holds that $\alpha = \text{OPT}$, as $a = c_{\mathcal{S}}$ is chosen with probability α and a is the only type that provides any utility. Hence, we can write this as $\text{OPT} \cdot \text{OPT}_{-\{c_{\mathcal{R}}\}}$.



Case 1: $\varrho_{c_{\mathcal{S}}} > \varrho_E$ Case 2: $\varrho_{c_{\mathcal{S}}}, \varrho_{2nd} \leq \varrho_E$ Case 3: $\varrho_{c_{\mathcal{S}}} \leq \varrho_E < \varrho_{2nd}$

Figure 12: The three cases considered in the proof of Lemma 4.12

Lemma 4.12
Let OPT *and* $\mathrm{OPT}_{-\{t\}}$ *denote the expected utility in the basic scenario for type sets* $[n]$ *and* $[n] \setminus \{t\}$*, respectively. Then, the following holds:*

$$\sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \geq \mathrm{OPT} \left(n - 2 - \frac{1}{n-1} \right) \quad .$$

Proof. We subdivide the proof into three different cases, depending on the values of $\varrho_{c_{\mathcal{S}}}, \varrho_{2nd}$ and ϱ_E. The first case is $\varrho_{c_{\mathcal{S}}} > \varrho_E$, then $\varrho_{c_{\mathcal{S}}}, \varrho_{2nd} \leq \varrho_E$, and finally $\varrho_{c_{\mathcal{S}}} \leq \varrho_E < \varrho_{2nd}$. An illustration of the three cases is given in Figure 12.

Case 1: $\varrho_{c_{\mathcal{S}}} > \varrho_E$
The Pareto Procedure chooses $a = b = c_{\mathcal{S}}$. Clearly, if $c_{\mathcal{S}} = c_{\mathcal{R}}$, it holds that $\mathrm{OPT}_{-\{c_{\mathcal{R}}\}} = 0$. For all other $t \neq c_{\mathcal{R}}$, we have $\mathrm{OPT}_{-\{t\}} = \mathrm{OPT}$. Thus, the inequality in the lemma holds. Hence, we assume that $c_{\mathcal{S}} \neq c_{\mathcal{R}}$. This implies $\mathrm{OPT}_{-\{c_{\mathcal{R}}\}} = 1$ since $\varrho_{c_{\mathcal{S}}} \geq \varrho_{E-c_{\mathcal{R}}}$. For $t \in L$, we use a lower bound on the Pareto frontier as described in the proof of Lemma 4.9, using the slope from $(\varrho_E, \mathrm{OPT})$ to $(1, 0)$ and the change in the average value for $\mathcal{R}$, i.e., $\varrho_{E-t} - \varrho_E = \frac{n \cdot \varrho_E - \varrho_t}{n-1} - \varrho_E = \frac{\varrho_E - \varrho_t}{n-1}$. In combination, this means that $\mathrm{OPT}_{-\{t\}} \geq \mathrm{OPT} \cdot \left(1 - \frac{\varrho_E - \varrho_t}{(n-1)\cdot(1-\varrho_E)}\right)$. Note that $\mathrm{OPT} = 1$, which means that for $t \in H$, $\mathrm{OPT}_{-\{t\}} = \mathrm{OPT}$. Hence, we get

$$\begin{aligned}
\sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} &= \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \\
&= \sum_{t \in L} \mathrm{OPT}_{-\{t\}} + \sum_{t \in H \setminus \{c_{\mathcal{S}}, c_{\mathcal{R}}\}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \\
&\geq \mathrm{OPT} \cdot \left(d - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in L} (\varrho_E - \varrho_t) \right) + \sum_{t \in H \setminus \{c_{\mathcal{S}}, c_{\mathcal{R}}\}} \mathrm{OPT} + \mathrm{OPT} \\
&= \mathrm{OPT} \cdot \left(d - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in H} (\varrho_t - \varrho_E) \right) + (n-d-2) \cdot \mathrm{OPT} + \mathrm{OPT} \\
&\geq \mathrm{OPT} \cdot \left(n - 1 - \frac{(n-d)\cdot(1-\varrho_E)}{(n-1)\cdot(1-\varrho_E)} \right) \\
&\geq \mathrm{OPT} \cdot (n-2) \quad \geq \quad \mathrm{OPT} \cdot \left(n - 2 - \frac{1}{n-1} \right) \quad .
\end{aligned}$$

Note that for the fourth line, we use that $c_{\mathcal{S}} \neq c_{\mathcal{R}}$ and $\sum_{t \in L} (\varrho_E - \varrho_t) = \sum_{t \in H} (\varrho_t - \varrho_E)$ since $\sum_{t \in [n]} \varrho_t = \sum_{t \in L} \varrho_t + \sum_{t \in H} \varrho_t = n \cdot \varrho_E$. For the fifth line, we use $\varrho_t \leq 1$. This shows the claim of the lemma for case 1.

Case 2: $\varrho_{c_\mathcal{S}}, \varrho_{2nd} \leq \varrho_E$
Clearly, the Pareto Procedure chooses $a = c_\mathcal{S}$ and $b = c_\mathcal{R}$. The Pareto frontier consists of the line segment between $c_\mathcal{S}$ and $c_\mathcal{R}$ and $(\varrho_E, \mathrm{OPT})$ is somewhere on that line segment which has a (negative) slope value of $\frac{-\mathrm{OPT}}{1-\varrho_E}$. We shortly discuss the implications of removing a type $t \neq c_\mathcal{S}$. If a type $t \in L \setminus \{c_\mathcal{S}\}$ is removed, $\varrho_{E-t} \geq \varrho_E$ and $\mathrm{OPT}_{-\{t\}} = \mathrm{OPT} \cdot \left(1 - \frac{\varrho_E - \varrho_t}{(n-1)\cdot(1-\varrho_E)}\right)$. Note that in this case, we use the true Pareto frontier and not only a lower bound. Since $\varrho_{2nd} \leq \varrho_E$, we can see that $H = \{c_\mathcal{R}\}$. Hence, we get

$$\begin{aligned}
\sum_{t \neq c_\mathcal{S}, c_\mathcal{R}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_\mathcal{R}\}} &= \sum_{t \in L \setminus \{c_\mathcal{S}\}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_\mathcal{R}\}} \\
&\geq \mathrm{OPT} \cdot \left((n-2) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in L} (\varrho_E - \varrho_t) \right) \\
&= \mathrm{OPT} \cdot \left((n-2) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in H} (\varrho_t - \varrho_E) \right) \\
&= \mathrm{OPT} \cdot \left(n - 2 - \frac{1}{n-1} \right) \quad .
\end{aligned}$$

The first inequality holds due to $\varrho_{c_\mathcal{S}} \leq \varrho_E$ and $\mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_\mathcal{R}\}} \geq 0$. For the third line, we again use that $\sum_{t \in L}(\varrho_E - \varrho_t) = \sum_{t \in H}(\varrho_t - \varrho_E)$. For the fourth line, we plug in $H = \{c_\mathcal{R}\}$ with $\varrho_{c_\mathcal{R}} = 1$ which shows the claim for case 2.

Case 3: $\varrho_{c_\mathcal{S}} \leq \varrho_E < \varrho_{2nd}$
Similar to case 2, the Pareto Procedure chooses $a = c_\mathcal{S}$ and $b = c_\mathcal{R}$. The loss of utility for $\mathcal{S}$ when a type $t \in L$ is removed stays the same. When removing a type $t \in H$, this leads to an improvement in utility for $\mathcal{S}$ and $\mathrm{OPT}_{-\{t\}} \geq \mathrm{OPT}$. For $t \in H \setminus \{c_\mathcal{R}\}$, we get

$$\begin{aligned}
\mathrm{OPT}_{-\{t\}} &= \min\left\{1, \mathrm{OPT} + \frac{\mathrm{OPT} \cdot (\varrho_t - \varrho_E)}{(n-1)\cdot(1-\varrho_E)}\right\} \\
&= \mathrm{OPT} \cdot \left(1 + \min\left\{\frac{\varrho_t - \varrho_E}{(n-1)\cdot(1-\varrho_E)}, \frac{\varrho_E - \varrho_{c_\mathcal{S}}}{1-\varrho_E}\right\}\right) \quad .
\end{aligned}$$

When removing type $c_\mathcal{R}$, the Pareto frontier shifts. If $\varrho_{E-c_\mathcal{R}} \leq \varrho_{c_\mathcal{S}}$, the Pareto Procedure deterministically chooses $c = c_\mathcal{S}$ and $\mathrm{OPT}_{-\{c_\mathcal{R}\}} = 1$. Otherwise, the new Pareto frontier consists of the line segment between $c_\mathcal{S}$ and the type with the second highest utility for $\mathcal{R}$. The slope of this line segment has (negative) value $\frac{-1}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}}$. This means that $\mathrm{OPT}_{-\{c_\mathcal{R}\}} = \frac{\varrho_{2nd} - \varrho_{E-c_\mathcal{R}}}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}}$. In combination, we have $\mathrm{OPT}_{-\{c_\mathcal{R}\}} = \min\left\{1, \frac{\varrho_{2nd} - \varrho_{E-c_\mathcal{R}}}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}}\right\} \geq \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}}$. This gives us

$$\begin{aligned}
&\sum_{t \neq c_\mathcal{S}, c_\mathcal{R}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_\mathcal{R}\}} \\
&= \sum_{t \in L \setminus \{c_\mathcal{S}\}} \mathrm{OPT}_{-\{t\}} + \sum_{t \in H \setminus \{c_\mathcal{R}\}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \min\left\{1, \frac{\varrho_{2nd} - \varrho_{E-c_\mathcal{R}}}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}}\right\} \\
&\geq \mathrm{OPT} \cdot \left[(d-1) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in L \setminus \{c_\mathcal{S}\}} (\varrho_E - \varrho_t) + (n-d-1) \right. \\
&\qquad \left. + \sum_{t \in H \setminus \{c_\mathcal{R}\}} \min\left\{\frac{\varrho_t - \varrho_E}{(n-1)\cdot(1-\varrho_E)}, \frac{\varrho_E - \varrho_{c_\mathcal{S}}}{1-\varrho_E}\right\} + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_\mathcal{S}}} \right]
\end{aligned}$$

$$
\begin{aligned}
&\geq \mathrm{OPT} \cdot \Bigg[(n-2) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in L} (\varrho_E - \varrho_t) \\
&\qquad + \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \min\left\{ \frac{\varrho_t - \varrho_E}{(n-1)\cdot(1-\varrho_E)}, \frac{\varrho_E - \varrho_{c_{\mathcal{S}}}}{1-\varrho_E} \right\} + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} \Bigg] \\
&= \mathrm{OPT} \cdot \Bigg[(n-2) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in H} (\varrho_t - \varrho_E) \\
&\qquad + \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \min\left\{ \frac{\varrho_t - \varrho_E}{(n-1)\cdot(1-\varrho_E)}, \frac{\varrho_E - \varrho_{c_{\mathcal{S}}}}{1-\varrho_E} \right\} + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} \Bigg] \\
&= \mathrm{OPT} \cdot \Bigg[(n-2) - \frac{1}{(n-1)\cdot(1-\varrho_E)} \cdot \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} (\varrho_t - \varrho_E) - \frac{\varrho_{c_{\mathcal{R}}} - \varrho_E}{(n-1)\cdot(1-\varrho_E)} \\
&\qquad - \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \min\left\{ \frac{\varrho_t - \varrho_E}{(n-1)\cdot(1-\varrho_E)}, \frac{\varrho_E - \varrho_{c_{\mathcal{S}}}}{1-\varrho_E} \right\} + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} \Bigg] \\
&= \mathrm{OPT} \cdot \Bigg[(n-2) + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} - \frac{1}{n-1} \\
&\qquad - \frac{1}{n-1} \cdot \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \left(\frac{\varrho_t - \varrho_E}{1-\varrho_E} - \min\left\{ \frac{\varrho_t - \varrho_E}{1-\varrho_E}, \frac{(\varrho_E - \varrho_{c_{\mathcal{S}}})\cdot(n-1)}{1-\varrho_E} \right\} \right) \Bigg] \quad .
\end{aligned}
$$

For the final equality, we use that $\varrho_{c_{\mathcal{R}}} = 1$. We continue with two different subcases.

Subcase 3.1: $\varrho_E < \frac{\varrho_{2nd}+(n-1)\varrho_{c_{\mathcal{S}}}}{n}$
Here, we see

$$
\begin{aligned}
&\sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \\
&\quad \geq \mathrm{OPT} \cdot \Bigg[(n-2) + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} - \frac{1}{n-1} \\
&\qquad\quad - \frac{1}{n-1} \cdot \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \left(\frac{\varrho_t - \varrho_E}{1-\varrho_E} - \min\left\{ \frac{\varrho_t - \varrho_E}{1-\varrho_E}, \frac{(\varrho_E - \varrho_{c_{\mathcal{S}}})\cdot(n-1)}{1-\varrho_E} \right\} \right) \Bigg] \\
&\quad \geq \mathrm{OPT} \cdot \left[(n-2) + 1 - \frac{\varrho_E - \varrho_{c_{\mathcal{S}}}}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} - \frac{1}{n-1} - \frac{n-2}{n-1} \right] \\
&\quad \geq \mathrm{OPT} \cdot \left[n - 2 - \frac{1}{n} \right] \quad .
\end{aligned}
$$

For the penultimate line we bound the value of the sum by $\frac{n-2}{n-1}$. Towards this end, we use that the value of the min is at least 0 and $\varrho_t \leq 1$. Hence, for each $t \in H \setminus \{c_{\mathcal{R}}\}$, we have $\frac{\varrho_t - \varrho_E}{1-\varrho_E} - \min\left\{ \frac{\varrho_t - \varrho_E}{1-\varrho_E}, \frac{(\varrho_E - \varrho_{c_{\mathcal{S}}})\cdot(n-1)}{1-\varrho_E} \right\} \leq 1$ and there are at most $n-2$ types in $H \setminus \{c_{\mathcal{R}}\}$. In the last inequality, we used the assumption that $\varrho_E < \frac{\varrho_{2nd}+(n-1)\cdot\varrho_{c_{\mathcal{S}}}}{n}$ and thus

$$
-\frac{\varrho_E - \varrho_{c_{\mathcal{S}}}}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} \geq -\frac{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}}{n \cdot (\varrho_{2nd} - \varrho_{c_{\mathcal{S}}})} = -\frac{1}{n} \quad .
$$

Subcase 3.2: $\varrho_E \geq \frac{\varrho_{2nd} + (n-1) \cdot \varrho_{c_{\mathcal{S}}}}{n}$
In this case, we see

$$\sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \left(\frac{\varrho_t - \varrho_E}{1 - \varrho_E} - \min \left\{ \frac{\varrho_t - \varrho_E}{1 - \varrho_E}, \frac{(\varrho_E - \varrho_{c_{\mathcal{S}}}) \cdot (n-1)}{1 - \varrho_E} \right\} \right) = 0$$

as $\varrho_t \leq \varrho_{2nd}$ for all $t \in H \setminus \{c_{\mathcal{R}}\}$ and thus $\varrho_t - \varrho_E \leq \varrho_{2nd} - \varrho_E \leq (\varrho_E - \varrho_{c_{\mathcal{S}}}) \cdot (n-1)$ due to our assumption $\varrho_E \geq \frac{\varrho_{2nd} + (n-1) \cdot \varrho_{c_{\mathcal{S}}}}{n}$. This implies

$$\begin{aligned}
&\sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \\
&\quad \geq \mathrm{OPT} \cdot \Bigg[(n-2) + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} - \frac{1}{n-1} \\
&\qquad\qquad - \frac{1}{n-1} \sum_{t \in H \setminus \{c_{\mathcal{R}}\}} \left(\frac{\varrho_t - \varrho_E}{1 - \varrho_E} - \min \left\{ \frac{\varrho_t - \varrho_E}{1 - \varrho_E}, \frac{(\varrho_E - \varrho_{c_{\mathcal{S}}})(n-1)}{1 - \varrho_E} \right\} \right) \Bigg] \\
&\quad = \mathrm{OPT} \cdot \left[(n-2) + \frac{\varrho_{2nd} - \varrho_E}{\varrho_{2nd} - \varrho_{c_{\mathcal{S}}}} - \frac{1}{n-1} \right] \\
&\quad \geq \mathrm{OPT} \cdot \left[n - 2 - \frac{1}{n-1} \right] \quad .
\end{aligned}$$

This shows case 3 and thus, concludes the proof of the lemma. □

Analogous to the cardinal case, by OPT_i we denote in the following the success probability of the Pareto Mechanism when applied to the basic scenario with random subset A_i. This allows us to prove the following Corollary 4.13, an improved version of Corollary 4.10, in a similar fashion, i.e., by repeatedly applying Lemma 4.12.

Corollary 4.13
For all $i \in [n]$ it holds that $\mathrm{OPT}_i \geq \frac{(i-2) \cdot (i-1)}{(n-2) \cdot (n-1)} \cdot \mathrm{OPT}$.

Proof. We generate A_i by iteratively removing a random type, starting with the complete set of types $[n]$. For the first step, we have

$$\begin{aligned}
\mathrm{OPT}_{n-1} &= \frac{1}{n} \cdot \sum_{t \in [n]} \mathrm{OPT}_{-\{t\}} \\
&= \frac{1}{n} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \frac{1}{n} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \\
&\geq \frac{1}{n} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}} + \frac{1}{n} \cdot \mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \\
&\geq \frac{n - 2 - \frac{1}{n-1}}{n} \cdot \mathrm{OPT} \quad .
\end{aligned}$$

For the second line, we use that $\mathrm{OPT}_{-\{c_{\mathcal{S}}\}} = 0$. The first inequality is due to $\mathrm{OPT} \leq 1$ and the final line uses Lemma 4.12. For $i < n-1$, we repeatedly apply the same procedure to get

$$\mathrm{OPT}_i \geq \mathrm{OPT} \cdot \prod_{\ell = i+1}^{n} \frac{\ell - 2 - \frac{1}{\ell - 1}}{\ell}$$

$$
\begin{aligned}
&= \mathrm{OPT} \cdot \prod_{\ell=i+1}^{n} \frac{\ell \cdot (\ell-3)+1}{\ell \cdot (\ell-1)} \\
&\geq \mathrm{OPT} \cdot \prod_{\ell=i+1}^{n} \frac{\ell-3}{\ell-1} \\
&= \mathrm{OPT} \cdot \frac{i-2}{i} \cdot \frac{i-1}{i+1} \cdot \frac{i}{i+2} \cdots \frac{n-4}{n-2} \cdot \frac{n-3}{n-1} \\
&= \mathrm{OPT} \cdot \frac{(i-2)\cdot(i-1)}{(n-2)\cdot(n-1)} \quad .
\end{aligned}
$$

□

With Corollary 4.13 and Lemma 4.12, we can now prove Theorem 4.11 in a similar fashion to the proof of Theorem 4.8.

Proof of Theorem 4.11. Clearly, in rounds $i = 1, \ldots, s$, the success probability of the Growing Pareto Mechanism is 0. In the following rounds $i = s+1, \ldots, n-1$, the success probability is at least $\Pr[\sigma_i = \mathrm{YES}] \cdot \mathrm{OPT}_i$. By Lemma 4.6, this is $\frac{1}{i} \cdot \frac{s}{i-1} \cdot \mathrm{OPT}_i$. If the final round n is reached, a signal $\sigma_n = \mathrm{YES}$ is always sent, regardless of the type observed. Hence, we underestimate the success probability in that round by 0. Using linearity of expectation and optimizing the sample size $s = \lfloor \gamma \cdot n \rfloor$ for a constant $\gamma \in [0, 1]$, we obtain

$$
\begin{aligned}
&\sum_{i=s+1}^{n-1} \frac{1}{i} \cdot \frac{s}{i-1} \cdot \frac{(i-1)\cdot(i-2)}{(n-1)\cdot(n-2)} \cdot \mathrm{OPT} \\
&\qquad = \mathrm{OPT} \cdot \frac{s}{(n-1)\cdot(n-2)} \cdot \sum_{i=s+1}^{n-1} \frac{i-2}{i} \\
&\qquad = \mathrm{OPT} \cdot \frac{s}{(n-1)\cdot(n-2)} \cdot \left((n-1) - s - 2 \cdot \sum_{i=s+1}^{n-1} \frac{1}{i} \right) \\
&\qquad = \mathrm{OPT} \cdot \left(\frac{s}{n-2} - \frac{s^2}{(n-1)\cdot(n-2)} - \frac{2s \cdot (H_{n-1} - H_s)}{(n-1)\cdot(n-2)} \right) \\
&\qquad = \mathrm{OPT} \cdot \left(\gamma - \gamma^2 - o(1) \right) \quad .
\end{aligned}
$$

In the penultimate line, we use $H_\ell = \sum_{i=1}^{\ell} 1/i$, i.e., the ℓ-th harmonic number. Clearly, $\gamma - \gamma^2$ is maximized at $\gamma = 1/2$. Thus, we choose a sample size $s = \lfloor n/2 \rfloor$. This implies the theorem, i.e., a success probability of at least $(1/4 - o(1)) \cdot \mathrm{OPT}$ for the Growing Pareto Mechanism. □Theorem 4.11

This concludes the section on the secretary scenario without Disclosure. In the following, we discuss our results for the scenarios with disclosure. We start with the basic scenario, i.e., $\mathcal{S}$ and $\mathcal{R}$ both knowing the complete vector of types.

Basic Scenario with Disclosure

In the setting with disclosure, $\mathcal{R}$ learns the types of dismissed actions before receiving the next signal. For the basic scenario, this implies that $\mathcal{R}$ knows exactly what the remaining set of types $\mathcal{C} \subseteq [n]$ looks like. Clearly, $\mathcal{S}$ still has the advantage of knowing the realized type of the current action. Only in round n – should the process reach this round – both $\mathcal{S}$ and $\mathcal{R}$ know exactly the type of the final action. Since values are nonnegative, sending $\sigma_n = \text{YES}$ in the final round and accepting action n are the only sensible choices for $\mathcal{S}$ and $\mathcal{R}$, respectively, regardless of the type of action n. Knowing this, $\mathcal{S}$ can therefore optimize the expected utility in the penultimate round for all possible sets of types with cardinality 2 while satisfying persuasiveness constraints. This allows $\mathcal{S}$ to then optimize for round $n-2$ and subsets of size 3. Continuing this train of thought, we get a characterization of the optimal signaling mechanism using backwards induction. Since this optimal mechanism requires the solution of an exponential number of linear programs, we also describe another variant of the Pareto Mechanism which takes the shrinking set of types into account. This mechanism runs in polynomial time and achieves an approximation ratio of at least $1/2 - o(1)$ for ordinal sender utility and $1/3 - o(1)$ for cardinal sender utility, compared to the performance of the Pareto Mechanism in the corresponding basic scenario without disclosure. We complement these bounds by showing that no mechanism can achieve a better ratio than $1/2$, thereby proving that the Shrinking Pareto Mechanism is asymptotically optimal for the case of ordinal sender utility.

We begin by describing our optimal mechanism using backwards induction. The mechanism works for both ordinal as well as cardinal utility for $\mathcal{S}$ by setting $\xi_{max} = 1$ and $\xi_t = 0$ for all types $t \neq c_{\mathcal{S}}$ in the ordinal case. Then, expected utility for $\mathcal{S}$ and success probability, i.e., the probability of taking an action with type $c_{\mathcal{S}}$, are the same.

Theorem 4.14
For cardinal sender and receiver utility, an optimal persuasive mechanism for the sender's expected utility in the basic scenario with disclosure can be computed by solving 2^n linear programs.

For completeness, we state the result for ordinal sender utility in the following corollary.

Corollary 4.15
For ordinal sender and cardinal receiver utility, an optimal persuasive mechanism for the sender's success probability in the basic scenario with disclosure can be computed by solving 2^n linear programs.

Proof of Theorem 4.14. Consider the following. At the beginning of round $i \in [n]$, the set $\mathcal{C} \subseteq [n]$ remains. Suppose $\mathcal{S}$ observes type t in the current round and has already computed an optimal persuasive signaling scheme φ for rounds $i+1, \dots, n$ for the remaining type set $\mathcal{C} \setminus \{t\}$. We denote by $u^{\mathcal{S}}_{\mathcal{C}\setminus\{t\}}$ the expected utility from φ for $\mathcal{S}$. Similarly, we denote by $u^{\mathcal{R}}_{\mathcal{C}\setminus\{t\}}$ the expected utility for $\mathcal{R}$. Note that φ is persuasive, hence, it is in the receiver's interest to follow the recommendations given by φ.

Without loss of generality, we assume that $\max_{t\in\mathcal{C}} \xi_t > 0$, otherwise, the expected utility of $\mathcal{S}$ is 0 and performing an online search for a type with maximum receiver utility yields an optimal persuasive scheme.

For round n, the optimal persuasive signaling scheme is to send $\sigma_n = \text{YES}$ with probability 1. For rounds $i < n$, we denote by

$$x_t^{\mathcal{C}} = \Pr[\sigma_i = \text{YES} \mid t \in \mathcal{C} \text{ arrives in round } i]$$

the probability to send a signal YES in round i conditional on type t being observed in round i with remaining type set $\mathcal{C}$. Under the assumption that $\boldsymbol{x}$ gives rise to a persuasive extension of φ to round i and $\mathcal{R}$ therefore follows σ_i, the expected utility for $\mathcal{S}$ is

$$\frac{1}{|\mathcal{C}|} \cdot \sum_{t \in \mathcal{C}} \left(x_t^{\mathcal{C}} \cdot \xi_t + \left(1 - x_t^{\mathcal{C}}\right) \cdot u_{\mathcal{C} \setminus \{t\}}^{\mathcal{S}} \right) \ .$$

Clearly, this is a linear function. For every type set $\mathcal{C} \subseteq [n]$ and every type $t \in \mathcal{C}$, $x_t^{\mathcal{C}} \in [0, 1]$ is a linear constraint. Thus, by expressing the persuasiveness constraints as linear functions, we are able to state this optimization problem as a linear program. There are two persuasiveness constraints. Clearly, when receiving a YES-signal, $\mathcal{R}$ should be incentivized to take the current action and when getting a NO-signal, $\mathcal{R}$ should not want to take the current action.

Let us now show that both can be satisfied, i.e., the persuasiveness constraints can be expressed as linear functions and the resulting $\boldsymbol{x}$ indeed provides a persuasive extension to φ. Assume that $\sigma_i = \text{YES}$. By our definition of $\boldsymbol{x}$, this happens with a probability $\Pr[\sigma_i = \text{YES} \mid \mathcal{C}] = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} x_t^{\mathcal{C}}$. Without loss of generality, this probability can be assumed to be positive, otherwise, there is nothing to prove as $\mathcal{R}$ would not get a YES-signal. By Bayes' law, the probability that the current action has type t is $p_t^Y = \frac{x_t^{\mathcal{C}}}{\sum_{t' \in \mathcal{C}} x_{t'}^{\mathcal{C}}}$. Hence, by following the recommendation to take the current action, $\mathcal{R}$ obtains a utility of $\sum_{t \in \mathcal{C}} p_t^Y \cdot \varrho_t$. If $\mathcal{R}$ deviates and dismisses the action, $\mathcal{S}$ will not reveal any more information to $\mathcal{R}$. Therefore, the expected utility for $\mathcal{R}$ becomes $\sum_{t \in \mathcal{C}} p_t^Y \cdot \frac{1}{|\mathcal{C}|-1} \sum_{t' \in \mathcal{C} \setminus \{t\}} \varrho_{t'}$. Hence, to be persuasive, the signaling scheme must guarantee that

$$\sum_{t \in \mathcal{C}} p_t^Y \cdot \varrho_t \geq \sum_{t \in \mathcal{C}} p_t^Y \cdot \frac{1}{|\mathcal{C}| - 1} \cdot \sum_{t' \in \mathcal{C} \setminus \{t\}} \varrho_{t'} \ .$$

Clearly, this is not a linear inequality, but multiplying both sides by $\sum_{t' \in \mathcal{C}} x_{t'}^{\mathcal{C}}$ gives us the equivalent inequality

$$\sum_{t \in \mathcal{C}} x_t^{\mathcal{C}} \cdot \varrho_t \geq \sum_{t \in \mathcal{C}} x_t^{\mathcal{C}} \cdot \frac{1}{|\mathcal{C}| - 1} \cdot \sum_{t' \in \mathcal{C} \setminus \{t\}} \varrho_{t'} \ ,$$

or, equivalently

$$\sum_{t \in \mathcal{C}} x_t^{\mathcal{C}} \cdot \left(\varrho_t - \frac{1}{|\mathcal{C}| - 1} \cdot \sum_{t' \in \mathcal{C} \setminus \{t\}} \varrho_{t'} \right) \geq 0 \ . \tag{4.8}$$

Similarly, $\mathcal{R}$ should also want to follow a NO-signal, i.e., dismiss the action in round i if $\sigma_i = \text{NO}$. Again, we assume without loss of generality that the probability for a NO-signal is positive, otherwise, there is nothing to prove. We denote by $p_t^N = \frac{1 - x_t^{\mathcal{C}}}{\sum_{t' \in \mathcal{C}} (1 - x_{t'}^{\mathcal{C}})}$ the probability that the current action has type t if the current signal is $\sigma_i = \text{NO}$ for remaining subset $\mathcal{C}$ of types. When deviating and therefore taking the current action against the recommendation, $\mathcal{R}$ gets an expected utility of $\sum_{t \in \mathcal{C}} p_t^N \cdot \varrho_t$. If, on the other hand, $\mathcal{R}$ follows the signal, the expected utility is $\sum_{t \in \mathcal{C}} p_t^N \cdot u_{\mathcal{C} \setminus \{t\}}^{\mathcal{R}}$. Thus, the following constraint arises.

$$\sum_{t \in \mathcal{C}} p_t^N \cdot u_{\mathcal{C} \setminus \{t\}}^{\mathcal{R}} \geq \sum_{t \in \mathcal{C}} p_t^N \cdot \varrho_t$$

Again, this is not a linear inequality but can be equivalently expressed as

$$\sum_{t \in \mathcal{C}} \left(1 - x_t^{\mathcal{C}}\right) \cdot u_{\mathcal{C} \setminus \{t\}}^{\mathcal{R}} \geq \sum_{t \in \mathcal{C}} \left(1 - x_t^{\mathcal{C}}\right) \cdot \varrho_t$$

or

$$\sum_{t\in\mathcal{C}} \left(1 - x_t^{\mathcal{C}}\right) \cdot \left(u^{\mathcal{R}}_{\mathcal{C}\setminus\{t\}} - \varrho_t\right) \geq 0 \ . \tag{4.9}$$

Hence, we can express the optimization problem for each subset of types $\mathcal{C}$ as a linear program. The following claim states that (4.9) is redundant and thus, the linear program can be expressed using only a single persuasiveness constraint.

Claim 1
Constraint (4.9) *is redundant and implied by* (4.8).

We show the claim below. Using the claim, the linear program to determine $u^{\mathcal{S}}_{\mathcal{C}}$ is:

$$\max_{\boldsymbol{x}} \quad \frac{1}{|\mathcal{C}|}\sum_{t\in\mathcal{C}} \left(x_t^{\mathcal{C}} \cdot \xi_t + \left(1 - x_t^{\mathcal{C}}\right) \cdot u^{\mathcal{S}}_{\mathcal{C}\setminus\{t\}}\right) \tag{4.10a}$$

$$\text{s.t.} \quad \sum_{t\in\mathcal{C}} x_t^{\mathcal{C}} \left[\varrho_t - \frac{1}{|\mathcal{C}|-1}\sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'}\right] \geq 0 \tag{4.10b}$$

$$x_t^{\mathcal{C}} \in [0,1] \text{ for all } t \in \mathcal{C} \tag{4.10c}$$

Solving the corresponding linear program for every subset $\mathcal{C} \subseteq [n]$ yields an optimal signaling scheme. $\square_{\text{Theorem 4.14}}$

Proof of Claim 1. The values $u^{\mathcal{R}}_{\mathcal{C}\setminus\{t\}}$ denote the expected utility for $\mathcal{R}$ from the optimal persuasive signaling scheme for the remaining set of types $\mathcal{C} \setminus \{t\}$. Since $\mathcal{R}$ can always take a random action, we clearly have that $u^{\mathcal{R}}_{\mathcal{C}\setminus\{t\}} \geq \frac{1}{|\mathcal{C}|-1} \cdot \sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'}$. Using $x_t^{\mathcal{C}} \in [0,1]$, we can directly lower bound the left-hand side of (4.9) by plugging this inequality into the left-hand side of (4.9). We get

$$\begin{aligned}
&\sum_{t\in\mathcal{C}} \left(1 - x_t^{\mathcal{C}}\right) \cdot \left(u^{\mathcal{R}}_{\mathcal{C}\setminus\{t\}} - \varrho_t\right) \\
&\qquad \geq \sum_{t\in\mathcal{C}} \left(1 - x_t^{\mathcal{C}}\right) \cdot \left(\frac{1}{|\mathcal{C}|-1} \cdot \sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'} - \varrho_t\right) \\
&\qquad = \sum_{t\in\mathcal{C}} \left(\frac{1}{|\mathcal{C}|-1} \cdot \sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'} - \varrho_t\right) - \sum_{t\in\mathcal{C}} x_t^{\mathcal{C}} \cdot \left(\frac{1}{|\mathcal{C}|-1} \cdot \sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'} - \varrho_t\right) \\
&\qquad = \sum_{t\in\mathcal{C}} x_t^{\mathcal{C}} \cdot \left(\varrho_t - \frac{1}{|\mathcal{C}|-1} \cdot \sum_{t'\in\mathcal{C}\setminus\{t\}} \varrho_{t'}\right) \ .
\end{aligned}$$

The final line is the left-hand side of (4.8). Thus, if (4.8) is satisfied, then (4.9) must be satisfied as well. This proves the claim. $\square$

Before we move on to our polynomial time algorithms based on the Pareto Procedure, we show that even with an optimal signaling mechanism, there are cases in which $\mathcal{S}$ cannot achieve more than $1/2$ of the optimal utility in the basic scenario without disclosure. This result again holds for both cardinal and ordinal utility for $\mathcal{S}$ as $\xi_{max} = 1$ and $\xi_t = 0$ for all $t \neq c_{\mathcal{S}}$.

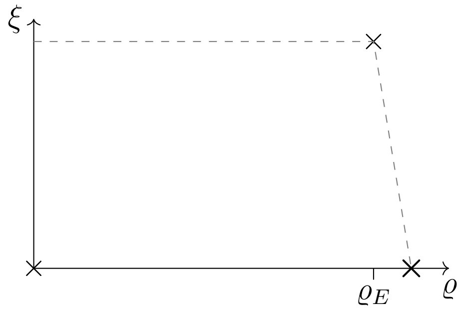


Figure 13: Instance for the proof of Theorem 4.16

Theorem 4.16
For both cardinal and ordinal sender utility and cardinal receiver utility, for every $\varepsilon > 0$, there is an instance such that every persuasive mechanism in the basic scenario with disclosure guarantees at most a fraction of $\left(\frac{1}{2} + \varepsilon\right)$ of the optimum in the basic scenario without disclosure.

Proof. We use the following class of instances with n types. Only type 1 provides any positive utility to $\mathcal{S}$. It has a value-pair $(\varrho_1, \xi_1) = (\frac{n-2}{n-1}, 1)$. Type 2 provides no utility for $\mathcal{R}$, having a value-pair of $(\varrho_2, \xi_2) = (0, 0)$. The remaining types $t = 3, \dots, n$ are all great for $\mathcal{R}$ with values $(\varrho_t, \xi_t) = (1, 0)$.[1] For an illustration, see Figure 13.

Clearly, type 1 is the best type for $\mathcal{S}$ and further, the only type with positive utility for $\mathcal{S}$. Hence, the expected utility $\mathcal{S}$ obtains is equal to the success probability, i.e., the probability that an action with type 1 is taken by $\mathcal{R}$. This allows us to state the proof for ordinal and cardinal sender utility at the same time. In the following, we will only use "expected utility".

The expected utility for $\mathcal{R}$ when taking a random action is $\varrho_E = \frac{n-2}{n-1}$, hence, the Pareto Mechanism would wait for type 1 to be observed and signal YES in that round. Hence, the expected utility for $\mathcal{S}$ in the basic scenario without disclosure is 1.

For the expected utility obtainable by $\mathcal{S}$ in the scenario with disclosure, we use the notation of the linear programs derived in Theorem 4.14, i.e., $x_t^{\mathcal{C}}$ is the probability that $\mathcal{S}$ signals YES in a round conditional on the remaining type set $\mathcal{C}$ and current type $t \in \mathcal{C}$. Clearly, setting $x_2^{\mathcal{C}} = 0$, i.e., sending NO when type 2 is observed, for all $\mathcal{C} \ni 2$ can only increase the sender's objective (4.10a) and does not negatively impact constraint (4.10b). Hence, we can assume without loss that $x_2^{\mathcal{C}} = 0$ for all $\mathcal{C} \ni 2$ in an optimal scheme.

Since type 1 is the only type providing positive value for $\mathcal{S}$, an optimal mechanism will set $x_1^{\mathcal{C}} = 1$ without loss of generality for all $\mathcal{C} \ni 1$. If $\varrho_1 < \frac{1}{|\mathcal{C}|-1} \cdot \sum_{t' \in \mathcal{C} \setminus \{1\}} \varrho_{t'}$, $\mathcal{S}$ cannot achieve a higher expectation than $\frac{1}{|\mathcal{C}|}$, which can be obtained by setting $x_t^{\mathcal{C}} = 1$ for all $t \in \mathcal{C}$. If, on the other hand, $\varrho_1 \geq \frac{1}{|\mathcal{C}|-1} \cdot \sum_{t' \in \mathcal{C} \setminus \{1\}} \varrho_{t'}$, setting $x_1^{\mathcal{C}} = 1$ and $x_t^{\mathcal{C}} = 0$ for all $t \in \mathcal{C} \setminus \{1\}$ is optimal. If $1 \notin \mathcal{C}$, $\mathcal{S}$ cannot get a positive utility. Thus, without loss of generality, $x_t^{\mathcal{C}} = 1$ for all $t \in \mathcal{C}$, $t \neq 2$.

Let $\ell = \sqrt{n}$. We denote by E the event that

1. type 1 or type 2 is among the first $n - \ell$ actions, and

[1]For simplicity, we assume that all these values are the same. Using slight perturbations of these values, our original assumption of distinct values can be guaranteed without requiring substantial changes to the proof.

Algorithm 11: Shrinking Pareto Mechanism

```
Input: Set of valuation pairs (ϱ_t, ξ_t)_{t∈[n]}, online sequence of types θ_1, ..., θ_n
Set R_1 = [n] and recSent = False.
for round i = 1, ..., n do
    Set R_{i+1} = R_i \ {θ_i}.
    if recSent = True then Signal NO.
    else
        Let c_i be the type chosen by Pareto Procedure on set R_i.
        if c_i = θ_i then Signal YES and set recSent = True.
        else Signal NO.
```

2. type 2 is observed before type 1.

The probability that both types arrive in the final ℓ rounds is $\frac{\ell\cdot(\ell-1)}{n\cdot(n-1)}$. Since the rounds in which types 1 and 2 arrive are drawn uniformly at random, this means that type 2 is observed prior to type 1 with a probability of $1/2$. Combining these probabilities gives us

$$\Pr[E] = \frac{1}{2}\cdot\left(1-\frac{\ell\cdot(\ell-1)}{n\cdot(n-1)}\right) = \frac{1}{2} - \frac{\ell\cdot(\ell-1)}{2n\cdot(n-1)} \quad .$$

Now, condition on event E occurring and type 2 being observed in round $i \le n-\ell$. Since type 2 does not get a recommendation, in round $i+1$ the set $\mathcal{C}$ of remaining types consists only of type 1 and a subset of types $3,\dots,n$ of size $n-i-1$. Clearly,

$$\frac{1}{|\mathcal{C}|-1}\cdot\sum_{t'\in\mathcal{C}\setminus\{1\}} \varrho_{t'} = \frac{1}{n-i-1}\cdot(n-i-1) = 1 > \frac{n-2}{n-1} = \varrho_1 \quad .$$

This means that $x_t^{\mathcal{C}} = 1$ for all $t \in \mathcal{C}$ in an optimal mechanism in round $i+1$ directly after type 2 was observed in round i. Hence, the process ends after round $i+1$. This means that type 1 is taken with probability only $1/|\mathcal{C}| \le 1/\ell$ since E requires $i \le n-\ell$. Overall, the expected utility of an optimal mechanism in the setting with disclosure can be upper bounded by

$$(1-\Pr[E])\cdot 1 + \Pr[E]\cdot\frac{1}{\ell} \le \frac{1}{2} + \frac{\ell\cdot(\ell-1)}{n\cdot(n-1)} + \frac{1}{2\ell} - \frac{\ell-1}{2n\cdot(n-1)} = \frac{1}{2} + o(1)$$

as $\ell = \sqrt{n}$. □

As our optimal mechanism requires solving an exponential number of linear programs, we now turn to another variant of the Pareto Mechanism. In every round, it considers the shrinking set of types and thus is named the *Shrinking Pareto Mechanism* (Algorithm 11). If and only if the type of the current action is the one computed by the Pareto Procedure on the set R_i of types remaining in round i, the mechanism sends a signal $\sigma_i =$ YES. As before, the same mechanism can be used for both cardinal as well as ordinal sender utility by adjusting the types' values.

In our analysis of the Shrinking Pareto Mechanism, we first show that it is persuasive in Lemma 4.17. Then, we show that it yields a $(1/3 - o(1))$-approximation for the case of cardinal sender utility (Theorem 4.18) and a ($1/2 - o(1)$)-approximation for ordinal sender utility (Theorem 4.19). Hence, for the latter case of ordinal sender utility, the Shrinking Pareto Mechanism is an asymptotically optimal mechanism by Theorem 4.16.

Lemma 4.17
For cardinal receiver utility, the Shrinking Pareto Mechanism is persuasive in the basic scenario with disclosure.

Proof. Recall that the Pareto Procedure for a set R of types chooses a type $c \in R$ such that $\mathbb{E}[\varrho(c)] \geq \varrho_E$, where $\varrho_E = \sum_{t \in R} \varrho_t$. We condition on the set R_i of types remaining at the start of round i. If the Shrinking Pareto Mechanism sends $\sigma_i = \text{YES}$ in round i, this means that the current action i has type $\theta_i = c$ as identified by the Pareto Procedure. Thus, we have

$$\mathbb{E}[\varrho(\theta_i) \mid \sigma_i = \text{YES} \wedge R_i] \geq \frac{1}{|R_i|} \sum_{t \in R_i} \varrho_t \ .$$

We show that the probabilities to send a signal $\sigma_i = \text{YES}$ for a type t conditioned on the remaining set of types being R_i and the current type being t satisfy the constraints of the linear programs identified in the proof of Theorem 4.14 to show persuasiveness. Hence, we use the following notation. Let $x_t^{R_i} = \Pr[\sigma_i = \text{YES} \mid R_i \wedge (\theta_i = t \in R_i)]$. The Pareto Procedure uses a convex combination of the (up to) two types a and b such that $\sum_{t \in R_i} x_t^{R_i} = 1$. Hence, we get

$$\begin{aligned}
\sum_{t \in R_i} x_t^{R_i} \cdot \varrho_t \geq \varrho_E & \\
&= \frac{1}{|R_i|} \cdot \sum_{t \in R_i} \varrho_t \\
&= \frac{1}{|R_i|} \cdot \sum_{t \in R_i} x_t^{R_i} \cdot \sum_{t \in R_i} \varrho_t \\
&= \frac{1}{|R_i|} \cdot \sum_{t \in R_i} x_t^{R_i} \cdot \sum_{t' \in R_i \setminus \{t\}} \varrho_{t'} + \frac{1}{|R_i|} \cdot \sum_{t \in R_i} x_t^{R_i} \cdot \varrho_t
\end{aligned}$$

and thus

$$\sum_{t \in R_i} x_t^{R_i} \cdot \varrho_t \geq \frac{1}{|R_i| - 1} \cdot \sum_{t \in R_i} x_t^{R_i} \cdot \sum_{t' \in R_i \setminus \{t\}} \varrho_{t'} \ .$$

Clearly, $x_t^{R_i} \in [0, 1]$ for all sets of remaining types R_i and all $t \in R_i$. Hence, $\boldsymbol{x}^{R_i}$, which consists of the signaling probabilities of the Shrinking Pareto Mechanism for R_i represents a feasible solution to the linear program (4.10). Since this is a sufficient condition for persuasiveness, the Shrinking Pareto Mechanism is persuasive. □

We continue with the approximation ratio for cardinal sender utility in the setting with disclosure.

Theorem 4.18
For cardinal sender and receiver utility in the basic scenario with disclosure, the Shrinking Pareto Mechanism scenario obtains a ($1/3 - o(1)$)-approximation of the optimum in the corresponding basic scenario.

Proof. We use the following notation. Similar to OPT, the optimal expected utility for $\mathcal{S}$ in the basic scenario without disclosure for the complete type set $[n]$, we denote by SP the expected utility obtained by the Shrinking Pareto Mechanism for set $[n]$. Since in each round i with $\sigma_i = \text{NO}$, some type θ_i is rejected and removed from the set of remaining types, we also define OPT_{-M} and SP_{-M} to denote the expected utility obtained from type set $[n] \setminus M$ by the Pareto Mechanism and by the Shrinking Pareto Mechanism, respectively.

In the first round, $\sigma_1 = \text{YES}$ with probability $1/n$ and $\mathbb{E}\left[\xi_{\theta_1} \mid \sigma_1 = \text{YES}\right] = \text{OPT}$. Otherwise, if $\sigma_1 = \text{NO}$, type θ_1 is rejected and $\mathcal{S}$ achieves an expected utility of $\text{SP}_{-\{\theta_1\}}$. Recursively, we can lower bound SP as follows.

$$\begin{aligned}
\text{SP} &= \frac{1}{n} \cdot \sum_{t \in [n]} \Pr[\theta_1 = t, \sigma_1 = \text{YES}] \cdot \xi_t + \Pr[\theta_1 = t, \sigma_1 = \text{NO}] \cdot \text{SP}_{-\{t\}} \\
&= \frac{\text{OPT}}{n} + \frac{1}{n} \left((1-\alpha) \cdot \text{SP}_{-\{a\}} + \alpha \cdot \text{SP}_{-\{b\}} + \sum_{t \neq a,b} \text{SP}_{-\{t\}} \right) ,
\end{aligned}$$

where a and b denote the types identified by the Pareto Procedure and α the corresponding probability that a is the type chosen by the Pareto Procedure. Clearly, $\text{SP}_{-\{a\}} \geq 0$ and $\text{SP}_{-\{b\}} \geq 0$. Hence, we can bound

$$\begin{aligned}
\text{SP} &\geq \frac{\text{OPT}}{n} + \frac{1}{n} \cdot \sum_{t \neq a,b} \text{SP}_{-\{t\}} \\
&\geq \frac{\text{OPT}}{n} + \frac{1}{n} \cdot \sum_{t \neq a,b} \left(\frac{1}{n-1} \cdot \sum_{t' \in [n] \setminus \{t\}} \Pr[\theta_2 = t', \sigma_2 = \text{YES}] \cdot \xi_{t'} \right. \\
&\qquad\qquad \left. + \Pr[\theta_2 = t', \sigma_2 = \text{NO}] \cdot \text{SP}_{-\{t,t'\}} \right) \\
&= \frac{\text{OPT}}{n} + \frac{1}{n \cdot (n-1)} \cdot \sum_{t \neq a,b} \text{OPT}_{-\{t\}} \\
&\qquad + \frac{1}{n \cdot (n-1)} \cdot \sum_{t \neq a,b} \sum_{t' \in [n] \setminus \{t\}} \Pr[\theta_2 = t', \sigma_2 = \text{NO}] \cdot \text{SP}_{-\{t,t'\}} \\
&\geq \frac{\text{OPT}}{n} + \sum_{t \neq a,b} \frac{\text{OPT}_{-\{t\}}}{n \cdot (n-1)} + \sum_{t \neq a,b} \sum_{\substack{t' \in [n] \setminus \{t\} \\ t' \neq a_{-t}, b_{-t}}} \frac{\text{OPT}_{-\{t,t'\}}}{n \cdot (n-1) \cdot (n-2)} + \ldots
\end{aligned}$$

where we use a_{-t} and b_{-t} to denote the types identified by the Pareto Procedure for type set $[n] \setminus \{t\}$. In Lemma 4.9, we showed that $\sum_{t \neq a,b} \text{OPT}_{-\{t\}} \geq (n-3) \cdot \text{OPT}$. Using this inequality, we continue bounding

$$\begin{aligned}
\text{SP} &\geq \frac{\text{OPT}}{n} + \sum_{t \neq a,b} \frac{\text{OPT}_{-\{t\}}}{n \cdot (n-1)} + \sum_{t \in [n] \setminus \{a,b\}} \sum_{\substack{t' \in [n] \setminus \{t\} \\ t' \neq a_{-t}, b_{-t}}} \frac{\text{OPT}_{-\{t,t'\}}}{n \cdot (n-1) \cdot (n-2)} + \ldots \\
&\geq \frac{\text{OPT}}{n} + \frac{n-3}{n \cdot (n-1)} \text{OPT} + \sum_{t \in [n] \setminus \{a,b\}} \frac{n-4}{n \cdot (n-1) \cdot (n-2)} \text{OPT}_{-\{t\}} + \ldots \\
&\geq \text{OPT} \cdot \frac{1}{n \cdot (n-1) \cdot (n-2)} \cdot \left(\sum_{\ell=1}^{n-3} (n-\ell) \cdot (n-\ell-1) \right)
\end{aligned}$$

$$
\begin{aligned}
&= \mathrm{OPT} \cdot \frac{1}{3} \cdot \frac{n^3 - 3n^2 + 2n - 6}{n^3 - 3n^2 + 2n} \\
&= \mathrm{OPT} \cdot \left(\frac{1}{3} - \frac{2}{n^3 - 3n^2 + 2n} \right) \quad .
\end{aligned}
$$

Hence, the Shrinking Pareto Mechanism achieves an expected utility of at least $1/3 - o(1)$ of the expected utility obtained by the Pareto Mechanism in the corresponding instance without disclosure. □

We can improve this bound for the case of ordinal sender utility to $1/2 - o(1)$, which asymptotically matches the upper bound of Theorem 4.16. The proof is similar to the one for cardinal sender utility. We use Lemma 4.12, which improves the approximation guarantee of Lemma 4.9 used in the previous proof.

Theorem 4.19
For ordinal sender and cardinal receiver utility in the basic scenario with disclosure, the Shrinking Pareto Mechanism yields a success probability of at least $(1/2 - o(1))$ times the optimal success probability in the corresponding basic scenario.

Proof. Again, we denote by OPT the success probability of $\mathcal{S}$ that the action taken has type $c_{\mathcal{S}}$ when using the Pareto Mechanism and similarly, we use SP to denote the success probability from the Shrinking Pareto Mechanism. By OPT_{-M} and SP_{-M}, we denote the respective success probabilities for the subset of types $[n] \setminus M$. Recall that the Pareto Procedure returns a single type c. To this end, the procedure identifies two types a and b. For the ordinal sender case, we have $a = c_{\mathcal{S}}$ and $b = c_{\mathcal{R}}$. For these choices, the procedure determines a probability α such that $c = a$ with probability α and $c = b$ with probability $1 - \alpha$.

In the first round, the probability that $\theta_1 = c_{\mathcal{S}}$ is $1/n$ and conditional on that, $\sigma_1 = \mathrm{YES}$ with probability $\alpha = \mathrm{OPT}$. With probability $1 - \mathrm{OPT}$, $\sigma_1 = \mathrm{NO}$ is sent and no utility can be extracted any more. Analogously, if $\theta_1 = b$, with probability OPT a signal NO is sent and YES is sent with probability $1 - \mathrm{OPT}$. For all other types $t \neq a, b$, NO is signaled with probability 1.

We use $c'_{\mathcal{R}}$ and $c''_{\mathcal{R}}$ to denote the type with the second and third highest utility for $\mathcal{R}$, respectively. We get

$$
\begin{aligned}
\mathrm{SP} &= \frac{1}{n} \cdot \mathrm{OPT} + \frac{1}{n} \cdot \mathrm{OPT} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}\}} + \frac{1}{n} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{SP}_{-\{t\}} \\
&= \frac{\mathrm{OPT}}{n} + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \left[\mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \cdot \left(1 + \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}}\right) + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}} \right] \\
&\quad + \frac{1}{n \cdot (n-1)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \left[\mathrm{OPT}_{-\{t\}} \cdot \left(1 + \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}}\right) + \sum_{t' \neq c_{\mathcal{S}}, c_{\mathcal{R}}, t} \mathrm{SP}_{-\{t, t'\}} \right] \\
&= \frac{\mathrm{OPT}}{n} + \underbrace{\frac{\mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}}}{n \cdot (n-1)} + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \frac{\mathrm{OPT}_{-\{t\}}}{n \cdot (n-1)}}_{\geq \frac{1}{n \cdot (n-1)} \cdot \mathrm{OPT} \cdot \left(n - 2 - \frac{1}{n-1}\right)} \\
&\quad + \frac{\mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}}}{n \cdot (n-1)} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}}
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \frac{1}{n \cdot (n-1)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \left[\mathrm{OPT}_{-\{t\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}} + \sum_{t' \neq c_{\mathcal{S}}, c_{\mathcal{R}}, t} \mathrm{SP}_{-\{t, t'\}} \right] \\
&\geq \frac{\mathrm{OPT}}{n} + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \left(n - 2 - \frac{1}{n-1} \right) + \frac{\mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}}}{n \cdot (n-1)} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} \\
&\quad + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}} \\
&\quad + \frac{1}{n \cdot (n-1)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \left[\mathrm{OPT}_{-\{t\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, t\}} + \sum_{t' \neq c_{\mathcal{S}}, c_{\mathcal{R}}, t} \mathrm{SP}_{-\{t, t'\}} \right] \\
&= \frac{\mathrm{OPT}}{n} + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \left(n - 2 - \frac{1}{n-1} \right) \\
&\quad + \frac{\mathrm{OPT} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}}}{n \cdot (n-1) \cdot (n-2)} \cdot \Bigg[\mathrm{OPT}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} + \mathrm{OPT}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, c''_{\mathcal{R}}\}} \\
&\qquad + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}, c''_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, t\}} \Bigg] \\
&\quad + \frac{\mathrm{OPT}}{n \cdot (n-1) \cdot (n-2)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \Bigg[\mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} \\
&\qquad + \mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, t\}} + \sum_{t' \neq c_{\mathcal{S}}, c_{\mathcal{R}}, t, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t, t'\}} \Bigg] \\
&\quad + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \frac{\mathrm{OPT}_{-\{t\}}}{n \cdot (n-1) \cdot (n-2)} \cdot \Bigg[\mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} + \mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, t\}} \\
&\qquad + \sum_{t' \neq t, c_{\mathcal{R}}, c_{\mathcal{S}}, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t, t'\}} \Bigg] \\
&\quad + \frac{1}{n \cdot (n-1) \cdot (n-2)} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}} \sum_{t' \neq c_{\mathcal{S}}, c_{\mathcal{R}}, t} \Bigg[\mathrm{OPT}_{-\{t, t'\}} \\
&\qquad + \mathrm{OPT}_{-\{t, t'\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, t, t'\}} + \sum_{t'' \neq t, t', c_{\mathcal{S}}, c_{\mathcal{R}}} \mathrm{SP}_{-\{t, t', t''\}} \Bigg] \\
&= \frac{\mathrm{OPT}}{n} + \frac{\mathrm{OPT}}{n \cdot (n-1)} \cdot \left(n - 2 - \frac{1}{n-1} \right) \\
&\quad + \frac{\mathrm{OPT}}{n \cdot (n-1) \cdot (n-2)} \cdot \Bigg[\mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} \\
&\qquad + \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, c''_{\mathcal{R}}\}} \\
&\qquad + \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} \cdot \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}, c''_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, t\}} \\
&\qquad + \sum_{t \neq c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \left(\mathrm{OPT}_{-\{c_{\mathcal{R}}, t\}} \cdot \mathrm{SP}_{-\{c_{\mathcal{R}}, c'_{\mathcal{R}}, t\}} + \sum_{t' \neq t, c_{\mathcal{S}}, c_{\mathcal{R}}, c'_{\mathcal{R}}} \mathrm{SP}_{-\{c_{\mathcal{R}}, t, t'\}} \right) \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{1}{n\cdot(n-1)\cdot(n-2)} \cdot \sum_{t\neq c_{\mathcal{S}},c_{\mathcal{R}}} \Bigg[\mathrm{OPT}_{-\{t\}}\cdot \mathrm{OPT}_{-\{c_{\mathcal{R}},t\}} + \sum_{t'\neq c_{\mathcal{S}},c_{\mathcal{R}},t} \mathrm{OPT}_{-\{t,t'\}} \\
&\qquad + \mathrm{OPT}_{-\{t\}}\cdot \mathrm{OPT}_{-\{c_{\mathcal{R}},t\}}\cdot \mathrm{SP}_{-\{c_{\mathcal{R}},c'_{\mathcal{R}},t\}} + \mathrm{OPT}_{-\{t\}}\cdot \sum_{t'\neq c_{\mathcal{S}},c_{\mathcal{R}},c'_{\mathcal{R}},t} \mathrm{SP}_{-\{c_{\mathcal{R}},t,t'\}} \\
&\qquad + \sum_{t'\neq c_{\mathcal{S}},c_{\mathcal{R}},t} \Bigg(\mathrm{OPT}_{-\{t,t'\}}\cdot \mathrm{SP}_{-\{c_{\mathcal{R}},t,t'\}} + \sum_{t''\neq t,t',c_{\mathcal{S}},c_{\mathcal{R}}} \mathrm{SP}_{-\{t,t',t''\}} \Bigg) \Bigg] \\
\geq\ & \frac{\mathrm{OPT}}{n} + \frac{\mathrm{OPT}}{n\cdot(n-1)}\cdot\left(n-2-\frac{1}{n-1}\right) \\
&+ \frac{n-3-\frac{1}{n-2}}{n\cdot(n-1)\cdot(n-2)}\cdot\left(\mathrm{OPT}\cdot \mathrm{OPT}_{-\{c_{\mathcal{R}}\}} + \sum_{t\neq c_{\mathcal{S}},c_{\mathcal{R}}} \mathrm{OPT}_{-\{t\}}\right) + \ldots \\
\geq\ & \frac{\mathrm{OPT}}{n}\cdot\left[1 + \frac{n-2-\frac{1}{n-1}}{n-1} + \frac{\left(n-2-\frac{1}{n-1}\right)\cdot\left(n-3-\frac{1}{n-2}\right)}{(n-1)\cdot(n-2)} + \ldots\right] \\
=\ & \frac{\mathrm{OPT}}{n}\cdot\sum_{\ell=0}^{n-2}\prod_{j=1}^{\ell}\frac{n-j-1-\frac{1}{n-j}}{n-j} \\
\geq\ & \frac{\mathrm{OPT}}{n}\cdot\sum_{\ell=0}^{n-2}\prod_{j=1}^{\ell}\frac{n-j-2}{n-j-1} \\
=\ & \frac{\mathrm{OPT}}{n}\cdot\sum_{\ell=0}^{n-2}\frac{n-2-\ell}{n-2} \\
=\ & \frac{\mathrm{OPT}}{n}\cdot\left[(n-1) - \frac{(n-1)\cdot(n-2)}{2(n-2)}\right] \\
=\ & \mathrm{OPT}\cdot\left(\frac{1}{2}-\frac{1}{2n}\right)\ .
\end{aligned}
$$

Thus, using the Shrinking Pareto Mechanism, $\mathcal{S}$ has a success probability of at least $\left(\frac{1}{2}-\frac{1}{2n}\right)\cdot \mathrm{OPT}$. This completes the proof. □

As our final setting for cardinal receiver utility, we will consider the secretary scenario with disclosure next.

Secretary Scenario with Disclosure

For the secretary scenario with disclosure of dismissed types, neither $\mathcal{S}$ nor $\mathcal{R}$ have any information on the valuation pairs a priori. As in the previous section on the basic scenario with disclosure, types of dismissed actions are revealed to $\mathcal{R}$ before $\mathcal{S}$ learns the type of the next action. Recall that our mechanisms are persuasive even if $\mathcal{R}$ were to know the valuation pairs. Hence, for our analysis of persuasiveness, we assume that $\mathcal{R}$ has information on the types.

We will show that $\mathcal{S}$ cannot achieve a success probability of more than $^2/_n\cdot\mathrm{OPT}$, i.e., only an $O(^1/_n)$-approximation of the optimal success probability in the corresponding basic scenario without disclosure.

Since the following *Trivial Mechanism* already provides a success probability of $^1/_n$ and an expected utility of $^1/_n\cdot\xi_{max}\geq {^1/_n}\cdot\mathrm{OPT}$, this already shows a matching lower bound of $\Omega(^1/_n)$ for the approximation ratio. The Trivial Mechanism always sends a

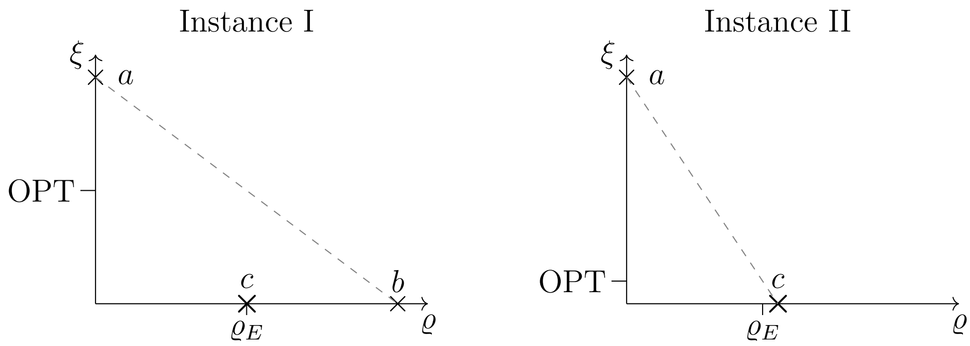


Figure 14: Instances I and II for the proof of Theorem 4.20

recommendation in the first round, i.e., $\sigma_1 = \text{YES}$, $\sigma_i = \text{NO}$ for $i = 2, \ldots, n$. Clearly, this mechanism is persuasive as $\mathcal{R}$ does not get any information. Hence, a deviation to a different action does not increase the receiver's expected utility or success probability, respectively.

We first show the bound for the ordinal sender case in Theorem 4.20.

Theorem 4.20
For ordinal sender and cardinal receiver utility in the secretary scenario with disclosure, there is no persuasive mechanism that guarantees $\mathcal{S}$ a success probability greater than $^2/_n \cdot \text{OPT}$, where OPT *is the success probability in the corresponding basic instance.*

Clearly, interpreting the same instance for a sender with cardinal utility, i.e., setting $\xi_{max} = 1$ and $\xi_t = 0$ for all $t \neq c_{\mathcal{S}}$ shows the corresponding bound in Corollary 4.21.

Corollary 4.21
For cardinal sender and receiver utility in the secretary scenario with disclosure, there is no persuasive mechanism that guarantees $\mathcal{S}$ an expected utility of more than $\frac{2}{n} \cdot \text{OPT}$, where OPT *is the optimal expected utility in the corresponding basic instance.*

To prove the theorem, we use two different instances with n types. Clearly, this already gives $\mathcal{S}$ a lot of knowledge about the types. Still, the two different instances suffice to show the bound on the approximation ratio of $O(^1/_n)$.

In both instances, there is a type a with value-pair $(\varrho_a, \xi_a) = (0, 1)$ and this is the best type for $\mathcal{S}$ and the only one with positive utility for $\mathcal{S}$. The best type for $\mathcal{R}$ in instance I is type b with value-pair $(\varrho_b, \xi_b) = (1, 0)$. The remaining $n - 2$ types in instance I are all the same. Hence, we refer to them as type c. All have the same value-pair of $(\varrho_c, \xi_c) = (^1/_2, 0)$. In instance II, there are $n - 1$ copies of type c instead. Hence, the instances only differ by a single type. In instance I, there is a type b which does not exist in instance II. Here, it is replaced by another copy of type c. An illustration of the instances is given in Figure 14.

Consider an arbitrary direct and persuasive mechanism. Since the mechanism is required to be persuasive even if $\mathcal{R}$ were to know the underlying instance a priori, let us assume that this is the case. In round $i = 1$, all types can potentially be revealed as $\mathcal{S}$ does not know whether the instance is I or II. We denote the probabilities for a signal $\sigma_1 = \text{YES}$ by p_a^1, p_b^1, and p_c^1, depending on the type which was revealed.

Assuming that in the first round, the signal was $\sigma_1 = \text{NO}$ and thus $\mathcal{R}$ did not take the action, in the second round $i = 2$, the set of types that can be revealed depends on θ_1, the type of the first round. For example, if $\theta_1 = a$, then $\theta_2 = a$ is not possible. The following table displays the possible scenarios.

θ_2	a	a	b	b	c	c	c
θ_1	c	b	c	a	c	a	b
$\Pr[\sigma_2 = \text{YES} \mid \theta_1, \theta_2]$	$p^2_{a,c}$	$p^2_{a,b}$	$p^2_{b,c}$	$p^2_{b,a}$	$p^2_{c,c}$	$p^2_{c,a}$	$p^2_{c,b}$

Observe that $p^2_{t,t'}$ is the probability to signal YES in round 2 if $\theta_2 = t$ and $\theta_1 = t'$.

We subsume the rounds $i = 3, \dots, n-1$ using variables $p^i_{t,t'}$ for $t, t' \in \{a, b, c\}$. Again, we assume all previous signals were NO, i.e., $\sigma_1, \dots, \sigma_{i-1} = \text{NO}$. We denote the set of types observed in rounds $1, \dots, i-1$ by A_{i-1}. There are 8 possible cases. For short, we only write c if type c has been observed multiple times.

θ_i	a	a	b	b	c	c	c	c
A_{i-1}	c	b, c	c	a, c	c	a, c	b, c	a, b, c
$\Pr[\sigma_i = \text{YES} \mid A_{i-1}, \theta_i]$	$p^i_{a,c}$	$p^i_{a,b}$	$p^i_{b,c}$	$p^i_{b,a}$	$p^i_{c,c}$	$p^i_{c,a}$	$p^i_{c,b}$	1

Clearly, for $i = 3$, the final case is only a and b having arrived and having been rejected. Additionally, in the final case, if a and b have been observed (and have been dismissed), both $\mathcal{S}$ and $\mathcal{R}$ know that only actions of type c will be revealed in future rounds. Hence, without loss of generality, the signal will be $\sigma_i = \text{YES}$ immediately.

In round $i = n$, if all previous signals have been NO, a direct mechanism always signals YES with probability 1.

In the following lemma, we identify properties that every persuasive mechanism has to satisfy. We will then be able to prove Theorem 4.20 using these insights.

Lemma 4.22
For every mechanism that is persuasive in both instances I and II, it must hold for round $i = 1$

$$p^1_b \geq p^1_a \qquad \textit{and} \qquad p^1_c \geq p^1_a$$

and for every round $i = 2, \dots, n-1$

$$p^i_{b,c} \geq p^i_{a,c} \ , \qquad p^i_{b,a} \geq p^i_{c,a} \ , \qquad p^i_{c,b} \geq p^i_{a,b} \qquad \textit{and} \qquad p^i_{c,c} \geq p^i_{a,c} \ .$$

These conditions imply that a persuasive mechanism cannot be more likely to send a signal YES for type a rather than type b or c. Note that the lemma only has restrictions for rounds $i = 1, \dots, n-1$. Technically, the statement also holds for round $i = n$, as in that round, the probability for a signal $\sigma_n = \text{YES}$ is 1 for any type, assuming $\sigma_i = \text{NO}$ for all previous rounds.

Proof. We condition on a signal $\sigma_i = \text{YES}$. Persuasiveness dictates that the expectation for $\mathcal{R}$ in round i is at least the expectation for $\mathcal{R}$ in round $i+1$, i.e.,

$$\mathbb{E}\left[\varrho(\theta_i) \mid \sigma_i = \text{YES}\right] \geq \mathbb{E}\left[\varrho(\theta_{i+1}) \mid \sigma_i = \text{YES}\right] \ . \tag{4.11}$$

Only if this inequality is satisfied does $\mathcal{R}$ have an incentive to follow the signal and take action i.

We begin with round $i = 1$:

Instance I: Clearly,

$$\mathbb{E}\left[\varrho(\theta_1) \mid \sigma_1 = \text{YES}\right] = p_a^1 \cdot 0 + p_b^1 \cdot 1 + (n-2) \cdot p_c^1 \cdot \frac{1}{2}$$

and

$$\begin{aligned}\mathbb{E}\left[\varrho(\theta_2) \mid \sigma_1 = \text{YES}\right] &= p_a^1 \cdot \frac{\frac{n-2}{2}+1}{n-1} + p_b^1 \cdot \frac{\frac{n-2}{2}}{n-1} + (n-2) \cdot p_c^1 \cdot \frac{\frac{n-3}{2}+1}{n-1} \\ &= p_a^1 \cdot \frac{n}{2(n-1)} + p_b^1 \cdot \frac{n-2}{2(n-1)} + p_c^1 \cdot \frac{n-2}{2} \quad .\end{aligned}$$

Hence, inequality (4.11) implies $p_b^1 \geq p_a^1$.

Instance II: In this instance, type b does not exist. Thus, we get

$$\mathbb{E}\left[\varrho(\theta_1) \mid \sigma_1 = \text{YES}\right] = p_a^1 \cdot 0 + (n-1) \cdot p_c^1 \cdot \frac{1}{2}$$

and

$$\begin{aligned}\mathbb{E}\left[\varrho(\theta_2) \mid \sigma_1 = \text{YES}\right] &= p_a^1 \cdot \frac{(n-1) \cdot \frac{1}{2}}{n-1} + (n-1) \cdot p_c^1 \cdot \frac{\frac{n-2}{2}}{n-1} \\ &= p_a^1 \cdot \frac{1}{2} + p_c^1 \cdot \frac{n-2}{2} \quad .\end{aligned}$$

Clearly, inequality (4.11) implies $p_c^1 \geq p_a^1$.

Now, consider rounds $i = 2, \ldots, n-1$. We begin with instance I.

Instance I, A_{i-1} contains only c: We have

$$\mathbb{E}\left[\varrho(\theta_i) \mid \sigma_i = \text{YES}\right] = p_{a,c}^i \cdot 0 + p_{b,c}^i + (n-i-1) \cdot p_{c,c}^i \cdot \frac{1}{2}$$

and

$$\begin{aligned}&\mathbb{E}\left[\varrho(\theta_{i+1}) \mid \sigma_i = \text{YES}\right] \\ &\quad = p_{a,c}^i \cdot \frac{\frac{n-i-1}{2}+1}{n-i} + p_{b,c}^i \cdot \frac{\frac{n-i-1}{2}}{n-i} + (n-i-1) \cdot p_{c,c}^i \cdot \frac{\frac{n-i-2}{2}+1}{n-i} \\ &\quad = p_{a,c}^i \cdot \frac{n-i+1}{2(n-i)} + p_{b,c}^i \cdot \frac{n-i-1}{2(n-i)} + p_{c,c}^i \cdot \frac{n-i-1}{2}\end{aligned}$$

Inequality (4.11) implies $p_{b,c}^i \geq p_{a,c}^i$.

Instance I, A_{i-1} contains a, not b: We have

$$\mathbb{E}\left[\varrho(\theta_i) \mid \sigma_i = \text{YES}\right] = p_{b,a}^i \cdot 1 + p_{c,a}^i \cdot (n-i) \cdot \frac{1}{2}$$

and

$$\begin{aligned}\mathbb{E}\left[\varrho(\theta_{i+1}) \mid \sigma_i = \text{YES}\right] &= p_{b,a}^i \cdot \frac{(n-i) \cdot \frac{1}{2}}{n-i} + (n-i) \cdot p_{c,a}^i \cdot \frac{\frac{n-i-1}{2}+1}{n-i} \\ &= p_{b,a}^i \cdot \frac{1}{2} + p_{c,a}^i \cdot \frac{n-i+1}{2}\end{aligned}$$

Inequality (4.11) implies $p_{b,a}^i \geq p_{c,a}^i$.

Instance I, A_{i-1} contains b, not a: We have

$$\mathbb{E}\left[\varrho(\theta_i) \mid \sigma_i = \text{YES}\right] = p^i_{a,b} \cdot 0 + p^i_{c,b} \cdot (n-i) \cdot \frac{1}{2}$$

and

$$\begin{aligned}\mathbb{E}\left[\varrho(\theta_{i+1}) \mid \sigma_i = \text{YES}\right] &= p^i_{a,b} \cdot \frac{(n-i)\cdot\frac{1}{2}}{n-i} + (n-i)\cdot p^i_{c,b} \cdot \frac{\frac{n-i-1}{2}}{n-i} \\ &= p^i_{a,b} \cdot \frac{1}{2} + p^i_{c,b} \cdot \frac{n-i-1}{2}\end{aligned}$$

Inequality (4.11) implies $p^i_{c,b} \geq p^i_{a,b}$.

Instance I, A_{i-1} contains a and b: For $i = 2$ this case does not occur. For $i \geq 3$, $\mathcal{S}$ and $\mathcal{R}$ both know that they are in instance I and only type c is left. Hence, any scheme that guarantees a signal YES in one of the remaining rounds is persuasive. Clearly, the expectation for $\mathcal{R}$ is $1/2$, regardless of the round $i' \geq i$. As stated in the table above, without loss of generality, sending a YES-recommendation in round i with probability 1 is persuasive.

These are all possible cases for instance I. We continue with instance II which does not have a type b.

Instance II, A_{i-1} contains only c: We have

$$\mathbb{E}\left[\varrho(\theta_i) \mid \sigma_i = \text{YES}\right] = p^i_{a,c} \cdot 0 + p^i_{c,c} \cdot (n-i) \cdot \frac{1}{2}$$

and

$$\begin{aligned}\mathbb{E}\left[\varrho(\theta_{i+1}) \mid \sigma_i = \text{YES}\right] &= p^i_{a,c} \cdot \frac{(n-i)\cdot\frac{1}{2}}{n-i} + (n-i)\cdot p^i_{c,c} \cdot \frac{\frac{n-i-1}{2}}{n-i} \\ &= p^i_{a,c} \cdot \frac{1}{2} + p^i_{c,c} \cdot \frac{n-i-1}{2}\end{aligned}$$

Inequality (4.11) implies $p^i_{c,c} \geq p^i_{a,c}$.

Instance II, A_{i-1} contains a: Since the scheme is persuasive even if $\mathcal{R}$ were to know the types a priori, we can assume that $\mathcal{R}$ knows that the remaining actions have type c. As long as $\mathcal{R}$ knows that there will be a YES-signal in one of the remaining rounds, the scheme is persuasive. The expected utility for $\mathcal{R}$ is $\frac{1}{2}$, regardless of the round in which the action is taken.

Overall, we have shown the lemma using the implications of the persuasiveness constraints of the different compositions of A_{i-1} for instances I and II. $\square$

Using these constraints, we are now able to prove Theorem 4.20. We show that any mechanism that satisfies the constraints of the previous lemma in instance I cannot get a better success probability than $1/n$ and OPT $= 1/2$. Hence, the approximation ratio is $2/n$.

Proof of Theorem 4.20. Using the Pareto Mechanism for instance I in the basic scenario without disclosure, $\mathcal{S}$ has a success probability of $1/2$. With probability $1/2$, type a is recommended, and with probability $1/2$, type b is recommended. Clearly, this gives $\mathcal{R}$ an expected utility of $\varrho_E = 1/2$.

For persuasiveness, $\mathcal{S}$ must assume that $\mathcal{R}$ knows whether it is instance I or II. Hence, sending a signal YES on types a and b with probability $1/2$ each cannot be persuasive – $\mathcal{R}$ would never follow the recommendations in instance II. Clearly, this is shown by the constraints identified in Lemma 4.22, which state that a signal YES upon seeing type a cannot be more likely than a signal YES upon seeing b or c, respectively. We formalize this by showing that there exists an optimal persuasive mechanism which always sends a YES-recommendation in the first round, i.e., the above Trivial Mechanism which always sends $\sigma_1 = \text{YES}$ is optimal. To this end, we will use a backwards induction showing that when reaching round $i \in [n]$ without a previous YES-signal, it is optimal for $\mathcal{S}$ to signal $\sigma_i = \text{YES}$.

Clearly, for $i = n$, an optimal persuasive mechanism sends a signal $\sigma_n = \text{YES}$ if $\sigma_{i'} = \text{NO}$ for all $i' < n$. Now, suppose the inductive assumption holds for rounds $i+1, \ldots, n$. We consider round $i \geq 2$ and distinguish the different compositions of A_{i-1}.

A_{i-1} **contains only** c**:** If action i has type a, the success probability is $p^i_{a,c}$. If type b is revealed, $\mathcal{S}$ has a success probability of $(1 - p^i_{b,c}) \cdot \frac{1}{n-i}$ since the inductive assumption is that $\sigma_{i+1} = \text{YES}$ if $\sigma_1, \ldots, \sigma_i = \text{NO}$. The probability that $\theta_{i+1} = a$ is $\frac{1}{n-i}$ due to the uniform random order. Similarly, if action i has type c, the success probability is $(1 - p^i_{c,c}) \cdot \frac{1}{n-i}$. This means that $\mathcal{S}$ wants to maximize

$$p^i_{a,c} + \frac{1 - p^i_{b,c}}{n-i} + (n-i-1) \cdot \frac{1 - p^i_{c,c}}{n-i}$$

subject to the constraints $p^i_{b,c} \geq p^i_{a,c}$ and $p^i_{c,c} \geq p^i_{a,c}$ as identified in Lemma 4.22. Clearly, making these constraints tight by minimizing both $p^i_{b,c}$ and $p^i_{c,c}$, i.e., setting $p^i_{b,c} = p^i_{a,c}$ and $p^i_{c,c} = p^i_{a,c}$, maximizes the expression. Thus, we can rewrite it as

$$p^i_{a,c} + \frac{1 - p^i_{a,c}}{n-i} + (n-i-1) \cdot \frac{1 - p^i_{a,c}}{n-i} = 1 \ .$$

Hence, if $p^i_{a,c} = p^i_{b,c} = p^i_{c,c}$, the expression simplifies to a constant which is independent of the probabilities to send a signal YES. Clearly, this means that $p^i_{a,c} = p^i_{b,c} = p^i_{c,c} = 1$ is a feasible and optimal choice, which means that $\sigma_i = \text{YES}$ if $\sigma_1, \ldots, \sigma_{i-1} = \text{NO}$ for the set A_{i-1} is optimal and persuasive.

A_{i-1} **contains** a**, not** b**:** If $\mathcal{S}$ has observed type a and the action was not taken, the success probability is clearly 0. Hence, setting $p_{b,a} = p_{c,a} = 1$ is feasible and an optimal choice for $\mathcal{S}$. This means that $\sigma_i = \text{YES}$ if $\sigma_1, \ldots, \sigma_{i-1} = \text{NO}$ for the set A_{i-1} is optimal and persuasive.

A_{i-1} **contains** b**, not** a**:** If action i has type a, the success probability is $p^i_{a,b}$. Type b has already been observed and cannot arrive anymore. If type c is revealed in round i, $\mathcal{S}$ has a success probability of $(1 - p^i_{c,b}) \cdot \frac{1}{n-i}$, following the same argument as in the first composition of A_{i-1}. An optimal mechanism thus wants to maximize

$$p_{a,b}^i + (n-i)\cdot\frac{1-p_{c,b}^i}{n-i} = p_{a,b}^i + 1 - p_{c,b}^i$$

subject to the constraint $p_{c,b}^i \geq p_{a,b}^i$ as identified in Lemma 4.22. Again, tightening the constraint by minimizing $p_{c,b}^i$ and thus setting $p_{c,b}^i = p_{a,b}^i$ maximizes the expression. We can rewrite it as

$$p_{a,b}^i + 1 - p_{a,b}^i = 1 \ .$$

Again, the expression simplifies to a constant independent of the probability values $p_{a,b}^i = p_{c,b}^i$. This clearly gives us a feasible and optimal choice of $p_{a,b}^i = p_{c,b}^i = 1$, which means that σ_i = YES if $\sigma_1, \ldots, \sigma_{i-1}$ = NO for the set A_{i-1} is optimal and persuasive.

A_{i-1} **contains** a **and** b**:** If a has arrived and been rejected, the success probability is 0. We already observed in the second case above that sending a recommendation σ_i = YES if $\sigma_1, \ldots, \sigma_{i-1}$ = NO for the set A_{i-1} is persuasive and optimal for $\mathcal{S}$.

This only leaves round $i = 1$. Clearly, if type a is observed, the success probability is p_a^1 and if action 1 has type b or type c, the success probabilities are $(1-p_b^1)\cdot\frac{1}{n-1}$ and $(1-p_c^1)\cdot\frac{1}{n-1}$, respectively.

This means that $\mathcal{S}$ wants to maximize

$$p_a^1 + \frac{1-p_b^1}{n-1} + (n-2)\cdot\frac{1-p_c^1}{n-1}$$

subject to the constraints $p_b^1 \geq p_a^1$ and $p_c^1 \geq p_a^1$ of Lemma 4.22. Again, the expression is maximized if $p_b^1 = p_a^1$ and $p_c^1 = p_a^1$. We can rewrite it as

$$p_a^1 + \frac{1-p_a^1}{n-1} + (n-2)\cdot\frac{1-p_a^1}{n-1} = 1 \ .$$

Since the expression simplifies to a constant and is therefore independent of the values $p_a^1 = p_b^1 = p_c^1$, we can set $p_a^1 = p_b^1 = p_c^1 = 1$. This means that setting σ_1 = YES gives an optimal mechanism.

The mechanism provides a success probability of $^1/_n$ for $\mathcal{S}$, which is a $^2/_n$-approximation to the optimal success probability in the corresponding basic scenario without disclosure. This proves the theorem. $\square_{\text{Theorem 4.20}}$

This concludes the discussion of cardinal receiver utility. In the next section, we discuss the results summarized in Table 4 for ordinal receiver utility.

4.2.2 Ordinal Utility for $\mathcal{R}$

In this section, we discuss ordinal utility for $\mathcal{R}$, i.e., $\mathcal{R}$ is only interested in taking the action with type $c_{\mathcal{R}}$. As in the previous section on cardinal utility for $\mathcal{R}$, we subdivide the section into the 4 different settings. We start by analyzing the basic scenario without disclosure as our benchmark case. Then, we discuss the secretary scenario without disclosure and finally the two disclosure scenarios, the basic as well as the secretary one.

Algorithm 12: Elementary Mechanism

Input: Set of valuation pairs $(\varrho_t, \xi_t)_{t\in[n]}$, online sequence of types $\theta_1, \ldots, \theta_n$

```
Draw x ~ Unif[0, 1].
if x ≤ 1/n then Set c = c_R.
else Set c = c_S.
for round i = 1, ..., n do
    if c = θ_i then Signal YES.
    else Signal NO.
```

Benchmark: Basic Scenario without Disclosure

In the basic scenario without disclosure, which serves as the benchmark for the other settings, both $\mathcal{S}$ and $\mathcal{R}$ know the types a priori. As in the corresponding setting for cardinal receiver utility, this means that $\mathcal{S}$ essentially faces an offline problem. Since $\mathcal{R}$ only extracts utility from $c_{\mathcal{R}}$ and $\mathcal{S}$ knows $c_{\mathcal{S}}$ as well as $c_{\mathcal{R}}$, an optimal mechanism will only send a YES-signal in a round i with $\theta_i = c_{\mathcal{S}}$ or $\theta_i = c_{\mathcal{R}}$. The *Elementary Mechanism* (Algorithm 12) decides before the actions are revealed whether to send YES for type $c_{\mathcal{S}}$ or $c_{\mathcal{R}}$ and then simply waits for the round. The mechanism takes advantage of the fact that $\mathcal{R}$ does not know the order of the types and hence only has a probability of finding type $c_{\mathcal{R}}$ of $1/n$.

We show that the Elementary Mechanism is persuasive. Clearly, it also guarantees a success probability of $1 - o(1)$ and an expected utility of $\xi_{max} \cdot (1 - o(1))$ to $\mathcal{S}$.

Proposition 4.23
For both cardinal and ordinal sender utility and ordinal receiver utility, the Elementary Mechanism is persuasive in the basic scenario without disclosure. It yields a success probability of $(1 - o(1))$ *and an expected utility of* $(1 - o(1)) \cdot \xi_{\max}$ *for* $\mathcal{S}$.

Proof. The mechanism chooses $c = c_{\mathcal{S}}$ with a probability of $1 - 1/n = 1 - o(1)$. If it is persuasive and $\mathcal{R}$ follows the signal, this clearly implies that with probability $1 - o(1)$, the action $\mathcal{R}$ takes has type $c_{\mathcal{S}}$.

Let us now show that the mechanism is indeed persuasive. Recall that the setting is without disclosure, which means that in round i, $\mathcal{R}$ only knows the signals $\sigma_1, \ldots, \sigma_i$. Clearly, the mechanism only sends a single YES-signal.

Assume that the mechanism has reached round i and $\sigma_i = \text{YES}$. Using Bayes' law, the success probability for $\mathcal{R}$ when taking the action in round i is

$$\begin{aligned}
\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] &= \frac{\Pr[\sigma_i = \text{YES} \mid \theta_i = c_{\mathcal{R}}] \cdot \Pr[\theta_i = c_{\mathcal{R}}]}{\Pr[\sigma_i = \text{YES}]} \\
&= \frac{\frac{1}{n} \cdot \frac{1}{n}}{\frac{1}{n} \cdot \left(\frac{1}{n} + \frac{n-1}{n}\right)} = \frac{1}{n} \quad .
\end{aligned}$$

If, on the other hand, $\mathcal{R}$ decides to deviate and take an action in some later round $i' > i$, the mechanism does not reveal any additional information. This means that

$$\Pr[\theta_{i'} = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \frac{\Pr[\sigma_i = \text{YES} \mid \theta_{i'} = c_{\mathcal{R}}] \cdot \Pr[\theta_{i'} = c_{\mathcal{R}}]}{\Pr[\sigma_i = \text{YES}]}$$

$$= \frac{\frac{n-1}{n} \cdot \frac{1}{n-1} \cdot \frac{1}{n}}{\frac{1}{n} \cdot \left(\frac{n-1}{n} + \frac{1}{n}\right)} = \frac{1}{n} \ .$$

Clearly, this does not give $\mathcal{R}$ any incentive to deviate to a later round $i' > i$.

Finally, assume that $\sigma_1, \ldots, \sigma_i = \text{NO}$. Then,

$$\begin{aligned} \Pr[\theta_i = c_\mathcal{R} \mid \sigma_i = \text{NO}] &= \frac{\Pr[\sigma_i = \text{NO} \mid \theta_i = c_\mathcal{R}] \cdot \Pr[\theta_i = c_\mathcal{R}]}{\Pr[\sigma_i = \text{NO}]} \\ &= \frac{\frac{n-1}{n} \cdot \frac{1}{n}}{\frac{1}{n} \cdot \left(n - 2 + \frac{1}{n} + \frac{n-1}{n}\right)} = \frac{1}{n} \ . \end{aligned}$$

Hence, taking an action without a recommendation to do so does not increase the success probability for $\mathcal{R}$.

Combining these results, this shows that the Elementary Mechanism is persuasive. □

Clearly, the Elementary Mechanism is an optimal persuasive mechanism. Every persuasive mechanism must guarantee a success probability of at least $1/n$ to $\mathcal{R}$. Since the Elementary Mechanism ensures that $\mathcal{S}$ gets $c_\mathcal{S}$ with the remaining probability of $1 - 1/n$, no persuasive mechanism can achieve a better utility for $\mathcal{S}$. Finally, observe that in the case that $c_\mathcal{S} = c_\mathcal{R}$, both $\mathcal{S}$ and $\mathcal{R}$ always get the action with their respective best type.

We continue with the secretary scenario without disclosure.

Secretary Scenario without Disclosure

In the secretary scenario, neither $\mathcal{S}$ nor $\mathcal{R}$ know the valuation-pairs of the types a priori. Still, for the schemes employed by the sender to be persuasive, they must be persuasive even if $\mathcal{R}$ were to know the types. As in the previous basic setting, our mechanism will decide before the online process starts whether to optimize for $\mathcal{S}$ or for $\mathcal{R}$. Since $\mathcal{S}$ does not know the valuations, the *Simple Secretary Mechanism* (Algorithm 13) will then use the classic secretary algorithm due to Dynkin [38] to perform a one-dimensional optimization for either $\mathcal{S}$ or $\mathcal{R}$.

The classic, one-dimensional secretary algorithm works as follows. The first $s = \lfloor n/e \rfloor$ rounds serve as sample phase in which the options are observed. In rounds $s+1, \ldots, n-1$, the first option which constitutes the best option thus far is chosen. In case that no option has been taken in a prior round, the final option in round n is always chosen, regardless of its value. This results in a success probability of at least $1/e$, converging to $1/e$ for large values of n which is optimal for the one-dimensional problem.

Our mechanism uses the classic secretary algorithm for the values $(\varrho_{\theta_i})_{i \in [n]}$ with probability p and for the values $(\xi_{\theta_i})_{i \in [n]}$ with probability $1 - p$. Since the success probability of the classic secretary algorithm is at least $1/e$, a probability $p = e/n = o(1)$ suffices to guarantee $\mathcal{R}$ an overall success probability of $1/n$. We will show below that this indeed results in a persuasive mechanism. This means that the mechanism optimizes for $\mathcal{S}$ with probability $1 - e/n$, which yields a success probability of at least $1/e - 1/n = 1/e - o(1)$ for $\mathcal{S}$. Since $1/e + o(1)$ is the optimal success probability for $\mathcal{S}$ in the one-dimensional setting, the mechanism is asymptotically optimal.

The result is formalized in the following theorem.

Algorithm 13: Simple Secretary Mechanism

Input: Number of rounds n, online sequence of types $\theta_1, \dots, \theta_n$
Set $A_0 = \emptyset$ and $recSent = False$.
Draw $x \sim \text{Unif}[0,1]$. `// Choose whose values to optimize for`
if $x \leq {}^{e}/_{n}$ **then** Set $v = \varrho$.
else Set $v = \xi$.
for *round* $i = 1, \dots, \lfloor {}^{n}/_{e} \rfloor$ **do**
  Set $A_i = A_{i-1} \cup \{\theta_i\}$ and signal NO.
for *round* $i = \lfloor {}^{n}/_{e} \rfloor, \dots, n-1$ **do**
  Set $A_i = A_{i-1} \cup \{\theta_i\}$. **if** $recSent = True$ **then** Signal NO.
  else
    if $v(\theta_i) \geq \max_{t \in A_i} v(t)$ **then** Signal YES and set $recSent = True$.
    else Signal NO.
for *round* n **do**
  if $recSent = True$ **then** Signal NO.
  else Signal YES.

Theorem 4.24
For both cardinal and ordinal sender utility and ordinal receiver utility, the Simple Secretary Mechanism is persuasive in the secretary scenario without disclosure. It yields a success probability of ${}^{1}/_{e} - o(1)$ *and an expected utility of* $({}^{1}/_{e} - o(1)) \cdot \xi_{\max}$ *for* $\mathcal{S}$.

Proof. We assume that $n \geq 3$ such that $s = \lfloor {}^{n}/_{e} \rfloor \geq 1$. Otherwise, the first action is recommended with probability 1. This is the Trivial Mechanism and as such persuasive. Additionally, it gives $\mathcal{S}$ a success probability of at least ${}^{1}/_{2} > {}^{1}/_{e}$.

Clearly, if $\mathcal{R}$ follows the recommendations given by the mechanism, with probability at least $(1 - {}^{e}/_{n}) \cdot {}^{1}/_{e} = {}^{1}/_{e} - {}^{1}/_{n} = {}^{1}/_{e} - o(1)$, type $c_{\mathcal{S}}$ is taken. For the cardinal case, this implies an expected utility of at least $({}^{1}/_{e} - o(1)) \cdot \xi_{max}$ for $\mathcal{S}$.

Let us now show persuasiveness.

First, observe that $\mathcal{R}$ does not learn which variant is used, i.e., whether the mechanism optimizes for $\mathcal{S}$ or for $\mathcal{R}$ until an action is taken and utilities are realized. Second, regardless of the variant of the classic secretary algorithm used, the probability for a signal $\sigma_i =$ YES in round i only depends on the type of action i and the set A_i of types observed in rounds $1, \dots, i$. Clearly, if $i \leq s$, the probability is 0 for a signal $\sigma_i =$ YES. For round $i \in \{s+1, \dots, n-1\}$, the probability for $\sigma_i =$ YES only depends on whether the best type in A_i is revealed in round i and the second best type was observed in the sample phase. Regardless of an optimization for $\mathcal{S}$ or $\mathcal{R}$, this is $\Pr[\sigma_i = \text{YES} \mid A_i] = \frac{1}{i} \cdot \frac{s}{i-1}$. For round $i = n$, unless a prior YES-signal has been sent, the signal is always YES with probability 1.

Note that the mechanism always sends at most a single YES-signal and $\sigma_i =$ YES for some $i \in [n]$ means $\sigma_\ell =$ NO for all $\ell \in [n] \setminus \{i\}$.

We first show that the Simple Secretary Mechanism is persuasive for the case that the utility-values are negatively correlated, i.e., good types for $\mathcal{R}$ are bad for $\mathcal{S}$ and vice versa. Formally, the type with the ℓ-th best utility for $\mathcal{S}$ has the ℓ-th lowest or

$n-\ell+1$-highest utility for $\mathcal{R}$. Following this special case, we argue why persuasiveness holds for arbitrarily correlated utility pairs.

Consider a round $i \in [n-1]$. We consider the different cases.

$\sigma_i =$ **YES:** Clearly, if the mechanism optimizes for $\mathcal{S}$, due to the negatively correlated utility values, $\sigma_i =$ YES implies that $\theta_i \neq c_{\mathcal{R}}$. If the mechanism chose to optimize for $\mathcal{R}$ instead, the type of the current action i has the highest value for $\mathcal{R}$ among the observed types, i.e., $\varrho(\theta_i) \geq \varrho(t)$ for all $t \in A_i$. Due to the random order of the types, the best type is in A_i with probability i/n. Overall, this means that $\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \frac{e}{n} \cdot \frac{i}{n} = \frac{e \cdot i}{n^2}$.

A deviation to a later round $i' > i$ has a success probability of $\Pr[\theta_{i'} = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \frac{n-i}{n} \cdot \frac{1}{n-i} = \frac{1}{n}$. The probability that $c_{\mathcal{R}} \notin A_i$ is $\frac{n-i}{n}$. If that is the case, the probability that action i' has type $c_{\mathcal{R}}$ is $\frac{1}{n-i}$. Hence, a deviation to a later round does not increase the success probability for $\mathcal{R}$ as $i > s = \lfloor n/e \rfloor$, which means that $i > n/e$ and thus $\frac{e \cdot i}{n^2} > \frac{1}{n}$.

$i > s$ **and** $\sigma_\ell =$ **NO for all** $\ell \leq i$**:** If the mechanism optimizes for $\mathcal{R}$, clearly $\theta_i \neq c_{\mathcal{R}}$. If the mechanism chose to optimize for $\mathcal{S}$ instead, clearly $\theta_i \neq c_{\mathcal{S}}$. By our assumption, $c_{\mathcal{R}} \neq c_{\mathcal{S}}$ and thus, $\theta_i = c_{\mathcal{R}}$ cannot be ruled out. With probability i/n, $c_{\mathcal{R}}$ is among the first i types. The fact that $\sigma_\ell =$ NO for all $\ell \leq i$ implies that the best type for $\mathcal{S}$ among the types in A_i was observed during the sample phase. Since we assume negatively correlated utility values, $c_{\mathcal{R}}$ cannot be the best type for $\mathcal{S}$ in A_i. This means that the probability that $c_{\mathcal{R}}$ is action i's type is $\frac{1}{i-1}$, conditional on $c_{\mathcal{R}} \in A_i$ and $\sigma_\ell =$ NO for all $\ell \leq i$. Overall, the probability is

$$\Pr\left[\theta_i = c_{\mathcal{R}} \mid \bigwedge_{\ell=1}^{i} \sigma_\ell = \text{NO}\right] = \frac{n-e}{n} \cdot \frac{i}{n} \cdot \frac{1}{i-1} = \frac{(n-e) \cdot i}{n^2 \cdot (i-1)} .$$

Now, consider a subsequent round $i' > i$. The type $\theta_{i'}$ of action i' is the best type among the type set $A_{i'}$ of observed types with probability $\frac{s}{i' \cdot (i'-1)}$ as noted above, i.e., $v(\theta_{i'}) \geq v(t)$ for all $t \in A_{i'}$. Since we condition on $\sigma_1, \ldots, \sigma_i =$ NO, this means that

$$\Pr\left[v(\theta_{i'}) \geq \max_{t \in A_{i'}} v(t) \mid \bigwedge_{\ell=1}^{i} \sigma_\ell = \text{NO}\right] = \frac{s}{i' \cdot (i'-1)} \Big/ \frac{s}{i} = \frac{i}{i' \cdot (i'-1)} ,$$

where v is ξ or ϱ, depending on the choice of the mechanism. As observed for the previous case, a type which is best so far is $c_{\mathcal{R}}$ with probability $\frac{e \cdot i'}{n^2}$. In total, we get for all $i' < n$ that

$$\Pr\left[\theta_{i'} = c_{\mathcal{R}} \mid \bigwedge_{\ell=1}^{i} \sigma_\ell = \text{NO}\right] = \frac{i}{i' \cdot (i'-1)} \cdot \frac{e \cdot i'}{n^2} = \frac{e \cdot i}{(i'-1) \cdot n^2} .$$

Observe that for round $i' = n$, $\mathcal{S}$ always sends a signal $\sigma_n =$ YES, regardless of the value. The above paragraph already includes the analysis for the special case that in round n, the type is best so far. But, if the sender-optimization is run, a type that is not best so far for $\mathcal{S}$ might be $c_{\mathcal{R}}$. The probability that no type after the sample phase was best so far, conditional on $\sigma_\ell =$ NO for $\ell = 1, \ldots, i$

is $\frac{s}{n}\Big/\frac{s}{i} = \frac{i}{n}$. Additionally, the probability that the $\mathcal{S}$-optimization is run by the mechanism and a type which is not best so far is $c_{\mathcal{R}}$ is $\frac{n-e}{n} \cdot \frac{1}{n-1}$. Hence, the overall probability is

$$\Pr\left[\theta_n = c_{\mathcal{R}} \mid \bigwedge_{\ell=1}^{i} \sigma_\ell = \text{NO}\right] = \frac{e \cdot i}{(n-1) \cdot n^2} + \frac{i}{n} \cdot \frac{n-e}{n} \cdot \frac{1}{n-1} \ .$$

Together, these results imply that by following the mechanism, $\mathcal{R}$ has a success probability of

$$\sum_{i'=i+1}^{n} \frac{e \cdot i}{(i'-1) \cdot n^2} + \frac{i \cdot (n-e)}{n^2 \cdot (n-1)} = \frac{i}{n^2} \cdot \left(\frac{n-e}{n-1} + \sum_{i'=i+1}^{n} \frac{e}{i'-1}\right) \tag{4.12}$$

conditional on $\theta_\ell = \text{NO}$ for $\ell = 1, \ldots, i$. Thus, $\mathcal{R}$ wants to follow the mechanism if

$$\frac{i}{n^2} \cdot \left(\frac{n-e}{n-1} + \sum_{i'=i+1}^{n} \frac{e}{i'-1}\right) \geq \frac{i}{n^2} \cdot \frac{n-e}{i-1} \ ,$$

or, equivalently

$$e \cdot \left(\frac{1}{i-1} - \frac{1}{n-1} + \sum_{i'=i+1}^{n} \frac{1}{i'-1}\right) \geq \frac{n}{i-1} - \frac{n}{n-1} \ ,$$

where we canceled the common factor of $\frac{i}{n^2}$, brought all terms with factor e to the left-hand side and the remaining terms to the right-hand side. The left-hand side can further be simplified by including the two fractions in the summation and shifting the summation index. Together with a common denominator for the right-hand side, this gives us the equivalent formulation of

$$e \cdot \left(\sum_{i'=i-1}^{n-2} \frac{1}{i'}\right) \geq \frac{n \cdot (n-i)}{(n-1) \cdot (i-1)} \ .$$

Purely mathematically, this holds for $i = n$, but in the context of the mechanism, this does not occur as $\sigma_n = \text{YES}$ if $\sigma_\ell = \text{NO}$ for all $\ell \in [n-1]$. This means we can express the constraint for $i < n$ as

$$\frac{e \cdot (n-1)}{n} \geq \frac{n-i}{(i-1) \cdot \sum_{i'=i-1}^{n-2} \frac{1}{i'}} \ . \tag{4.13}$$

Clearly, the left-hand side of (4.13) is fixed for a given value of n. In the following claim, we show that the right-hand side is monotonically decreasing in i and thus $i = s + 1$ provides the strongest lower bound for (4.13).

Claim 2
The lower bound in equation (4.13) *is monotonically decreasing in* i.

n	3	4	5	6	7	8	9	10	11	12	13
s	1	1	1	2	2	2	3	3	4	4	4
LHS	$\frac{2e}{3}$	$\frac{3e}{4}$	$\frac{4e}{5}$	$\frac{5e}{6}$	$\frac{6e}{7}$	$\frac{7e}{8}$	$\frac{8e}{9}$	$\frac{9e}{10}$	$\frac{10e}{11}$	$\frac{11e}{12}$	$\frac{12e}{13}$
RHS	1	$\frac{4}{3}$	$\frac{18}{11}$	$\frac{18}{13}$	$\frac{120}{77}$	$\frac{50}{29}$	$\frac{700}{459}$	$\frac{560}{341}$	$\frac{3780}{2509}$	$\frac{4410}{2761}$	$\frac{55440}{32891}$

Table 5: Inequality (4.14) for $3 \le n \le 13$. LHS > RHS for every entry since $e \ge 5/2$.

Proof of Claim 2. For $i > s+1$, we show that the right-hand side of (4.13) does not decrease when the value of i decreases to $i-1$, or

$$\frac{n-i+1}{(i-2)\cdot\sum_{i'=i-2}^{n-2}\frac{1}{i'}} \ge \frac{n-i}{(i-1)\cdot\sum_{i'=i-1}^{n-2}\frac{1}{i'}} \ .$$

This holds if and only if

$$\begin{aligned}\sum_{i'=i-1}^{n-2}\frac{1}{i'} &\ge \frac{(n-i)\cdot(i-2)}{(n-i+1)\cdot(i-1)}\cdot\sum_{i'=i-2}^{n-2}\frac{1}{i'}\\ &= \frac{(n-i)\cdot(i-2)}{(n-i+1)\cdot(i-1)}\cdot\left(\sum_{i'=i-1}^{n-2}\frac{1}{i'}+\frac{1}{i-2}\right)\\ &= \frac{(n-i)\cdot(i-2)}{(n-i+1)\cdot(i-1)}\cdot\sum_{i'=i-1}^{n-2}\frac{1}{i'}+\frac{n-i}{(n-i+1)\cdot(i-1)} \ .\end{aligned}$$

This inequality is satisfied if and only if

$$(n-1)\cdot\sum_{i'=i-1}^{n-2}\frac{1}{i'} \ge n-i \ .$$

Clearly, for $i = n-1$, the inequality states $\frac{n-1}{n-2} \ge 1$, which is true. The fact that the inequality holds for some i implies that it also holds for $i-1$. This is due to the fact that the left-hand side increases by $\frac{n-1}{i-2} > 1$ but the right-hand side increases by 1 when the value of i decreases to $i-1$. This proves the claim.

$\square_{\text{Claim 2}}$

Using the claim, we can plug $i = s+1$ into (4.13), which gives us

$$\frac{e\cdot(n-1)}{n} \ge \frac{n-s-1}{s\cdot\left(\sum_{i'=s}^{n-2}\frac{1}{i'}\right)} \ . \tag{4.14}$$

Recall that $s = \lfloor n/e \rfloor$. For $n \le 13$, Table 5 shows the left-hand and right-hand sides of inequality (4.14). Since it holds for all $n \le 13$, we will now assume $n \ge 14$ and use the following inequality (cf. [75])

$$\frac{1}{2(\ell+1)} < H_\ell - \ln\ell - \gamma < \frac{1}{2\ell} \ .$$

Here, $\gamma \approx 0.5772$ is the Euler-Mascheroni constant and $H_\ell = \sum_{i=1}^{\ell} 1/i$ the ℓ-th harmonic number. This allows us to bound

$$\sum_{i'=\lfloor n/e\rfloor}^{n-2}\frac{1}{i'} = H_{n-2} - H_{\lfloor n/e\rfloor - 1}$$

$$\begin{aligned}
&\geq \frac{1}{2(n-1)} + \ln(n-2) - \frac{1}{2(\lfloor n/e \rfloor - 1)} - \ln\left(\lfloor n/e \rfloor - 1\right) \\
&\geq 1 + \frac{1}{2(n-1)} - \frac{1}{2\left(\lfloor n/e \rfloor - 1\right)}
\end{aligned} \tag{4.15}$$

since $n - 2 \geq e \cdot (\lfloor n/e \rfloor - 1)$ and thus $\ln \frac{n-2}{\lfloor n/e \rfloor - 1} \geq 1$.

Recall that we want to show (4.14) for $s = \lfloor n/e \rfloor$, or equivalently

$$\frac{\left\lfloor \frac{n}{e} \right\rfloor \cdot e \cdot (n-1)}{n} \cdot \sum_{i'=\lfloor n/e \rfloor}^{n-2} \frac{1}{i'} \geq n - \left\lfloor \frac{n}{e} \right\rfloor - 1 \ .$$

We can lower bound

$$\begin{aligned}
&\frac{\left\lfloor \frac{n}{e} \right\rfloor \cdot e \cdot (n-1)}{n} \cdot \sum_{i'=\lfloor n/e \rfloor}^{n-2} \frac{1}{i'} \\
&\qquad \geq \frac{\left(\frac{n}{e} - 1\right) \cdot e \cdot (n-1)}{n} \cdot \left(1 + \frac{1}{2(n-1)} - \frac{1}{2\left(\left\lfloor \frac{n}{e} \right\rfloor - 1\right)}\right) \\
&\qquad \geq \left(n - e - 1 + \frac{e}{n}\right) \cdot \left(1 + \frac{1}{2(n-1)} - \frac{1}{2\left(\frac{n}{e} - 2\right)}\right) \\
&\qquad = n - 1 - e + \frac{e}{n} + \frac{n - 1 - e + \frac{e}{n}}{2(n-1)} - \frac{n - 1 - e + \frac{e}{n}}{2\left(\frac{n}{e} - 2\right)} \\
&\qquad \geq n - 1 + \frac{1}{2} - e - \frac{e}{2(n-1)} - \frac{e \cdot (n - 2e + e - 1 + \frac{e}{n})}{2(n-2e)} \\
&\qquad \geq n - 1 + \frac{1}{2} - e - \frac{e}{2(n-1)} - \frac{e}{2} - \frac{e^2}{2(n-2e)} \\
&\qquad \geq n - 1 + \frac{1}{2} - 3 - \frac{3}{26} - \frac{3}{2} - \frac{8}{16} \\
&\qquad = n - 1 - \frac{120}{26} \\
&\qquad \geq n - 1 - 5 \\
&\qquad \geq n - \left\lfloor \frac{n}{e} \right\rfloor - 1 \ ,
\end{aligned}$$

showing (4.14). We used that $n \geq 14$, thus $\left\lfloor \frac{n}{e} \right\rfloor \geq 5$, and $e \leq \sqrt{8} \leq 3$.

$i \leq s$**:** Finally, consider the case that $\mathcal{R}$ wants to blindly take an action during the sample phase. Since no information is given to $\mathcal{R}$ by the mechanism, the probability of getting an action with type $c_{\mathcal{R}}$ is $1/n$. This does not depend on the round $i \leq s$ in which $\mathcal{R}$ takes the action or the variant of the classic secretary algorithm the mechanism is using.

We can use the bound obtained in equation (4.12) to bound the success probability for $\mathcal{R}$ when following the mechanism. Since $i \leq s$, $\mathcal{R}$ will always get a NO-signal until round i from the mechanism, so we can drop the conditioning on having only NO up to round i. Since the mechanism starts the phase in which YES-signals are sent in round $s+1$, this gives $\mathcal{R}$ a success probability of

$$\frac{s}{n^2} \cdot \left(\frac{n-e}{n-1} + \sum_{i'=s+1}^{n} \frac{e}{i'-1} \right) \quad .$$

Thus, $\mathcal{R}$ should wait and not take action $i \leq s$ if

$$\begin{aligned} \frac{1}{n} &\leq \frac{s}{n^2} \cdot \left(\frac{n-e}{n-1} + \sum_{i'=s+1}^{n} \frac{e}{i'-1} \right) \\ &= \frac{s}{n^2} \cdot \left(\frac{n}{n-1} + e \cdot \sum_{i'=s}^{n-2} \frac{1}{i'} \right) \quad . \end{aligned}$$

This inequality is equivalent to

$$\frac{1}{s} \leq \frac{1}{n-1} + \frac{e}{n} \cdot \sum_{i'=s}^{n-2} \frac{1}{i'} \quad ,$$

or

$$\frac{e \cdot (n-1)}{n} \geq \frac{n-1-s}{s \cdot \sum_{i'=s}^{n-2} \frac{1}{i'}}$$

which is exactly inequality (4.14) which we proved to hold for the previous case.

This concludes the proof for persuasiveness when the values for $\mathcal{S}$ and $\mathcal{R}$ are negatively correlated, i.e., the ℓ-th best type for $\mathcal{S}$ is the $n-\ell-1$-best type for $\mathcal{R}$. If that is not the case, the success probability for $\mathcal{R}$ changes when the mechanism optimizes for $\mathcal{S}$ – if the mechanism looks for $c_{\mathcal{R}}$, this change does not have an impact on receiver's success probability. If $c_{\mathcal{R}}$ is not the worst type for $\mathcal{S}$, the chances of seeing $c_{\mathcal{R}}$ as a type that is best so far increases. Assume that $c_{\mathcal{R}}$ has rank $x < n$ for $\mathcal{S}$, where a rank x means that there are $x-1$ better types in the type set. A signal $\sigma_i = \text{YES}$ for a type with rank x in round $i > s$ implies that all $x-1$ better types arrive in a later round. Otherwise, $c_{\mathcal{R}}$ cannot be a type that is best so far.

This means that the probability of getting $c_{\mathcal{R}}$ upon a signal $\sigma_i = \text{YES}$ when the mechanism uses the variant optimizing for $\mathcal{S}$ is $\frac{i}{n} \cdot \prod_{\ell=0}^{x-2} \frac{n-i-\ell}{n-\ell-1} \geq 0$.

Additionally, if action $i > s$ is not recommended, the probability that $\theta_i = c_{\mathcal{R}}$ weakly decreases. In this scenario, the probability is

$$\frac{i}{n} \cdot \frac{1}{i-1} \cdot \left(1 - \prod_{\ell=0}^{x-2} \frac{n-i-\ell}{n-1-\ell} \right) \leq \frac{i}{n \cdot (i-1)} \quad ,$$

where $\frac{i}{n \cdot (i-1)}$ is the probability that $\theta_i = c_{\mathcal{R}}$ conditional on the mechanism optimizing for $\mathcal{S}$ and $\sigma_\ell = \text{NO}$ for all $\ell \in [i]$.

Clearly, during the sample phase, the probability to get $c_{\mathcal{R}}$ when taking an action stays $1/n$. The same holds for a deviation to a round after a YES-signal has been sent. The calculations are similar to the ones done above for the negatively-correlated scenario.

Algorithm 14: Adaptive Elementary Mechanism

Input: Set of valuation pairs $(\varrho_t, \xi_t)_{t \in [n]}$, online sequence of types $\theta_1, \dots, \theta_n$

Set $recSent = False$.
for *round* $i = 1, \dots, n$ **do**
  if $recSent = False$ **then**
    if $c_{\mathcal{S}} = \theta_i$ **then** Signal YES and set $recSent = True$.
    else if $c_{\mathcal{R}} = \theta_i$ **then**
      Draw $x \sim \text{Unif}[0,1]$.
      if $x \leq \frac{1}{n-i}$ **then** Signal YES and set $recSent = True$.
      else Signal NO.
    else Signal NO.
  else Signal NO.

All in all, this means that $\mathcal{R}$ weakly increases the success probability by following the mechanism for a decreasing sender-rank x of $c_{\mathcal{R}}$. This means that the mechanism is persuasive when $c_{\mathcal{R}}$ has rank $x < n$. Since we showed above that it is persuasive for a sender-rank $x = n$ of $c_{\mathcal{R}}$, this clearly means that the mechanism is persuasive regardless of the correlation of the types' valuations. □

This concludes the secretary scenario without disclosure. In the following, we discuss the disclosure scenarios, starting with the basic scenario with disclosure.

Basic Scenario with Disclosure

In this section, we discuss the basic scenario with disclosure. This means that $\mathcal{S}$ and $\mathcal{R}$ both know the set of types a priori and $\mathcal{R}$ is informed of the types of rejected actions. This means that $\mathcal{S}$ can no longer use the Elementary Mechanism – after each rejection of a type $t \neq c_{\mathcal{R}}$, the set of remaining types shrinks and the probability to find $c_{\mathcal{R}}$ by taking a random action increases. Interestingly, even though the mechanism needs to be adapted to the new scenario with disclosure, our *Adaptive Elementary Mechanism* (Algorithm 14) achieves a success probability of $1 - o(1)$ for $\mathcal{S}$ and thus an expected utility of $(1 - o(1)) \cdot \xi_{max}$ in the cardinal setting.

The mechanism only sends a YES-signal on $c_{\mathcal{S}}$ or $c_{\mathcal{R}}$. In every round i, unless a YES-signal has been sent in a previous round, the mechanism proceeds as follows. If $\theta_i = c_{\mathcal{S}}$, the signal is $\sigma_i =$ YES with probability 1. If $\theta_i = c_{\mathcal{R}}$, the signal is $\sigma_i =$ YES with probability $\frac{1}{n-i}$. Otherwise, the mechanism signals NO. Clearly, this means that at most a single YES-signal is sent.

Our first result is that the mechanism is persuasive.

Lemma 4.25
For ordinal receiver utility, the Adaptive Elementary Mechanism is persuasive in the basic scenario with disclosure.

Proof. The receiver learns the type of dismissed actions. This means that $\mathcal{R}$ knows when $c_{\mathcal{R}}$ is no longer among the available options. In this case, a deviation from the recommendation is not profitable for $\mathcal{R}$. Hence, we assume that $c_{\mathcal{R}}$ has not been observed.

Consider round i and $\sigma_i = \text{YES}$. This means that the current action has type $c_{\mathcal{R}}$ with probability

$$\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \frac{\frac{1}{n-i}}{1 + \frac{1}{n-i}} = \frac{1}{n-i+1} \quad .$$

A deviation to a later round $i' > i$ means that $\mathcal{R}$ does not obtain additional information from $\mathcal{S}$. The probability that $\theta_i \neq c_{\mathcal{R}}$ conditional on $\sigma_i = \text{YES}$ is $1 - \frac{1}{n-i+1}$ and the remaining types are drawn in a uniform random order. If $\theta_i \neq c_{\mathcal{R}}$, this leaves $n - i$ types from which to draw the type of action i'. Therefore, $\mathcal{R}$ will get $c_{\mathcal{R}}$ in round $i' > i$ with probability

$$\Pr[\theta_{i'} = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \left(1 - \frac{1}{n-i+1}\right) \cdot \frac{1}{n-i} = \frac{1}{n-i+1} \quad .$$

Clearly, $\mathcal{R}$ does not increase the success probability of finding $c_{\mathcal{R}}$ when deviating from the YES-recommendation in round i.

Now, consider the scenario that $\sigma_\ell = \text{NO}$ for all $\ell \in [i]$. $\mathcal{R}$ is able to get $c_{\mathcal{R}}$ with probability

$$\Pr\left[\theta_i = c_{\mathcal{R}} \mid \bigwedge_{\ell=1}^{i} \sigma_\ell = \text{NO}\right] = \frac{\frac{n-i-1}{n-i}}{n-i-1 + \frac{n-i-1}{n-i}} = \frac{\frac{n-i-1}{n-i}}{\frac{(n-i-1)(n-i+1)}{n-i}} = \frac{1}{n-i+1}$$

by deviating from the recommendation and taking action i.

We know that the mechanism always sends a YES-recommendation. Assume it comes in round $i' > i$. We further know that $\mathcal{R}$ has a success probability of $\frac{1}{n-i'+1}$ conditional on $\sigma_{i'} = \text{YES}$ and $c_{\mathcal{R}}$ not having been dismissed in one of the intermediate rounds $\ell = i, \ldots, i'-1$. The probability that $c_{\mathcal{R}}$ is dismissed in round ℓ, conditional on $\sigma_\ell = \text{NO}$ is $\frac{1}{n-\ell+1}$. This means that the overall probability of getting type $c_{\mathcal{R}}$ eventually in the round with a signal YES is

$$\begin{aligned}
\Pr\left[\theta_{i'} = c_{\mathcal{R}} \mid \bigwedge_{\ell=i}^{i'-1} \sigma_\ell = \text{NO} \wedge \sigma_{i'} = \text{YES}\right] &= \prod_{\ell=i}^{i'-1} \left(1 - \frac{1}{n-\ell+1}\right) \cdot \frac{1}{n-i'+1} \\
&= \prod_{\ell=i}^{i'-1} \left(\frac{n-\ell}{n-\ell+1}\right) \cdot \frac{1}{n-i'+1} \\
&= \frac{1}{n-i+1} \quad .
\end{aligned}$$

Clearly, $\mathcal{R}$ cannot increase the success probability of getting $c_{\mathcal{R}}$ by deviating from the recommendation. Hence, the Adaptive Elementary Mechanism is persuasive. $\square$

Our second result is the approximation ratio of the Adaptive Elementary Mechanism.

Proposition 4.26
For ordinal receiver utility in the basic scenario with disclosure, the Adaptive Elementary Mechanism yields a success probability of $1 - o(1)$ *and an expected utility of* $(1 - o(1)) \cdot \xi_{\max}$ *for* $\mathcal{S}$.

Proof. Clearly, if $c_{\mathcal{S}} = c_{\mathcal{R}}$, both $\mathcal{S}$ and $\mathcal{R}$ always get an action with their respective best type. For the remainder of the proof, we therefore assume that $c_{\mathcal{S}} \neq c_{\mathcal{R}}$.

Conditional on $\theta_\ell \neq c_{\mathcal{S}}$ for $\ell = 1, \ldots, i-1$, action i has type $c_{\mathcal{S}}$ with probability $\frac{1}{n-i+1}$. We denote by u_r the probability that $\mathcal{R}$ eventually takes an action with type $c_{\mathcal{S}}$ if $r \geq 2$ rounds remain and $c_{\mathcal{S}}$ and $c_{\mathcal{R}}$ have not yet been observed. Clearly, if $c_{\mathcal{R}}$ has been observed (and the corresponding action was not taken), the probability for getting $c_{\mathcal{S}}$ is 1. If $c_{\mathcal{S}}$ is observed, it is taken with probability 1. If some other type $t \neq c_{\mathcal{S}}, c_{\mathcal{R}}$ is observed, $r-1$ rounds remain and $c_{\mathcal{S}}$ and $c_{\mathcal{R}}$ remain in the set of types to come. Hence, we can express u_r recursively, where $u_2 = \frac{1}{2}$ is the base case. If the set of remaining types consists of only $c_{\mathcal{S}}$ and $c_{\mathcal{R}}$, whichever one of them is observed first will get a signal YES with probability 1. The recursion for $r > 2$ is

$$u_r = \underbrace{\frac{1}{r} \cdot 1}_{c_{\mathcal{S}}} + \underbrace{\frac{1}{r} \cdot \frac{r-2}{r-1}}_{c_{\mathcal{R}}} + \underbrace{\frac{r-2}{r} \cdot u_{r-1}}_{\text{neither}} \quad .$$

Clearly, the overall success probability for $\mathcal{S}$ is u_n, where

$$\begin{aligned}
u_n &= \frac{1}{n} + \frac{1}{n} \cdot \frac{n-2}{n-1} + \frac{n-2}{n} \cdot u_{n-1} \\
&= \frac{2n-3}{n \cdot (n-1)} + \frac{n-2}{n} \cdot \left(\frac{2n-5}{(n-1) \cdot (n-2)} + \frac{n-3}{n-1} \cdot u_{n-2} \right) \\
&= \frac{2n-3}{n \cdot (n-1)} + \frac{2n-5}{n \cdot (n-1)} + \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot u_{n-2} \\
&= \frac{2n-3}{n \cdot (n-1)} + \frac{2n-5}{n \cdot (n-1)} + \frac{2n-7}{n \cdot (n-1)} + \frac{(n-4) \cdot (n-3)}{n \cdot (n-1)} \cdot u_{n-3} \\
&= \ldots \\
&= \sum_{\ell=1}^{n-2} \frac{2(n-\ell)-1}{n \cdot (n-1)} + \frac{(n-(n-1)) \cdot (n-(n-2))}{n \cdot (n-1)} \cdot u_2 \\
&= \sum_{\ell=1}^{n-1} \frac{2(n-\ell)-1}{n \cdot (n-1)} \\
&= \sum_{\ell=1}^{n-1} \frac{2 \cdot \ell}{n \cdot (n-1)} - \frac{1}{n} \\
&= 1 - \frac{1}{n} \quad .
\end{aligned}$$

Hence, the success probability for $\mathcal{S}$ is $u_n = 1 - 1/n$. In case of cardinal utility for $\mathcal{S}$, this means that $\mathcal{S}$ gets an expected utility of at least $(1 - 1/n) \cdot \xi_{max}$. □

Hence, in contrast to the case of cardinal receiver utility, the additional information for $\mathcal{R}$ does not decrease the success probability or the expected utility of $\mathcal{S}$ in the respective basic scenarios. In the next section, we discuss our final scenario, i.e., the secretary scenario with disclosure.

Secretary Scenario with Disclosure

In the secretary scenario with disclosure, neither $\mathcal{S}$ nor $\mathcal{R}$ know the values of the types a priori. Yet, persuasiveness dictates that even a receiver who does know the valuations

Algorithm 15: First-Opt Mechanism

Input: Number of rounds n, sample size s, online sequence of types $\theta_1, \dots, \theta_n$

Set $A_0 = \emptyset$ and $recSent = False$.

for *round* $i = 1, \dots, s$ **do**

  Set $A_i = A_{i-1} \cup \{\theta_i\}$ and signal NO.

for $i = s + 1$ *to* $n - 1$ **do**

  Set $A_i = A_{i-1} \cup \{\theta_i\}$.

  if $recSent = True$ **then** Signal NO.

  else

    if $\varrho(\theta_i) \geq \max_{t \in A_i} \varrho(t)$ **or** $\xi(\theta_i) \geq \max_{t \in A_i} \xi(t)$ **then**

      Signal YES, set $recSent = True$.

    else Signal NO.

for *round* n **do**

  if $recSent = True$ **then** Signal NO.

  else Signal YES.

should be interested in following the mechanism's recommendation. In contrast to the basic scenario with disclosure, a persuasive mechanism can no longer decide before the start of the online process whether to optimize for $\mathcal{S}$ or $\mathcal{R}$. Whenever it is revealed to $\mathcal{R}$ that the type of the action dismissed in the last round was best so far for either $\mathcal{R}$ or $\mathcal{S}$, the receiver would know which variant is run. To combat this, we use an adaptation of our Simple Secretary Mechanism called the *First-Opt Mechanism* (Algorithm 15). The mechanism recommends the first type which is best so far for either $\mathcal{S}$ or $\mathcal{R}$ after a sample phase of length $s = \lfloor n/2 \rfloor$ during which signals are NO. If no YES-signal has been sent up to round n, $\sigma_n =$ YES regardless of the values of θ_n.

In the following lemma, we show that the First-Opt Mechanism is persuasive. Afterwards, in Theorem 4.28, we prove the lower bound on the approximation guarantee for the mechanism. Finally, Theorem 4.29 shows that the First-Opt Mechanism is optimal for negatively correlated utility values for $\mathcal{S}$ and $\mathcal{R}$. Hence, no other mechanism can achieve a better success probability for $\mathcal{S}$ in such a setting. As before, we will focus on the case of ordinal utility for $\mathcal{S}$ as the results easily translate to the setting with cardinal sender utility.

Lemma 4.27

For ordinal receiver utility, the First-Opt Mechanism is persuasive in the secretary scenario with disclosure.

Proof. Recall that the mechanism should be persuasive even if $\mathcal{R}$ were to know the utility values a priori. $\mathcal{R}$ is only interested in type $c_{\mathcal{R}}$. This means that once an action with type $c_{\mathcal{R}}$ is dismissed, $\mathcal{R}$ has no incentive to deviate from the recommendations given by $\mathcal{S}$. Hence, we assume that $c_{\mathcal{R}}$ has not been rejected.

Now, consider round $i \leq n - 1$ with $\sigma_i =$ YES. This means that

$$\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] \geq \frac{1}{n - i + 1} \ .$$

This is due to the fact that all $n - i + 1$ remaining types in round i might individually be better for $\mathcal{S}$ or $\mathcal{R}$ than the previous types which are already revealed, resulting in a signal $\sigma_i =$ YES for any of them. The random order implies that $c_\mathcal{R}$ is drawn with probability $\frac{1}{n-i+1}$. A deviation by $\mathcal{R}$ would mean dismissing action i and taking another action in a later round $i' > i$. $\mathcal{S}$ will only signal NO after round i, so $\mathcal{R}$ does not get additional information from the sender's signals. The success probability for $\mathcal{R}$ is therefore

$$\Pr[\theta_{i'} = c_\mathcal{R} \mid \sigma_i = \text{YES}] \le \left(1 - \frac{1}{n-i+1}\right) \cdot \frac{1}{n-i} = \frac{1}{n-i+1} \quad ,$$

since the probability that $\theta_i \neq c_\mathcal{R}$ is at most $\left(1 - \frac{1}{n-i+1}\right)$ and if that is the case, $c_\mathcal{R}$ is drawn in round i' with probability $\frac{1}{n-i}$. Clearly, it is not profitable for $\mathcal{R}$ to dismiss an action with a YES-signal and wait for a later round.

Now, consider the case that $\sigma_\ell =$ NO for all $\ell \in [i]$. Clearly, if $i > s$, the type of the current action is neither best so far for $\mathcal{R}$ nor for $\mathcal{S}$. Hence, the probability that $\theta_i = c_\mathcal{R}$ is 0. It is optimal to wait for some round $i' > i$ with $\sigma_{i'} =$ YES. If $i \le s$, $\mathcal{R}$ does not get any information besides the disclosure of foregone types. This means that $\Pr[\theta_i = c_\mathcal{R}] = \frac{1}{n-i+1}$, since we assume that $c_\mathcal{R}$ has not been revealed in a previous round. If $\mathcal{R}$ decides to follow the mechanism, $\mathcal{S}$ will eventually send a signal $\sigma_{i'} =$ YES in a round $i' > s \ge i$. For $\theta_{i'} = c_\mathcal{R}$, two events need to occur: First, $c_\mathcal{R}$ cannot be the type of one of the actions $i, \dots, s$. The probability for this is

$$\Pr[\theta_i, \theta_{i+1}, \dots, \theta_s \neq c_\mathcal{R}] = \prod_{\ell=i}^{s} \left(1 - \frac{1}{n-\ell+1}\right) = \frac{n-s}{n-i+1} \quad .$$

Second, among the remaining types, $c_\mathcal{R}$ needs to be the first to be revealed among the set of types which are better than all previous types for $\mathcal{S}$ or $\mathcal{R}$. This set clearly has a size at most $n - s$. Due to the random order, the probability that $c_\mathcal{R}$ is the first type which is best so far for $\mathcal{S}$ or $\mathcal{R}$ is at least $\frac{1}{n-s}$. Hence, the overall success probability for $\mathcal{R}$ when following the mechanism is at least $\frac{n-s}{n-i+1} \cdot \frac{1}{n-s} = \frac{1}{n-i+1}$. Clearly, this incentivizes $\mathcal{R}$ to follow the mechanism and wait for a YES-signal rather than taking action i.

Overall, $\mathcal{R}$ maximizes the success probability of getting $c_\mathcal{R}$ by following the mechanism. Thus, the mechanism is persuasive. $\square$

Theorem 4.28
For ordinal receiver utility in the secretary scenario with disclosure, the First-Opt Mechanism with $s = \lfloor n/2 \rfloor$ yields a success probability of at least $1/4 - o(1)$ and an expected utility of at least $(1/4 - o(1)) \cdot \xi_{\max}$ for $\mathcal{S}$.

Proof. Let A_i denote the random set of types observed in rounds $1, \dots, i$. The first s rounds constitute the sample phase, hence, regardless of the observed values, no type is taken. After the sample phase, i.e., for $i > s$, we observe that we can generate the type in round i by first uniformly at random drawing the set A_i, and then drawing a type from this set uniformly at random. At most two types in A_i result in $\sigma_i =$ YES, namely the type with the best value for $\mathcal{S}$ and the type with the best value for $\mathcal{R}$. Depending on the instance and the set A_i, it could be that the same type is best for both $\mathcal{S}$ and $\mathcal{R}$. Thus, we can upper bound the probability that $\sigma_i =$ YES for a given set A_i by $2/i$. This means that

$$\Pr[\sigma_i = \text{NO} \mid A_i] \ge \begin{cases} 1 & i = 1, \dots, s \\ \frac{i-2}{i} & i = s+1, \dots, n-1 \end{cases} \quad . \tag{4.16}$$

Overall, the success probability for $\mathcal{S}$, i.e., the probability that $\mathcal{S}$ gets $c_{\mathcal{S}}$, conditioned on the set A_i of arrived candidates up to round i is

$$\begin{aligned}\Pr[\theta_i = c_{\mathcal{S}} \mid A_i] &= \Pr[c_{\mathcal{S}} \in A_i] \cdot \Pr[\theta_i = c_{\mathcal{S}} \mid c_{\mathcal{S}} \in A_i] \cdot \Pr[\sigma_1, \ldots, \sigma_{i-1} = \text{NO} \mid A_{i-1}] \\ &= \frac{i}{n} \cdot \frac{1}{i} \cdot \prod_{\ell=s+1}^{i-1} \Pr[\sigma_\ell = \text{NO} \mid A_\ell] \\ &\geq \frac{1}{n} \cdot \prod_{\ell=s+1}^{i-1} \frac{\ell-2}{\ell} \\ &= \frac{(s-1)\cdot s}{n\cdot(i-2)\cdot(i-1)} \quad .\end{aligned}$$

These probabilities are independent of the sets A_i, which means that the overall success probability for $\mathcal{S}$ is at least

$$\begin{aligned}\sum_{i=s+1}^{n} \frac{(s-1)\cdot s}{n\cdot(i-2)\cdot(i-1)} &= \frac{(s-1)\cdot s}{n} \cdot \sum_{i=s+1}^{n} \left(\frac{1}{i-2} - \frac{1}{i-1}\right) \\ &= \frac{(s-1)\cdot s}{n} \cdot \left(\frac{1}{s-1} - \frac{1}{n-1}\right) \\ &= \frac{s}{n} \cdot \left(1 - \frac{s-1}{n-1}\right) .\end{aligned}$$

Setting $s = \lfloor n/2 \rfloor$ maximizes the above expression. This gives $\mathcal{S}$ a success probability of $1/4 - o(1)$. For the cardinal setting, this gives $\mathcal{S}$ an expected utility of at least $(1/4 - o(1)) \cdot \xi_{max}$. □

Note that due to the symmetric structure of the mechanism, the same bound holds for $\mathcal{R}$, as well. This means that $\mathcal{R}$ also has a success probability of $1/4 - o(1)$ when $\mathcal{S}$ uses the mechanism.

Next, we show that the First-Opt Mechanism is optimal, i.e., there is no persuasive mechanism which achieves a higher success probability for $\mathcal{S}$. To do this, we use an instance with negatively correlated utility values for the types. This means that the ℓ-th best type for $\mathcal{S}$ is the $(n-\ell+1)$-th best type for $\mathcal{R}$, for all $\ell \in [n]$. Observe that cardinal utility values clearly define a ranking of the types. Besides this property, the values themselves are irrelevant for $\mathcal{S}$ as there is no bound on the size of the values. Even if all observed values are in a close range, it could be that the next type has a utility which is larger by several orders of magnitude. Hence, we assume that $\mathcal{S}$ ignores the cardinal values and focuses only on the rank of the types. Since we are studying ordinal utility for $\mathcal{R}$, clearly $\mathcal{R}$ is only interested in the rank of the types as well.

Theorem 4.29
If utilities of sender and receiver are negatively correlated, the First-Opt Mechanism maximizes the success probability for $\mathcal{S}$ among all persuasive mechanisms in the secretary scenario with disclosure in the case of ordinal receiver utility.

For the proof, we use a class of randomized *best-so-far* mechanisms. These mechanisms only signal YES in a round $i < n$ if the current type is best so far either for $\mathcal{S}$ or for $\mathcal{R}$. Additionally, we assume that in the final round, a signal YES is always sent unless there has been a prior signal YES. Intuitively, it makes sense that best-so-far mechanisms are good as sending a signal YES for a type that is not best so far among the observed types will never satisfy $\mathcal{S}$ or $\mathcal{R}$. We formalize this in Lemma 4.30, where we show that there exists a persuasive best-so-far mechanism φ' for any persuasive mechanism φ such that the success probability of φ' is at least as high as that of φ. Using this insight and showing some additional properties of best-so-far mechanisms in Lemmas 4.31 and 4.32, we then show that our First-Opt Mechanism is an optimal best-so-far mechanism for negatively correlated utility values.

Lemma 4.30

If utilities of sender and receiver are negatively correlated, then for every persuasive mechanism there is a persuasive best-so-far mechanism with weakly higher success probability for $\mathcal{S}$.

Proof. From a persuasive mechanism φ, we construct a new persuasive mechanism φ'. The new mechanism simply simulates the behavior of φ unless φ signals YES on some action $i < n$ whose type is not best so far for $\mathcal{S}$ or for $\mathcal{R}$. Clearly, a recommendation for such a type means that φ is not a best-so-far mechanism. Rather than sending a NO-recommendation, assume for now that φ' sends a third signal NO$'$. This surely informs $\mathcal{R}$ that the current type is not best so far and thus taking the action in the current round helps neither $\mathcal{S}$ nor $\mathcal{R}$. After the signal NO$'$ in round i, φ' then sends NO signals in every remaining round except for round n, in which the mechanism sends a signal $\sigma_n =$ YES.

This means that whenever $\mathcal{R}$ sees a signal YES in some round $i = 1, \ldots, n-1$, the type of action i is best so far for $\mathcal{S}$ or $\mathcal{R}$. Taking action i weakly increases the conditional probability that an action with type $c_\mathcal{R}$ or $c_\mathcal{S}$ is taken. When φ' sends $\sigma_i =$ NO$'$, this means that $\theta_i \notin \{c_\mathcal{S}, c_\mathcal{R}\}$. Since φ would have sent a signal YES and φ is a persuasive mechanism, this means that neither $\mathcal{S}$ nor $\mathcal{R}$ would have gotten their respective best type. Thus, deterministically taking the final action cannot decrease the probability of getting $c_\mathcal{S}$ or the probability of getting $c_\mathcal{R}$. If φ has not sent a YES recommendation in rounds $i = 1, \ldots, n-1$, φ' deterministically sends a signal $\sigma_n =$ YES. This clearly either emulates the behavior of φ or weakly increases the probability that $c_\mathcal{S}$ or $c_\mathcal{R}$ is taken.

Hence, φ' is persuasive and weakly increases the success probability for $\mathcal{S}$. Finally, observe that φ' stays persuasive when instead of sending a signal NO$'$ a signal NO is sent. This decreases the amount of information $\mathcal{R}$ has when receiving a NO-signal. Hence, $\mathcal{R}$ is further incentivized to follow the mechanism. □

The lemma allows us to restrict attention to best-so-far mechanisms. Note that not all best-so-far mechanisms are persuasive. As an example, consider our Simple Secretary Mechanism for the secretary scenario without disclosure. The mechanism reveals which version is run with the first action after the sample phase which has the highest utility for $\mathcal{S}$ or $\mathcal{R}$ so far but is not recommended. If $\mathcal{R}$ knows that the sender-optimization is run and $c_\mathcal{R}$ is not a very good type for $\mathcal{S}$ (which $\mathcal{R}$ can be assumed to know for persuasiveness), a signal $\sigma_i =$ YES in a round $i < n$ would most certainly mean that $\theta_i \neq c_\mathcal{R}$.

We denote by A_{i-1} the set of types observed in rounds $1, \ldots, i-1$. Conditional on A_{i-1} and $\sigma_\ell =$ NO for all $\ell \in [i-1]$, we denote by $p_i^\mathcal{X}$ the probability that $\sigma_i =$ YES if θ_i is the best type so far for $\mathcal{X} \in \{\mathcal{S}, \mathcal{R}\}$. We show a necessary condition for persuasive mechanisms.

Lemma 4.31
If a best-so-far mechanism is persuasive for negatively correlated utilities, it satisfies $p_i^{\mathcal{R}} \geq p_i^{\mathcal{S}}$ for all rounds $i \in [n]$ and all histories A_{i-1}.

Proof. We assume that no YES-signal has been sent in a prior round, otherwise, the mechanism only signals NO, regardless of the current type.

Clearly, for round $i = n$, $p_i^{\mathcal{R}} = p_i^{\mathcal{S}} = 1$ as $\sigma_n =$ YES if no previous YES-signal was sent. Now, consider the beginning of round $i \leq n-1$, before $\mathcal{S}$ observes θ_i. This means that both $\mathcal{S}$ and $\mathcal{R}$ know the set A_{i-1} of the first $i-1$ types which all have been rejected and thus disclosed. There are $n-i+1$ types left which $\mathcal{S}$ does not know. Since persuasiveness requires a receiver who knows the complete set of types to follow the mechanism, we can assume that $\mathcal{R}$ is able to partition the set of types that have not been revealed into three sets. The first set $B_{\mathcal{S}}$ includes all types that would be best so far for $\mathcal{S}$ in round i, the second set $B_{\mathcal{R}}$ includes the ones that are best so far for $\mathcal{R}$ in round i and the third set includes the rest of the types. Since we assume negatively correlated utilities, these sets clearly are disjoint.

Now, consider the following scenario. Among the types in A_{i-1} is the overall second best type for $\mathcal{R}$. Hence, a type which is best so far for $\mathcal{R}$ must be $c_{\mathcal{R}}$, i.e., $B_{\mathcal{R}} = \{c_{\mathcal{R}}\}$. All other types are better for $\mathcal{S}$ than the types in A_{i-1}, i.e., the set $B_{\mathcal{S}}$ consists of the top $n-i$ types for $\mathcal{S}$. This means that there are $n-i$ types which would be best so far for $\mathcal{S}$ and only a single type which would be best so far for $\mathcal{R}$ if it arrived in round i.

Towards a contradiction, assume that $p_i^{\mathcal{S}} > p_i^{\mathcal{R}}$. This means that

$$\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{YES}] = \frac{p_i^{\mathcal{R}}}{(p_i^{\mathcal{R}} + (n-i) \cdot p_i^{\mathcal{S}})} < \frac{1}{(n-i+1)} \quad .$$

Hence, a signal to take action i gives $\mathcal{R}$ a smaller probability to get $c_{\mathcal{R}}$ than a remaining random action. On the other hand, a signal not to take action i provides an incentive to take action i, i.e.,

$$\Pr[\theta_i = c_{\mathcal{R}} \mid \sigma_i = \text{NO}] = \frac{1 - p_i^{\mathcal{R}}}{1 - p_i^{\mathcal{R}} + (n-i) \cdot (1 - p_i^{\mathcal{S}})} > \frac{1}{n-i+1} \quad .$$

Clearly, this is a contradiction to the mechanism being persuasive.

Note that $\mathcal{S}$ knows nothing about the instance and is therefore unable to see whether such a situation arises from the set A_{i-1} of observed types. Since every set A_{i-1} can easily be extended to an instance such that the above situation arises, every persuasive mechanism needs to satisfy $p_i^{\mathcal{R}} \geq p_i^{\mathcal{S}}$ for every round $i \in [n]$ and history A_{i-1} of observed types up to round $i-1$. □

Let us now consider the class of best-so-far mechanisms with $p_i^{\mathcal{R}} \geq p_i^{\mathcal{S}}$ for all $i \in [n]$. Surely, it includes all persuasive best-so-far mechanisms. Additionally, it possibly contains non-persuasive best-so-far mechanism that satisfy this condition. We optimize the success probability for $\mathcal{S}$ within this class, where we assume that a signal YES automatically means that the current action is taken. For a given value $p_i^{\mathcal{S}}$, setting $p_i^{\mathcal{R}} = p_i^{\mathcal{S}}$ maximizes the conditional success probability upon a signal YES. Clearly, if the current action has a type that is not best so far for $\mathcal{S}$, the probability that it is $c_{\mathcal{S}}$ is 0. By decreasing $p_i^{\mathcal{R}}$ to its lowest viable value, the conditional probability that the current option is best so far for $\mathcal{S}$ is as high as possible. This means that we should set

$p_i = p_i^{\mathcal{R}} = p_i^{\mathcal{S}}$ in each round i to maximize the success probability for $\mathcal{S}$. Additionally, in the following Lemma, we show $p_i \in \{0, 1\}$ for all $i \in [n]$ in the scenario of negatively correlated utilities.

Lemma 4.32
There is an optimal persuasive Best-So-Far mechanism for negatively correlated utilities such that $p_i \in \{0, 1\}$ for all $i \in [n]$.

Proof. We show the lemma using backwards induction. By definition, in round n, we have $p_n = 1$. For round $i = n - 1$, condition on the event no signal YES has been sent in a prior round and the current action has a type which is best so far for either $\mathcal{S}$ or $\mathcal{R}$. Since $p_i^{\mathcal{R}} = p_i^{\mathcal{S}} = p_i$ for all i, we do not need to consider these cases separately and will just say that the type is best so far for short. Clearly, this means there are two options. Either recommend (i.e., take) the current type or wait for round n. Since $\mathcal{S}$ is able to determine the success probability for both choices, $\mathcal{S}$ can maximize the overall success probability by choosing whichever round provides the higher probability of getting $c_{\mathcal{S}}$. Hence, setting either $p_{n-1} = 0$ if the probability to get $c_{\mathcal{S}}$ in round n is higher or $p_{n-1} = 1$ if the probability of getting $c_{\mathcal{S}}$ in round $n - 1$ is higher is optimal. This procedure can be extended to the earlier rounds as well.

In round $i < n-1$, we condition on no YES-signal in a prior round and θ_i being best so far. Then, $\mathcal{S}$ is able to determine the probability to get $c_{\mathcal{S}}$ when taking the action in round i as well as the probability to get $c_{\mathcal{S}}$ when running the optimal scheme for rounds $i + 1, \ldots, n$ with $p_{i+1}, \ldots, p_n \in \{0, 1\}$. Deterministically choosing the option with the higher probability of getting $c_{\mathcal{S}}$ in round i maximizes the overall success probability for $\mathcal{S}$. Hence, we get $p_i \in \{0, 1\}$.

By induction, this proves that $p_i = p_i^{\mathcal{S}} = p_i^{\mathcal{R}} \in \{0, 1\}$ for all $i \in [n]$. □

This result allows us to prove Theorem 4.29, where we derive optimal values $p_i \in \{0, 1\}$ and show that this optimal best-so-far mechanism is our First-Opt Mechanism.

Proof of Theorem 4.29. In the following, we assume that $n \geq 3$ to avoid technicalities. Clearly, for $n = 1$, the first action is recommended and taken, and for $n = 2$, it does not matter whether action 1 or 2 is taken. Both provide a success probability of $\frac{1}{2}$ to $\mathcal{S}$. Setting $p_1 = 0$ and $p_2 = 1$, i.e., sending $\sigma_1 =$ NO and $\sigma_2 =$ YES is exactly our First-Opt Mechanism.

Our proof generalizes ideas for the classic secretary problem due to Beckmann [20]. Let us first introduce some notation. We denote by B_i the event that type θ_i is best so far (for either $\mathcal{S}$ or $\mathcal{R}$). By D_i, we denote the event that θ_i is dismissed and by D_{-i} the event that all previous types $\theta_1, \ldots, \theta_{i-1}$ have been dismissed. Note that we have not yet shown that our optimal mechanism is persuasive. With foresight, we assume that $\mathcal{R}$ still follows the recommendations given by the sender and thus $\mathcal{S}$ can essentially take or dismiss actions singlehandedly. When signaling, taking an action clearly is represented by a signal YES and a dismissal means a signal NO. Finally, let

$$u_i = \Pr[c_{\mathcal{S}} \text{ is taken in rounds } i, \ldots, n \mid B_i, D_{-i}] \qquad \text{and}$$
$$v_i = \Pr[c_{\mathcal{S}} \text{ is taken in rounds } i+1, \ldots, n \mid D_i, D_{-i}] \ .$$

Since the online process ends when an action is taken, to reach some round $i+1$, it must be the case that D_{-i} and D_i occur, or D_{-i} and B_i occur and $p_i = 0$. Let us consider the conditional probabilities of B_{i+1}, i.e., $\Pr[B_{i+1} \mid D_i, D_{-i}]$ and $\Pr[B_{i+1} \mid B_i, D_{-i}]$.

Due to the random order of the types, for each set A_i of types observed in rounds $\ell = 1, \dots, i$, when conditioning on A_i, the same probabilities for B_ℓ and D_ℓ arise for every round ℓ. This further implies that for every A_i, conditioned on the set, we have the same probability that events B_i, D_{-i} and D_i occur. Hence, these events are independent of the set A_i.

Since we focus on negatively correlated utilities, a single type cannot be better for both $\mathcal{S}$ and $\mathcal{R}$ than the previously observed types. Hence, in round $i+1$, conditioned on each set A_{i+1}, the probability $\Pr[B_{i+1} \mid A_{i+1}] = \frac{2}{i+1}$. Furthermore, since D_i and D_{-i} are independent of A_i, we have $\Pr[B_{i+1} \mid A_{i+1}, D_i, D_{-i}] = \Pr[B_{i+1} \mid D_i, D_{-i}] = \frac{2}{i+1}$. Similarly, $\Pr[B_i \mid A_i]$ is the same regardless of A_i, which gives us $\Pr[B_{i+1} \mid A_{i+1}, B_i, D_{-i}] = \Pr[B_{i+1} \mid B_i, D_{-i}] = \frac{2}{i+1}$.

This means that v_i can be expressed using u_{i+1} and v_{i+1} as follows.

$$v_i = \frac{2}{i+1} \cdot u_{i+1} + \frac{i-1}{i+1} \cdot v_{i+1} \tag{4.17}$$

Let us now condition on event B_i for $i > 1$, i.e., θ_i is best so far. Taking action i results in the sender-optimal type $c_{\mathcal{S}}$ with probability $\frac{i}{2n}$ since $c_{\mathcal{S}} \in A_i$ with probability $\frac{i}{n}$ and with probability $\frac{1}{2}$, θ_i is best so far for $\mathcal{S}$ (and with probability $\frac{1}{2}$, θ_i is best so far for $\mathcal{R}$). For $i = 1$, the probability is clearly $1/n$ and the first type is always best so far.

Dismissing action i results in eventually taking $c_{\mathcal{S}}$ with probability v_i. Hence, an optimal best-so-far mechanism will set

$$u_i = \max\left\{\frac{1}{n}, \frac{i}{2n}, v_i\right\} \quad , \tag{4.18}$$

where $\frac{i}{2n} \geq \frac{1}{n}$ for all $i \geq 2$.

We use backwards induction to solve this recurrence relation. The base cases are $u_n = \frac{1}{2}$ since the best-so-far type is either the best for $\mathcal{S}$ or for $\mathcal{R}$, each with probability $\frac{1}{2}$, and $v_n = 0$ since the process ends after n rounds. We get $v_{n-1} = \frac{n-2}{n} \cdot v_n + \frac{2}{n} u_n = \frac{1}{n}$, which implies $u_{n-1} = \frac{n-1}{2n} \geq \frac{1}{n}$. As long as $\frac{i}{2n} > v_i$ (or $1/n > v_i$ for $i = 1$), the sender's success probability is maximized by setting $p_i = 1$ and recommending action i if it has a type which is best so far. Otherwise, if $v_i \geq \frac{i}{2n}$ ($v_i \geq 1/n$ for $i = 1$), $\mathcal{S}$ should wait and thus set $p_i = 0$. While the current type might be best so far, this is due to the fact that only a very limited sample has been observed at that point and the probability of having seen $c_{\mathcal{S}}$ already is rather small. Note that the first two types are always best so far, no matter what their global respective ranks among all n types are. The following lemma summarizes the values for u_i and v_i for all $i \in [n]$. We prove the lemma below.

Lemma 4.33
The values of u_i and v_i are given by

$$u_i = \begin{cases} \frac{i}{2n} & i \geq \frac{n+1}{2} \\ \frac{n}{4(n-1)} & i < \frac{n+1}{2},\ n \text{ even}, \\ \frac{n+1}{4n} & i < \frac{n+1}{2},\ n \text{ odd}, \end{cases} \qquad v_i = \begin{cases} \frac{i \cdot (n-i)}{n \cdot (n-1)} & i \geq \frac{n+1}{2} \\ \frac{n}{4(n-1)} & i < \frac{n+1}{2},\ n \text{ even}, \\ \frac{n+1}{4n} & i < \frac{n+1}{2},\ n \text{ odd}. \end{cases}$$

The lemma shows that it is optimal to set $p_i = 0$ for $i \leq n/2$ and $p_i = 1$ for $i > n/2$. This means that, when encountering a type that is best so far in rounds $i = 1, \dots, \lfloor n/2 \rfloor$, it is optimal for $\mathcal{S}$ to dismiss the corresponding action, i.e., send a NO-signal. Whenever

$\mathcal{S}$ observes an action with a type that is best so far in a later round, it is then optimal to take that action, i.e., send a YES-signal.

Clearly, this is exactly our First-Opt Mechanism. We showed that the mechanism is persuasive, which means that $\mathcal{R}$ is incentivized to follow the sender's signals. This proves the theorem. $\square_{\text{Theorem 4.29}}$

Let us now prove Lemma 4.33.

Proof of Lemma 4.33. First, consider the case $i \geq \frac{n+1}{2}$ and start with $i = n$. The base cases are $v_n = 0 = \frac{n\cdot(n-n)}{n\cdot(n-1)}$ and $u_n = \frac{1}{2} = \max\left\{\frac{1}{n}, \frac{n}{2n}, 0\right\}$.

Now, assume that the lemma holds for $i + 1 \geq \frac{n+1}{2} + 1$. We show that it holds for i as well. Plugging in the definition of v_i, we get

$$\begin{aligned} v_i &= \frac{i-1}{i+1} \cdot v_{i+1} + \frac{2}{i+1} \cdot u_{i+1} \\ &= \frac{i-1}{i+1} \cdot \frac{(i+1)\cdot(n-(i+1))}{n\cdot(n-1)} + \frac{2}{i+1} \cdot \frac{i+1}{2n} \\ &= \frac{(i-1)\cdot(n-i-1)+(n-1)}{n\cdot(n-1)} \\ &= \frac{i\cdot(n-i)}{n\cdot(n-1)} \quad . \end{aligned}$$

Since $i \geq \frac{n+1}{2}$, we have $\frac{i}{2n} \geq \frac{i\cdot(n-i)}{n\cdot(n-1)} = v_i$. This means that $u_i = \frac{i}{2n}$.

Now, consider the second case in which $i < \frac{n+1}{2}$. The base case is $i = \frac{n}{2}$ or $i = \frac{n-1}{2}$, depending on the parity of n.

Case n even: We have

$$\begin{aligned} v_{n/2} &= \frac{\frac{n}{2}-1}{\frac{n}{2}+1} \cdot v_{n/2+1} + \frac{2}{\frac{n}{2}+1} \cdot u_{n/2+1} \\ &= \frac{n-2}{n+2} \cdot \frac{\frac{n+2}{2}\cdot\frac{n-2}{2}}{n\cdot(n-1)} + \frac{2}{\frac{n}{2}+1} \cdot \frac{\frac{n}{2}+1}{2n} \\ &= \frac{(n-2)^2+4(n-1)}{4n\cdot(n-1)} \\ &= \frac{n}{4(n-1)} \quad . \end{aligned}$$

Since $i = \frac{n}{2} < \frac{n+1}{2}$ and thus $\frac{n/2}{2n} = \frac{1}{4} < \frac{n}{4(n-1)} = v_i$, we now have $u_i = v_i$. This obviously leads to $v_i = v_{i+1}$ and thus $u_i = v_i$ for all $i < \frac{n+1}{2}$.

Case n odd: The case is similar to the previous one. We get

$$\begin{aligned} v_{\frac{n-1}{2}} &= \frac{\frac{n-1}{2}-1}{\frac{n+1}{2}} \cdot v_{\frac{n+1}{2}} + \frac{2}{\frac{n+1}{2}} \cdot u_{\frac{n+1}{2}} \\ &= \frac{n-3}{n+1} \cdot \frac{\frac{n+1}{2}\cdot\frac{n-1}{2}}{n\cdot(n-1)} + \frac{2}{\frac{n+1}{2}} \cdot \frac{\frac{n+1}{2}}{2n} \\ &= \frac{n-3+4}{4n} \end{aligned}$$

$$= \frac{n+1}{4n} \ .$$

Again, $i = \frac{n-1}{2} < \frac{n+1}{2}$ and thus $\frac{\frac{n-1}{2}}{2n} = \frac{1}{4} - \frac{1}{4n} < \frac{n+1}{4n} = v_i$. Hence, $u_i = v_i$ for all $i < \frac{n+1}{2}$.

This concludes the proof of the lemma. $\square_{\text{Lemma 4.33}}$

With this final result, we conclude our discussion of online Bayesian persuasion.

In the next chapter, we discuss online delegated search. In delegated search, $\mathcal{R}$ is the one with commitment power, unlike the Bayesian persuasion setting, where $\mathcal{S}$ has commitment power.

Chapter 5

Online Delegated Search

In this chapter, we discuss an online version of the delegated search problem. The process is the following.

1. Both $\mathcal{R}$ and $\mathcal{S}$ know the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of types of the n actions.
2. $\mathcal{R}$ commits to an acceptance scheme φ.
3. $\mathcal{S}$ learns the scheme φ.
4. In each round $i = 1, \ldots, n$:
   4.1. $\mathcal{S}$ learns the type θ_i of action i.
   4.2. $\mathcal{S}$ decides whether to propose action i, considering the reaction by $\mathcal{R}$ based on φ.
   4.3. If the action is not proposed, round $i+1$ begins. Otherwise, $\mathcal{R}$ decides whether to accept the proposal according to the acceptance scheme φ. Regardless of the decision, the process ends after a proposal.

As benchmark, we use the optimal expected utility if $\mathcal{R}$ were to perform a one-dimensional search. Clearly, the one-dimensional online search constitutes a prophet-inequality stopping problem [60, 50]. For prophet inequalities, an approximation factor of $1/2$ of the expected maximum of the n actions is always possible using a simple threshold rule [61, 71], and for certain classes of instances like IID instances, even better approximation ratios were recently shown [1, 29]. Interestingly, when studying an offline delegation problem with IID actions, Kleinberg and Kleinberg [58] were able to design constant-factor approximation algorithms using these results for prophet inequalities. As benchmark, Kleinberg and Kleinberg use the maximum value for $\mathcal{R}$ among the n actions – which is exactly the value $\mathcal{R}$ would get when performing a one-dimensional search offline. Their result is the following.

Theorem 5.1 ([58, Theorem 4])
If types are drawn IID, there always is a threshold τ such that a deterministic scheme φ with $\varphi_{ij} = 1$ if and only if $\varrho(\theta_{ij}) \geq \tau$ for all $i \in [n], j \in [m]$ or $\varphi_{ij} = 1$ if and only if $\varrho(\theta_{ij}) > \tau$ for all $i \in [n], j \in [m]$ guarantees $\mathcal{R}$ an expected utility of at least $1/2 \cdot \mathbb{E}\left[\max_{i \in [n]} \theta_i\right]$.

Additionally, for atomless distributions with independent utilities for $\mathcal{R}$ and $\mathcal{S}$, they are able to further increase the approximation guarantees using the improved results for prophet inequalities.

In contrast to the constant-factor approximation results for the offline scenario, our first result is a negative one. In Section 5.1, we use an IID instance to show that in general, $\mathcal{R}$ cannot expect a utility of more than $O\left(1/n\right)$ of the utility when performing a one-dimensional (online) search. Note that the optimal online search always guarantees a constant-factor approximation to the result of the optimal offline search. Since all our results will be in asymptotic notation, online vs offline search is a distinction without a difference. A trivial acceptance scheme, which only allows an action which has the highest a-priori expected utility for $\mathcal{R}$ guarantees an approximation ratio of $\Omega\left(1/n\right)$, matching the upper bound of $O\left(1/n\right)$.

Since our instance which proves the upper bound of $O\left(1/n\right)$ relies on an exponential ratio between the minimum and maximum value for $\mathcal{S}$ in the order of $n^{\Theta(n)}$, we study a parameterized instance in which the multiplicative discrepancy in the positive values for $\mathcal{S}$ is bounded by a factor $\alpha \geq 1$ in Section 5.2. We show that an approximation ratio of $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$ in terms of parameter α is possible. Clearly, setting $\alpha = n^{\Theta(n)}$ gives the matching upper bound using the same instance as above. Additionally, in Section 5.3, we consider a different parameterized setting with a parameter $\beta \geq 1$ which bounds the ratios between positive utility values of individual types for $\mathcal{S}$ and $\mathcal{R}$. Here, we show an $\Omega\left(\frac{1}{\log\beta}\right)$-approximation, where an upper bound of $O\left(\frac{\log\log\beta}{\log\beta}\right)$ holds due to our original instance above with $\beta = n^{\Theta(n)}$.

Furthermore, we consider a variant of the problem where $\mathcal{R}$ only learns $\varrho(\theta)$ if a type θ is proposed. Hence, $\mathcal{R}$ is not able to distinguish types with the same value in a single round. Thus, for all $i \in [n]$, it must hold that $\varphi_{ij} = \varphi_{ij'}$ for all types $j, j' \in [m]$ with $\varrho(\theta_{ij}) = \varrho(\theta_{ij'})$. Since $\mathcal{R}$ does not learn the sender's value, we dub this our *sender-oblivious proposal* scenario. Similarly, we call the original scenario the *sender-aware proposal* scenario. Clearly, our trivial mechanism does not take the sender's values into account, hence, we still get a $\Theta\left(1/n\right)$-approximation in the general case. In the parameterized settings, $\mathcal{R}$ is worse off. For parameter α, we describe a mechanism with an approximation ratio of $\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$ and show an upper bound of $O\left(1/\sqrt{\alpha}\right)$. For parameter β, we show a lower bound of $\Omega\left(1/\beta\right)$ and an upper bound of $O\left(1/\sqrt{\beta}\right)$. We summarize the results in Table 6.

Before going into the details of the results, we shortly discuss an introductory example with deterministic strategies, i.e. $\varphi_{ij} \in \{0, 1\}$ for all i, j. There are two rounds with the actions' types distributed according to Table 7.

round i	1		2	
type θ_{ij}	θ_{11}	θ_{12}	θ_{21}	θ_{22}
value-pair (ϱ_{ij}, ξ_{ij})	(1,3)	(8,3)	(4,2)	(4,16)
probability q_{ij}	0.75	0.25	0.75	0.25

Table 7: Introductory example for delegated online search

The benchmark consists of $\mathcal{R}$ performing a one-dimensional online search. If type θ_{12} is drawn in round 1, $\mathcal{R}$ would take action 1 with a utility of 8. Otherwise, $\mathcal{R}$ waits for round 2 and receives a utility of 4. In expectation, $\mathcal{R}$ gets a utility of 5.

	$\mathcal{S}$-aware proposals		$\mathcal{S}$-oblivious proposals	
General	$\Theta\left(\frac{1}{n}\right)$		$\Theta\left(\frac{1}{n}\right)$	
	Prop 5.2	Thm 5.3	Prop 5.2	Thm 5.3
α-bounded $\mathcal{S}$	$\Theta\left(\frac{\log\log\alpha}{\log\alpha}\right)$		$\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$	$O\left(\frac{1}{\sqrt{\alpha}}\right)$
	Thm 5.4	Cor 5.7	Thm 5.12	Thm 5.10
β-bounded ratio	$\Omega\left(\frac{1}{\log\beta}\right)$	$O\left(\frac{\log\log\beta}{\log\beta}\right)$	$\Omega\left(\frac{1}{\beta}\right)$	$O\left(\frac{1}{\sqrt{\beta}}\right)$
	Thm 5.19	Cor 5.22	Prop 5.16	Cor 5.17

Table 6: Approximation guarantees for delegated online search.

The sender prefers type θ_{22} with utility 16. Hence, as long as $\varphi_{22} = 1$, $\mathcal{S}$ will never propose action 1, regardless of the type. Thus, if $\varphi_{22} = 1$, $\mathcal{R}$ cannot get a higher expected utility than 4. An acceptance scheme which achieves this utility is $\varphi_{ij} = 1$ for all $i, j \in \{1, 2\}$. Note that the actual values φ_{1j} are not important as $\mathcal{S}$ will always wait for round 2. If $\mathcal{R}$ does not accept θ_{22}, $\mathcal{S}$ is not incentivized to wait for action 2. Clearly, an optimal acceptance scheme for sender-aware proposals would be $\varphi_{11} = 0, \varphi_{12} = 1, \varphi_{21} = 1, \varphi_{22} = 0$. This results in an expected utility 4.25 for $\mathcal{R}$. Recall that for online Bayesian persuasion, it was optimal to accept the last action if no other action was taken in a prior round. This example shows us that this is clearly not the case for online delegated search. Overall, the expected utility for $\mathcal{R}$ is maximized even though in some cases, no action is taken.

For sender-oblivious proposals, $\mathcal{R}$ cannot distinguish the possible types in round 2. Hence, $\varphi_{21} = \varphi_{22}$. As we have seen above, if $\varphi_{22} = 1$, $\mathcal{S}$ will always wait for round 2. Otherwise, if $\varphi_{21} = \varphi_{22} = 0$, the expected utility for $\mathcal{R}$ is upper bounded by 2.75. Clearly, accepting (only) the proposals in the second round is optimal for $\mathcal{R}$ with an expected utility of 4 in the case of sender-oblivious proposals.

5.1 General Case

For our first result, we show that the following deterministic *Trivial Acceptance Scheme* φ^{triv} provides an approximation ratio of at least $\Omega\left({}^{1}\!/_{n}\right)$. The scheme allows only action i^*, where $i^* = \arg\max_{i\in[n]} \mathbb{E}\left[\varrho(\theta_{ij})\right]$ is the round with the highest a-priori expectation for $\mathcal{R}$. Hence, $\varphi^{triv}_{i^*j} = 1$ for all $j \in [m]$ and $\varphi^{triv}_{ij} = 0$ for all $i \neq i^*$ and all $j \in [m]$. Note that this only requires access to the utility values of $\mathcal{R}$. Thus, the lower bound clearly holds for sender-oblivious proposals.

Proposition 5.2
For online delegation, the Trivial Acceptance Scheme φ^{triv} guarantees $\mathcal{R}$ at least a ${}^{1}\!/_{n}$-approximation of the expected utility for optimal (online) search. It can be computed in time polynomial in the input size.

Proof. Finding the action i^* with the highest a priori expected utility for $\mathcal{R}$ can clearly be done in time polynomial in the input size. If only action i^* is allowed, $\mathcal{S}$ always proposes this action. This means that $\mathcal{R}$ gets an expected utility of $\mathbb{E}\left[\varrho(\theta_{i^*})\right]$. The optimal search gives an expected utility of

$$\mathbb{E}\left[\max_{i\in[n]} \varrho(\theta_i)\right] \leq \mathbb{E}\left[\sum_{i=1}^{n} \varrho(\theta_i)\right] = \sum_{i=1}^{n} \mathbb{E}\left[\varrho(\theta_i)\right] \leq n \cdot \mathbb{E}\left[\varrho(\theta_{i^*})\right] \quad .$$

This shows the Proposition. □

We show a matching upper bound using an IID instance, i.e., all actions' types are drawn independently from a single distribution. Additionally, the values for $\mathcal{R}$ and $\mathcal{S}$ can be seen as being drawn independently from each other. The bound holds even when $\mathcal{R}$ receives full information on the proposed action's type. Clearly, with less information, $\mathcal{R}$ cannot achieve a better approximation.

Theorem 5.3
There is a class of instances of online delegation in the IID setting, in which every acceptance scheme φ obtains at most an $O(1/n)$-approximation of the expected utility for optimal (online) search.

Proof. We use the following class of instances with n actions. With probability $1/n$, the drawn type has utility 1 for $\mathcal{R}$. With the remaining utility $1 - 1/n$, the utility for $\mathcal{R}$ is 0. Independently from the receiver's value, $\mathcal{S}$ gets one out of n utility values, each with probability $1/n$. The values are $n^{4\ell}$ for $\ell = 1, \ldots, n$. In combination, we get the following distribution with $2n$ types as shown in Table 8.

Type	1	2	...	n	$n+1$	$n+2$	...	$2n$
$\mathcal{R}$		1				0		
$\mathcal{S}$	n^4	n^8	...	n^{4n}	n^4	n^8	...	n^{4n}
Probability		$\frac{1}{n^2}$				$\frac{1}{n} - \frac{1}{n^2}$		

Table 8: Distribution for proof of Theorem 5.3

Without loss of generality, we can assume that $\varphi_{ij} = 0$ for all i and $j > n$. All such types provide no utility to $\mathcal{R}$. Instead, if $\mathcal{R}$ accepts some type $j > n$ in round i with positive probability, $\mathcal{S}$ might be tempted to wait when a type $j' \leq n$ is drawn in an earlier round $i' < i$. Additionally, a proposal of an action in round $i < n$ with utility 0 implies that actions $i+1, \ldots, n$ are not considered – which might have provided positive utility for $\mathcal{R}$. Hence, $\mathcal{R}$ can weakly increase the expected utility by setting $\varphi_{ij} = 0$ for all $i \in [n]$ and $j > n$.

From the receiver's side, there are only two different outcomes, either no utility or utility 1. Hence, both online as well as offline searches will always result in a type with utility 1 if such a type was drawn at all. The probability for this to occur is $1 - (1 - 1/n)^n \geq 1 - 1/e = \Theta(1)$.

The sender, on the other hand, will never propose an action with type $j < n$ in round i if there exists a type $j' > j$ which $\mathcal{R}$ would accept in a later round $i' > i$. This is due to the fact that the expected utility from round i' is higher than the utility $\mathcal{S}$ would get from type j in round i.

More formally, we consider an optimal scheme $\varphi = (\varphi_{ij})_{i\in[n],j\in[n]}$ – since $\varphi_{ij} = 0$ for all $j > n$. We treat φ as a matrix and describe steps to bound the expected utility for $\mathcal{R}$.

First of all, we decrease all entries (i, j) with $\varphi_{ij} \leq {}^1\!/\!_n$ to 0. Clearly, this cannot decrease the expected utility by more than $n^2 \cdot \frac{1}{n^2} \cdot \frac{1}{n} = \frac{1}{n}$ – there are n^2 entries, each representing a type which is drawn with probability $\frac{1}{n^2}$ and accepted with probability $\varphi_{ij} \leq {}^1\!/\!_n$. Hence, we now have a matrix φ where each entry is either 0 or at least ${}^1\!/\!_n$.

Now, let us consider entries $\varphi_{ij} > {}^1\!/\!_n$. Assume that $\varphi_{i'j'} > {}^1\!/\!_n$ for some $i' < i, j' < j$. It is clearly beneficial for $\mathcal{S}$ to wait for round i and hope that type j is drawn instead of proposing type j' in round i'. Since $j' < j$, the sender's expected utility for round i is at least $n^{4j} \cdot \frac{1}{n^2} \cdot \frac{1}{n} = n^{4j-3} > n^{4j'}$. Thus, waiting and not proposing type j' in round i' can only increase the sender's expected utility. Hence, we can assume without loss of generality that $\varphi_{i'j'} = 0$ for all $i' < i, j' < j$ whenever $\varphi_{ij} > {}^1\!/\!_n$.

Similarly, consider $\varphi_{i'j} < \varphi_{ij}$ for $i' < i$. Setting $\varphi_{i'j} = \varphi_{ij}$ cannot decrease the receiver's utility and does not change the behavior of $\mathcal{S}$. Consider an earlier round $\hat{i} < i'$. If $\mathcal{S}$ encounters a better type $\hat{j} > j$ which has positive probability to be accepted in round $\hat{i}$, increasing $\varphi_{i'j}$ will not dissuade $\mathcal{S}$ from proposing $\hat{j}$ in round $\hat{i}$. If $\mathcal{S}$ encounters a worse type $j' < j$ in round $\hat{i}$, by the previous paragraph we have $\varphi_{\hat{i}j'} = 0$. Thus, the adjustment does not result in a change in the behavior of $\mathcal{S}$ in rounds prior to i'. If $\mathcal{S}$ wanted to propose type j in round i', the increase in $\varphi_{i'j}$ only increases the probability that $\mathcal{R}$ accepts the proposal. If $\mathcal{S}$ did not want to propose type j in round i', this means that in a later round, a better type is to come which has a positive acceptance probability. By our first adjustment, waiting increases the expected utility of $\mathcal{S}$. Hence, this adjustment does not influence the sender's behavior. It can only increase the overall probability of a proposal and thus the expected utility for $\mathcal{R}$. Similarly, if $\varphi_{ij'} < \varphi_{ij}$ for $j' < j$, we can set $\varphi_{ij'} = \varphi_{ij}$. This does not change the sender's incentives and can only (weakly) increase the receiver's expected utility.

We apply both sets of adjustments repeatedly, starting with the entry φ_{nn}, i.e., the entry in the last row and last column of the matrix. If $\varphi_{nn} > 0$, we are done with the final column, as all entries in the last column (and in the last row) now have the same value. Otherwise, we consider entry $n - 1$ in column n and continue in the same way. Once column n is adjusted, we move to column $n - 1$, starting with the final entry n. We repeat this process until column 1 has been adjusted as well.

As a final step, we apply our above assumption, i.e., $\varphi_{i'j'} = 0$ for all $i' < i, j' < j$ whenever $\varphi_{ij} > {}^1\!/\!_n$.

Note that when increasing the values, we did not require that the acceptance probability to be increased be greater than 0. This implies that we might increase some value φ_{ij}, only to set it to 0 in the final step.

After these adjustments, only $2n-1$ non-zero entries remain, forming a "Manhattan path" of non-zero entries from φ_{1n} to φ_{n1}, i.e., if $\varphi_{ij} > 0$, the path continues either with $\varphi_{(i+1)j} > 0$ or $\varphi_{i(j+1)} > 0$.

We can use this to upper-bound the expected utility for $\mathcal{R}$ by assuming that all $\varphi_{ij} = 1$ if $\varphi_{ij} > 0$ and $\mathcal{S}$ always proposes any acceptable action. Then, a union bound gives us a probability that $\mathcal{R}$ gets a proposal of $(2n - 1) \cdot \frac{1}{n^2} = O\left({}^1\!/\!_n\right)$. Since proposal probability and expected utility are the same, $\mathcal{R}$ cannot get a higher utility than $O\left({}^1\!/\!_n\right)$ in this online delegated search instance. Note that we decreased the expected utility by at most ${}^1\!/\!_n$ when decreasing $\varphi_{ij} \leq {}^1\!/\!_n$ to 0 as our first adjustment. Clearly, this does not change the overall asymptotic value of $O\left({}^1\!/\!_n\right)$ as an upper bound on the expected utility for $\mathcal{R}$. □

The values for $\mathcal{S}$ in the class of instances proving Theorem 5.3 differ by an exponential multiplicative factor of n^{4n-4}, i.e., $\max_{i,j} \xi_{ij} = n^{4n-4} \cdot \min_{i,j} \xi_{ij}$. In the following section, we use $\alpha \geq 1$ to parameterize the ratio between maximum and minimum value for $\mathcal{S}$. This allows us to study the effect this ratio has on the approximation guarantee obtainable for $\mathcal{R}$.

5.2 α-Bounded Sender Utility Values

In this section, we assume that

$$\alpha = \frac{\max\{\xi_{ij} \mid i \in [n], j \in [m]\}}{\min\{\xi_{ij} \mid i \in [n], j \in [m]\}} \ .$$

Clearly, this requires $\xi_{ij} > 0$ for all $i \in [n], j \in [m]$. For simplicity, we assume that this holds and will discuss at the end of each subsection what the implications are if there are types that provide no utility for $\mathcal{S}$.

Without loss of generality, we scale the values for $\mathcal{S}$ such that $\min\{\xi_{ij} \mid i \in [n], j \in [m]\} = 1$ and $\max\{\xi_{ij} \mid i \in [n], j \in [m]\} = \alpha$. This clearly does not change the incentives for either $\mathcal{S}$ or $\mathcal{R}$. For short, we say that $\mathcal{S}$ has *α-bounded utilities.*

In the first subsection, we consider the case of sender-aware proposals, i.e., $\mathcal{R}$ learns both $\xi(\theta)$ and $\varrho(\theta)$ when $\mathcal{S}$ proposes a type θ. In this setting, we show a tight approximation factor of $\Theta\left(\frac{\log\log\alpha}{\log\alpha}\right)$.

In the second subsection, we discuss sender-oblivious proposals. This decreases the information $\mathcal{R}$ learns when a type is proposed as the value for $\mathcal{S}$ is not revealed. We show an upper bound of $O\left(1/\sqrt{\alpha}\right)$ on the approximation factor as well as an acceptance scheme which gives an approximation guarantee of $\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$.

5.2.1 Sender-Aware Proposals

When $\mathcal{S}$ has α-bounded utilities, we can use our *Binning Algorithm* (Algorithm 16) to compute an acceptance scheme for $\mathcal{R}$ which provides an $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$-approximation to the expected utility of an optimal search by $\mathcal{R}$.

The algorithm partitions the types with the highest utility values for $\mathcal{R}$ up to a combined probability mass of just above $1/2$ into $O\left(\frac{\log\alpha}{\log\log\alpha}\right)$ many bins $B_0, B_1, \ldots$ Each bin B holds as many types as possible such that $\mathcal{S}$ still wants to propose the first action with a type from B, assuming that $\mathcal{R}$ only accepts types from that bin. Determining the bin B which provides the best expected utility for $\mathcal{R}$ and setting $\varphi_{ij} = 1$ for all $(i,j) \in B$ and $\varphi_{ij} = 0$ otherwise yields an acceptance scheme with approximation guarantee $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$.

More precisely, in lines 2-6, the algorithm considers all types in descending order of receiver value until a combined mass of at least $1/2$ is reached. The first type to surpass the combined mass of $1/2$ is put into a bin B_0, the other types are collected in the set $Q = \{(i,j) \mid \varrho_{ij} \geq \varrho_{i'j'} \ \forall (i',j') \notin Q\}$ such that $\sum_{(i,j)\in Q} q_{ij} < 1/2$. The set Q is then divided into $z = \lceil \log_2 \alpha \rceil$ classes depending on their sender utility in line 7 of the algorithm. The classes $\mathcal{C}_1, \ldots, \mathcal{C}_z$ are constructed such that $\mathcal{C}_\ell = \{(i,j) \in Q \mid \xi_{ij} \in [2^{\ell-1}, 2^\ell)\}$ for $\ell = 1, \ldots, z-1$ and $\mathcal{C}_z = \{(i,j) \in Q \mid \xi_{ij} \in [2^{z-1}, \alpha]\}$. This implies that in each class, the lowest and highest sender utilities differ by at most a factor of 2.

In lines 9-13, the classes are combined into bins. Classes are not split up but rather added completely to a bin such that the following two conditions are satisfied.

Algorithm 16: Binning Algorithm for α-bounded $\mathcal{S}$

Input: n distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$
Output: Acceptance Scheme φ

1 Set $U = [n] \times [m]$, $Q = B_0 = \emptyset$, and $q = 0$.
2 **while** $q < 1/2$ **do**
3   Pick $(i,j) \in U$ s.t. θ_{ij} has best utility $\varrho_{ij} \geq \varrho_{i'j'}$ for all $(i',j') \in U$, breaking ties arbitrarily.
4   **if** $q + q_{ij} \geq 1/2$ **then** Add (i,j) to B_0.
5   **else** Add (i,j) to Q.
6   Remove (i,j) from U, update $q = q + q_{ij}$.
7 Construct $z = \lceil \log_2 \alpha \rceil$ classes $\mathcal{C}_1, \ldots, \mathcal{C}_z$ such that $\mathcal{C}_\ell = \{(i,j) \in Q \mid \xi_{ij} \in [2^{\ell-1}, 2^\ell)\}$ for all $\ell = 1, \ldots, z-1$ and $\mathcal{C}_z = \{(i,j) \in Q \mid \xi_{ij} \in [2^{z-1}, \alpha]\}$.
8 Set $b = 1$, $s = z$, and $y = 0$. Open bin 1 and set $B_1 = \emptyset$.
9 **for** $\ell = z, \ldots, 1$ **do**
10   **if** $2^{\ell-1} < 2^s \cdot \sum_{(i,j) \in B_b \cup \mathcal{C}_\ell} q_{ij}$ **then**
11   Set $b = b + 1$ and $s = \ell$.
        `//` $\sum_{(i,j) \in Q} q_{ij} < 1/2$`, so no open bin stays empty`
12   Open the new bin b and set $B_b = \emptyset$.
13   Add class $\mathcal{C}_\ell$ to bin $B_b = B_b \cup \mathcal{C}_\ell$.
14 Set $b^* = \arg\max_{b=0,1,\ldots} \sum_{(i,j) \in B_b} q_{ij} \cdot \varrho_{ij}$, the index of the best bin for $\mathcal{R}$.
15 Set $\varphi_{ij} = 1$ for all $(i,j) \in B_{b^*}$ and $\varphi_{ij} = 0$ otherwise.
16 **return** φ

1. Each bin holds as many types as possible.
2. For each bin B, the sender's expected utility is maximized by proposing the first type from B rather than waiting for a different type from bin B in a later round.

In Theorem 5.4, we show the main result of this section.

Theorem 5.4
If the sender has α-bounded utilities, there is a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$-approximation of the expected utility for optimal (online) search. The scheme can be computed in time polynomial in the input size.

The proof follows from two lemmas and the following observation regarding its running time. For simplicity, let us assume that all numbers have constant size. Otherwise, the logarithmic size of the numbers needs to be taken into account when considering the running time. Since these numbers are given as input, this does not change the polynomial running time. The while-loop in lines 2-6 considers at most all $n \cdot m$ types. The construction of the classes uses only those types identified in the while-loop. Thus, this only requires time at most $O(n \cdot m)$. Finally, the binning of the classes again requires that at most all types identified in the while-loop are considered. Overall, we get a running time of Algorithm 16 of at most $O(n \cdot m)$. Lemma 5.5 shows that the approximation guarantee for $\mathcal{R}$ is at least $\frac{1}{4r}$ if r bins are opened in Algorithm 16.

Then, Lemma 5.6 bounds the number of bins opened by showing that $r = O\left(\frac{\log \alpha}{\log \log \alpha}\right)$. Combining these results shows the theorem.

Lemma 5.5
Let r be the number of bins opened in Algorithm 16. Then the scheme computed by the algorithm obtains at least an $\frac{1}{4r}$-approximation of the expected utility of the best option for $\mathcal{R}$ in hindsight.

Proof. After the while-loop in lines 2-6, types with a combined mass of $q \geq 1/2$ have been considered by the algorithm, i.e., $\sum_{(i,j) \in Q \cup B_0} q_{ij} \geq 1/2$. Hence, $2 \cdot \sum_{(i,j) \in Q \cup B_0} q_{ij} \geq 1$. Since $Q \cup B_0$ consist of the options with the highest utility for $\mathcal{R}$, we know

$$2 \cdot \sum_{(i,j) \in Q \cup B_0} q_{ij} \cdot \varrho_{ij} \geq \mathbb{E}_{\theta_i \sim \mathcal{D}_i}\left[\max_{i \in [n]} \varrho(\theta_i)\right] = \text{OPT} \quad .$$

Now consider the construction of the bins in the for-loop in lines 9-13. Suppose we split Q into $r-1$ bins $B_1, B_2, \ldots, B_{r-1}$. In the end, we choose B_{b^*}, the best one among the r bins $B_0, \ldots, B_{r-1}$. This implies

$$\sum_{(i,j) \in B_{b^*}} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{r} \cdot \sum_{(i,j) \in Q \cup B_0} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{2r} \cdot \text{OPT} \quad . \tag{5.1}$$

The resulting acceptance scheme φ restricts attention to B_{b^*} and accepts each proposed option θ_{ij} from the bin with probability 1. Clearly, if $b^* = 0$, there exists only a single type from a single round. Hence, $\mathcal{S}$ will propose an action with that type. Thus, we assume $b^* > 0$. Let $\ell^- = \min\{\ell \mid \mathcal{C}_\ell \subseteq B_{b^*}, \mathcal{C}_\ell \neq \emptyset\}$ be the (nonempty) class of smallest index in B_{b^*}, and ℓ^+ the one with largest index, respectively. Now suppose the sender observes θ_{ij} in round i with $(i,j) \in B_{b^*}$. Then, $\mathcal{S}$ decides to propose this option. If $\mathcal{S}$ proposes, $\mathcal{R}$ will accept and $\mathcal{S}$ gets a utility ξ_{ij}. Otherwise, the sender can only wait for another type from B_{b^*} in a later round. This is not profitable for $\mathcal{S}$ since the expected utility from a later round can be upper bounded by ξ_{ij} as follows.

$$\begin{aligned}
\sum_{(i',j') \in B_{b^*}, i' > i} q_{i'j'} \cdot \xi_{i'j'} &\leq \sum_{(i',j') \in B_{b^*}, i' > i} q_{i'j'} \cdot 2^{\ell^+} \\
&< 2^{\ell^+} \cdot \sum_{(i',j') \in B_{b^*}} q_{i'j'} \\
&\leq 2^{\ell^- - 1} \leq \xi_{ij}
\end{aligned}$$

First, we use that in B_{b^*}, all types have a utility for $\mathcal{S}$ of at most 2^{ℓ^+}. Then, we bound the combined mass in rounds $i' > i$ by the total mass of the bin B_{b^*}. By the condition in line 10 on the construction of the bins, we have $2^{\ell^- - 1} \geq 2^{\ell^+} \cdot \sum_{(i',j') \in B_{b^*}} q_{i'j'}$ and since ℓ^- is the index of the class with the smallest index in B_{b^*}, all types have a utility at least $2^{\ell^- - 1}$. Hence, the first type from bin B_{b^*} that is realized also gets proposed by $\mathcal{S}$ and accepted by $\mathcal{R}$.

For each $(i,j) \in B_{b^*}$, the probability that type θ_{ij} is proposed and accepted requires the following two events to occur.

1. No type $\theta_{i'j'}$ with $(i',j') \in B_{b^*}$ was proposed in a round $i' < i$, and

2. θ_{ij} is realized in round i.

Clearly, these events are independent and the second event has probability q_{ij}. For the first event, if $b^* = 0$, the probability is 1 as B_0 only holds a single type. For $b^* > 0$, define $\mu_i = \sum_{(i,j)\in B_{b^*}} q_{ij}$, the mass of all types in round i in B_{b^*}. Then, with probability $\prod_{i'<i}(1-\mu_{i'})$, no type from bin B_{b^*} was realized in an earlier round. Since $\sum_{i=1}^n \mu_i \leq \sum_{(i,j)\in Q} q_{ij} < 1/2$, we can minimize the expression $\prod_{i'<i}(1-\mu_{i'})$ for all $i > 1$ by setting $\mu_1 = 1/2$, $\mu_{i'} = 0$ for all $i' < i$. Clearly, this means that $\prod_{i=1}^n (1-\mu_i) \geq 1/2$ and the first event has a probability at least $1/2$.

Combining this with (5.1), φ guarantees $\mathcal{R}$ an expected utility of at least

$$\sum_{(i,j)\in B_{b^*}} \frac{1}{2} \cdot q_{ij} \cdot \varrho_{ij} = \frac{1}{2r} \cdot \sum_{(i,j)\in Q\cup B_0} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{4r} \cdot \mathrm{OPT} \quad .$$

This proves the lemma. □

Lemma 5.6

Let r be the number of bins opened in the for-loop of Algorithm 16. It holds that $r = O\left(\frac{\log \alpha}{\log\log \alpha}\right)$.

Proof. We use the terminology of the algorithm, i.e., $z = \lceil \log_2 \alpha \rceil$ is the number of classes constructed by Algorithm 7. For a bin B, we denote the combined mass of types in B by $q_B = \sum_{(i,j)\in B} q_{ij}$.

We want to show that $O\left(\frac{z}{\log z}\right)$ many bins are opened in the algorithm. We restrict attention to the number of bins opened during the for-loop in lines 9-13, as the single additional bin B_0 does not have any impact on the asymptotic number of bins opened in the algorithm. We condition on having opened r bins in the for-loop and then lower bound the number of classes covered using these r bins.

Now, consider a bin B which starts with class $\mathcal{C}_s$, i.e. s is the highest index of a class inside B. Classes $\mathcal{C}_\ell$ are added to B until $2^{\ell-1} < 2^s \cdot q_B$, or, equivalently, $s - \ell + 1 > \log_2 (1/q_B)$. This means that B holds at least $\log_2 (1/q_B)$ many classes.

Now, consider bins B_i and B_{i+1} and fix $\hat{q} = q_{B_i} + q_{B_{i+1}}$. By the previous paragraph, the bins contain at least $-\log_2 (q_{B_i}) - \log_2 (\hat{q} - q_{B_i})$ many classes. We can find an extreme point of this lower bound by taking the derivative with respect to q_{B_i} and solving for a root. This gives us

$$-\frac{1}{q_{B_i} \cdot \ln 2} + \frac{1}{\hat{q} - q_{B_i} \cdot \ln 2} = 0 \quad .$$

Clearly, $q_{B_i} = \hat{q}/2$ is a solution. Since the second derivative $\frac{1}{{q_{B_i}}^2} + \frac{1}{(\hat{q}-q_{B_i})^2}$ is positive, this means that $q_{B_i} = q_{B_{i+1}} = \hat{q}/2$ is a minimum. Repeatedly applying this balancing step, we see that the lower bound on the number of classes is minimized when $q_{B_i} = q_{B_{i'}}$ for all $i, i' \in [r]$, i.e., $q_{B_i} = \frac{1}{r} \cdot \sum_{(i,j)\in Q} q_{ij} < \frac{1}{2r}$. When we have r open bins, the smallest number of classes inside them is therefore

$$r \cdot \log_2 \left(1 \Big/ \frac{1}{2r}\right) = r \cdot \log_2(2r) = r + r \cdot \log_2 r \quad .$$

This implies that $z \geq r \cdot (\log_2 r + 1)$, otherwise, opening r bins would not be possible while satisfying the lower bound. Hence, we have $z = \Omega(r \log_2 r)$ and hence $r = O\left(\frac{z}{\log z}\right) = O\left(\frac{\log \alpha}{\log\log \alpha}\right)$ as desired. This proves the Lemma. □

Observe that the approximation ratio of this algorithm is tight in general. Consider the instances in Theorem 5.3 with $\alpha = n^{4n-4}$. The theorem shows that every scheme can obtain at most a ratio of $O\left(\frac{1}{n}\right) = O\left(\frac{\log\log\alpha}{\log\alpha}\right)$. We formalize this observation using the following corollary.

Corollary 5.7
There is a class of instances with α-bounded sender utilities in which every acceptance scheme φ obtains at most an $O\left(\frac{\log\log\alpha}{\log\alpha}\right)$-approximation of the expected utility for optimal (online) search.

Let us now discuss how to handle instances which include types with 0 utility for $\mathcal{S}$.

Remark 5.8
Clearly, our definition of α-bounded utilities for $\mathcal{S}$ requires $\xi_{ij} > 0$ for all $i \in [n], j \in [m]$. Still, we can modify Algorithm 16 to accommodate instances in which all strictly positive values for $\mathcal{S}$ are bounded by parameter α. If there are any types θ_{ij} with $\xi_{ij} = 0$ in the set Q of best types for $\mathcal{R}$, we construct another bin B_{-1} which holds all those types, i.e., $B_{-1} = \{(i,j) \in Q \mid \xi_{ij} = 0\}$. If B_{-1} is chosen as the best bin for $\mathcal{R}$, no option provides any utility for $\mathcal{S}$. Due to the tie-breaking in favor of $\mathcal{R}$, $\mathcal{S}$ can be assumed to perform an online search for $\mathcal{R}$. This guarantees $\mathcal{R}$ a $1/2$-approximation with respect to the best option from B_{-1}. The number of bins stays $O\left(\frac{\log\alpha}{\log\log\alpha}\right)$. If B_{-1} is not chosen as the best bin for $\mathcal{R}$, the original analysis can be applied.

Hence, we get the following corollary.

Corollary 5.9
If the sender has α-bounded utilities for all options with strictly positive utility, there is a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{\log\log\alpha}{\log\alpha}\right)$-approximation of the expected utility for optimal (online) search.

In the next subsection, we consider α-bounded sender utility with sender-oblivious proposals. Compared to the current subsection, this means that $\mathcal{R}$ learns a decreased amount of information.

5.2.2 Sender-Oblivious Proposals

Clearly, the upper bound of Corollary 5.7 still holds. In this subsection, we show that the reduced amount of information for $\mathcal{R}$ results in a significant decrease of the upper bound to $O(1/\sqrt{\alpha})$ in Theorem 5.10.

In contrast to the upper bound, our Binning Algorithm cannot be applied directly to this new instance. There might be types which have the same value for $\mathcal{R}$ but very different values for $\mathcal{S}$. In a sender-oblivious proposal scenario, $\mathcal{R}$ cannot distinguish those types and thus cannot attach different acceptance probabilities to them. We describe our *High-/Low-Algorithm* (Algorithm 17) which achieves an approximation guarantee of $\Omega\left(\frac{1}{\sqrt{\alpha}\cdot\log\alpha}\right)$.

We begin with the upper bound.

Theorem 5.10
There is a class of instances of online delegation with IID options, α-bounded utilities for the sender, and sender-oblivious proposals, in which every acceptance scheme φ obtains at most an $O\left(1/\sqrt{\alpha}\right)$-approximation of the expected utility for optimal (online) search.

Proof. We consider the following class of instances in which all n actions have a type drawn independently from the same distribution $\mathcal{D}$, where $\alpha \in [2, n^2]$. There exist only three types with utility values and probabilities as shown in Table 9.

Type	θ_1	θ_2	θ_3
ϱ	0	1	1
ξ	1	2	α
Probability	$1 - \frac{1}{n}$	$\frac{1}{n} - \frac{1}{n\cdot\sqrt{\alpha}}$	$\frac{1}{n\cdot\sqrt{\alpha}}$

Table 9: Distribution $\mathcal{D}$ for the upper bound on α-bounded utilities with sender-oblivious proposals.

Since $\varrho_2 = \varrho_3$, $\mathcal{R}$ does not know whether the current type is θ_2 or θ_3 when one of them is proposed by $\mathcal{S}$. Hence, we have $\varphi_{i2} = \varphi_{i3}$ for all $i \in [n]$. In an optimal scheme, we can assume that $\varphi_{i1} = 0$. Clearly, $\mathcal{R}$ gets no utility from actions with type θ_1 and proposing such an action in some round $i < n$ can only decrease the chance of getting a more profitable action in a later round.

To upper bound the expected utility for $\mathcal{R}$, we assume that $\mathcal{S}$ always proposes an action with type θ_3. Note that, depending on the value of the acceptance probability φ_{i2}, it might be profitable for $\mathcal{S}$ not to propose action i and instead wait for round $i' > i$ with $\varphi_{i'2} > \varphi_{i2}$. If the type of action i is θ_2, $\mathcal{S}$ clearly does not want to propose if the expected utility from observing and proposing an action of type θ_3 in a later round is higher than $2 \cdot \varphi_{i2}$. Hence, we get the following necessary condition for a proposal of θ_2 in round i:

$$\begin{aligned} 2 \cdot \varphi_{i2} &\geq \alpha \cdot \left(\sum_{\ell=i+1}^{n} \left(1 - \frac{1}{n\cdot\sqrt{\alpha}}\right)^{\ell-i-1} \cdot \frac{1}{n\cdot\sqrt{\alpha}} \cdot \varphi_{\ell 2} \right) \\ &= \frac{\sqrt{\alpha}}{n} \cdot \sum_{\ell=i+1}^{n} \left(1 - \frac{1}{n\cdot\sqrt{\alpha}}\right)^{\ell-i-1} \cdot \varphi_{\ell 2} \ . \end{aligned} \tag{5.2}$$

We use δ_i to denote whether inequality (5.2) is satisfied, i.e., we set $\delta_i = 1$ if (5.2) holds and $\delta_i = 0$ otherwise. With this notation, we can upper bound the receiver's utility when employing acceptance scheme φ by

$$\sum_{i=1}^{n} \varphi_{i2} \cdot \left(\frac{1}{n\cdot\sqrt{\alpha}} + \delta_i \cdot \left(\frac{1}{n} - \frac{1}{n\cdot\sqrt{\alpha}} \right) \right) \ . \tag{5.3}$$

We denote by $i^* = \min\{i \in [n] \mid \delta_i = 1\}$ the first round for which (5.2) is satisfied. Since $\varphi_{i^*2} \leq 1$, we get

$$2 \geq \frac{\sqrt{\alpha}}{n} \cdot \sum_{\ell=i^*+1}^{n} \left(1 - \frac{1}{n\cdot\sqrt{\alpha}}\right)^{\ell-i^*-1} \cdot \varphi_{\ell 2}$$

$$\geq \frac{\sqrt{\alpha}}{n} \cdot \sum_{\ell=i^*+1}^{n} \left(1 - \frac{1}{n \cdot \sqrt{\alpha}}\right)^n \cdot \varphi_{\ell 2}$$
$$\geq \frac{\sqrt{\alpha}}{n} \cdot \left(1 - \frac{1}{\sqrt{\alpha}}\right) \cdot \sum_{\ell=i^*+1}^{n} \varphi_{\ell 2} \ ,$$

where the last inequality follows from Bernoulli's inequality, i.e., $(1+x)^n \geq 1 + x \cdot n$ for all $x \geq -1$ and every $n \in \mathbb{N}$. Hence, we have

$$\sum_{\ell=i^*+1}^{n} \varphi_{\ell 2} \leq \frac{2n}{\sqrt{\alpha}} \cdot \left(1 - \frac{1}{\sqrt{\alpha}}\right)^{-1} \leq \frac{2n}{\sqrt{\alpha}} \cdot \left(1 - \frac{1}{\sqrt{2}}\right)^{-1} = \frac{2n}{\sqrt{\alpha}} \cdot \left(2 + \sqrt{2}\right) \leq \frac{7n}{\sqrt{\alpha}} \ .$$

For the second inequality, we use that $\alpha \in [2, n^2]$ and thus $(1 - 1/\sqrt{\alpha})^{-1} \leq (1 - 1/\sqrt{2})^{-1}$. For the equality, observe that $(1 - 1/\sqrt{2}) \cdot (2 + \sqrt{2}) = 1$. Plugging this result into the upper bound (5.3), the utility of $\mathcal{R}$ is upper bounded by

$$\begin{aligned}
\frac{1}{n} \cdot \sum_{i=1}^{n} \varphi_{i2} \cdot \left[\frac{1}{\sqrt{\alpha}} + \delta_i \cdot \left(1 - \frac{1}{\sqrt{\alpha}}\right)\right] &= \sum_{i=1}^{i^*-1} \frac{\varphi_{i2}}{n \cdot \sqrt{\alpha}} + \sum_{i=i^*}^{n} \frac{\varphi_{i2}}{n} \cdot \left[\frac{1}{\sqrt{\alpha}} + \delta_i \cdot \left(1 - \frac{1}{\sqrt{\alpha}}\right)\right] \\
&\leq \sum_{i=1}^{i^*-1} \frac{\varphi_{i2}}{n \cdot \sqrt{\alpha}} + \frac{\varphi_{i^*2}}{n} + \sum_{i=i^*+1}^{n} \frac{\varphi_{i2}}{n} \\
&\leq \frac{1}{n} \cdot \left[\frac{i^* - 1}{\sqrt{\alpha}} + 1 + \frac{7n}{\sqrt{\alpha}}\right] \\
&= O\left(\frac{1}{\sqrt{\alpha}}\right) \ ,
\end{aligned}$$

where we used that $\delta_i \leq 1$ and $\varphi_{i2} \leq 1$ for all $i \in [n]$.

An online search finds an action with type θ_2 or θ_3 whenever such a type exists, i.e., with probability $1 - (1 - 1/n)^n \geq 1 - 1/e$. This shows that $\mathcal{R}$ cannot get more than an $O\left(1/\sqrt{\alpha}\right)$-approximation for this class of instances. □

Before discussing our algorithm to compute a good acceptance scheme for the setting of α-bounded sender utilities with sender-oblivious proposals, we remark on types with sender-utility 0. For this setting, no better approximation ratio than $O\left(1/n\right)$ can be guaranteed, even for $\alpha = 1$. We discuss this in the following remark.

Remark 5.11
We consider an adaption of the IID instance in the proof of Theorem 5.10. There are three types, but only a single one provides positive utility to $\mathcal{S}$.

Type	θ_1	θ_2	θ_3
ϱ	0	1	1
ξ	0	0	1
Probability	$1 - \frac{1}{n}$	$\frac{1}{n} - \frac{1}{n^2}$	$\frac{1}{n^2}$

Clearly, only θ_3 has positive utility for $\mathcal{S}$ and thus $\alpha = 1$. If $\mathcal{R}$ performs a one-dimensional online search, the expected utility is at least $1 - 1/e$. As before, we can assume that in an optimal scheme $\varphi_{i1} = 0$ for all i. Additionally, it holds that $\varphi_{i2} = \varphi_{i3}$.

Algorithm 17: High-/Low-Algorithm for Sender-Oblivious Proposals

Input: n distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$

Output: Acceptance Scheme φ

Set $U = [n] \times [m]$.

Partition U into $U_L = \left\{ (i,j) \in U \;\middle|\; \sum_{\substack{t=1\\ \varrho_{it}=\varrho_{ij}}}^{m} q_{it} \cdot \xi_{it} < \sqrt{\alpha} \cdot \sum_{\substack{t=1\\ \varrho_{it}=\varrho_{ij}}}^{m} q_{it} \right\}$ and $U_H = U \setminus U_L$.

for $\ell = 1, \ldots, n$ **do**

  Set $\mathcal{D}_\ell^{(L)} = \mathcal{D}_\ell$ and $\mathcal{D}_\ell^{(H)} = \mathcal{D}_\ell$

  In $\mathcal{D}_\ell^{(L)}$, for every $(\ell, j) \in U_H$, set $\varrho_{\ell j} = 0$ and $\xi_{\ell j} = 1$.

  In $\mathcal{D}_\ell^{(H)}$, for every $(\ell, j) \in U_L$, set $\varrho_{\ell j} = 0$ and $\xi_{\ell j} = \sqrt{\alpha}$.

Set $\varphi_L = \mathit{AlgoLow}(\mathcal{D}_1^{(L)}, \ldots, \mathcal{D}_n^{(L)})$ and $\varphi_H = \mathit{AlgoHigh}(\mathcal{D}_1^{(H)}, \ldots, \mathcal{D}_n^{(H)})$.

return φ_L or φ_H whichever yields better expected utility for $\mathcal{R}$

$\mathcal{S}$ never wants to propose an action with type θ_2 if there is still a positive probability that an action with type θ_3 can be proposed and is taken by $\mathcal{R}$. The expected utility for $\mathcal{R}$ from any randomized acceptance scheme can be upper bounded by using the following deterministic acceptance scheme with $\varphi_{i1} = 0$, $\varphi_{i2} = 1$ for all $i \in [n]$. This means that $\mathcal{S}$ always proposes the first action with type θ_3. Additionally, type θ_2 is proposed in round n. Using a union bound, we can upper bound the expected utility for $\mathcal{R}$ by $\frac{1}{n^2} \cdot n + \frac{1}{n} - \frac{1}{n^2} = \frac{2}{n} - \frac{1}{n^2} = O\left(\frac{1}{n}\right)$.

Let us shortly argue why this is an upper bound on any randomized acceptance scheme. Clearly, type θ_2 is only proposed in the final round i after which $\varphi_{i'2} = 0$ for $i' > i$. Additionally, if $\varphi_{i2} < 1$, it might be in the sender's interest not to propose θ_3 in round i if the expected utility from a later round is better. This is not in the interest of $\mathcal{R}$ who can incentivize $\mathcal{S}$ to propose by increasing φ_{i2}. Additionally, under the assumption that a good type for $\mathcal{R}$ was proposed, not taking it with a positive probability can only hurt the overall expected utility.

Hence, we have shown that regardless of α, every acceptance scheme is $O(1/n)$-approximate.

For our lower bound, we use our *High-/Low-Algorithm* (Algorithm 17). The algorithm guarantees an approximation ratio of $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$.

Theorem 5.12

If the sender has α-bounded utilities and makes sender-oblivious proposals, there is a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation of the expected utility for optimal (online) search. The scheme can be computed in time polynomial in the input size.

We shortly describe the algorithm and the subroutines it uses. The proof of Theorem 5.12 then follows from an analysis of the subroutines.

Clearly, the High-/Low-Algorithm needs to take into account that in the scenario with sender-oblivious proposals, $\mathcal{R}$ cannot distinguish different types $\theta_{ij} \neq \theta_{ij'}$ if $\varrho_{ij} = \varrho_{ij'}$. Hence, for each round we consider the sets of types with the same utility for $\mathcal{R}$. We consider the *sender expectation* of such a subset, i.e., the expected utility for $\mathcal{S}$,

Algorithm 18: AlgoLow

Input: n distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$, where in every distribution individually, the set of options with the same value for $\mathcal{R}$ has a conditional sender expectation of less than $\sqrt{\alpha}$

Output: Acceptance Scheme φ

Set $Q = \text{RestrictTypes}(\mathcal{D}_1, \dots, \mathcal{D}_n, {}^1\!/_2)$.
Set $\ell = 1$, $q_1 = 0$, $B_1 = \emptyset$.
for $i = 1, \dots, n$ **do**
  Set $q^* = \sum_{(i,j) \in Q} q_{ij}$.
  if $q_\ell + q^* > {}^1\!/\!\sqrt{\alpha}$ **then** Set $\ell = \ell + 1$, $B_\ell = \{(i,j) \in Q\}$, $q_\ell = q^*$.
  else Add $B_\ell = B_\ell \cup \{(i,j) \in Q\}$.
Set $\varrho_{\ell'} = \sum_{(i,j) \in B_{\ell'}} q_{ij} \varrho_{ij}$ for all $1 \le \ell' \le \ell$.
Choose ℓ^* such that $\varrho_{\ell^*} \ge \varrho_{\ell'}$ for all $1 \le \ell' \le \ell$.
Set $\varphi_{ij} = 1$ for all $(i,j) \in B_{\ell^*}$.
return φ

conditional on a type from that subset being drawn. We say that a subset has *low sender expectation* if the sender expectation is less than $\sqrt{\alpha}$. Otherwise, the subset has *high sender expectation*.

Our algorithm partitions the set of types into sets U_L and U_H, depending on their sender expectation. Based on these sets, the algorithm creates two modified instances, namely $\mathcal{D}^{(L)} = \left(\mathcal{D}_1^{(L)}, \dots, \mathcal{D}_n^{(L)}\right)$ and $\mathcal{D}^{(H)} = \left(\mathcal{D}_1^{(H)}, \dots, \mathcal{D}_n^{(H)}\right)$. In $\mathcal{D}^{(L)}$, all subsets of types with high sender expectation are excluded from consideration. More formally, they are replaced with types that do not provide any utility for $\mathcal{R}$ and only minimal utility 1 for $\mathcal{S}$. Similarly, in $\mathcal{D}^{(H)}$, all subsets of types with low sender expectation are replaced with types that provide no utility for $\mathcal{R}$ and minimal utility $\sqrt{\alpha}$ for $\mathcal{S}$. These modifications ensure that the subsets of types with the same $\mathcal{S}$ utility either all have low or all have high sender expectation.

The algorithm then uses two different subroutines. The first one, AlgoLow (Algorithm 18), is run on the instance $\mathcal{D}^{(L)}$. AlgoLow produces an acceptance scheme φ_L which achieves an $\Omega\left({}^1\!/\!\sqrt{\alpha}\right)$-approximation in the modified instance $\mathcal{D}^{(L)}$.

Lemma 5.13
If the sender has α-bounded utilities, makes sender-oblivious proposals, and all options have low sender expectation, AlgoLow (Algorithm 18) constructs a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left({}^1\!/\!\sqrt{\alpha}\right)$-approximation of the expected utility for optimal (online) search. The algorithm can be implemented to run in a time polynomial in the input size.

The second subroutine of our High-/Low-Algorithm, AlgoHigh (Algorithm 19), is given $\mathcal{D}^{(H)}$ as input. For this instance, an acceptance scheme φ_H is computed which guarantees an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation in $\mathcal{D}^{(H)}$.

Lemma 5.14
If the sender has α-bounded utilities, makes sender-oblivious proposals, and all options have high sender expectation, AlgoHigh (Algorithm 19) constructs a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation of the expected

Algorithm 19: AlgoHigh

Input: n distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$, where in every distribution individually, the set of options with the same value for $\mathcal{R}$ has a conditional sender expectation of at least $\sqrt{\alpha}$

Output: Acceptance Scheme φ

Set $Q = \text{RestrictTypes}(\mathcal{D}_1, \ldots, \mathcal{D}_n, 1/4)$ and $z = \lceil \log_2 \sqrt{\alpha} \rceil$.

for $\ell = 1, \ldots, z-1$ **do**

  Set $\mathcal{C}_\ell = \left\{ (i,j) \in Q \;\middle|\; \frac{\sum_{(i,j') \in Q, \varrho_{ij} = \varrho_{ij'}} q_{ij} \xi_{ij}}{\sum_{(i,j') \in Q, \varrho_{ij} = \varrho_{ij'}} q_{ij}} \in \left[\sqrt{\alpha} \cdot 2^{\ell-1}, \sqrt{\alpha} \cdot 2^{\ell}\right) \right\}$

  Set $\varrho_\ell = \sum_{(i,j) \in \mathcal{C}_\ell} q_{ij} \varrho_{ij}$.

Set $\mathcal{C}_z = \left\{ (i,j) \in Q \;\middle|\; \frac{\sum_{(i,j') \in Q, \varrho_{ij} = \varrho_{ij'}} q_{ij} \xi_{ij}}{\sum_{(i,j') \in Q, \varrho_{ij} = \varrho_{ij'}} q_{ij}} \in \left[\sqrt{\alpha} \cdot 2^{z-1}, \alpha\right] \right\}$

Set $\varrho_z = \sum_{(i,j) \in \mathcal{C}_z} q_{ij} \cdot \varrho_{ij}$.

Choose $\ell^* = \arg\max_{\ell \in [z]} \varrho_\ell$.

Set $\varphi_{ij} = 1$ for all $(i,j) \in \mathcal{C}_{\ell^*}$.

return φ

utility for optimal (online) search. The algorithm can be implemented to run in a time polynomial in the input size.

We prove both Lemma 5.13 and Lemma 5.14 below.

Proof of Theorem 5.12. By Lemma 5.13, φ_L guarantees an $\Omega\left(1/\sqrt{\alpha}\right)$-approximation in the sub-instance $\mathcal{D}^{(L)}$ in which only types with low sender expectation are considered. Similarly, by Lemma 5.14, φ_H guarantees an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation in the remaining sub-instance of the remaining types with high sender expectation $\mathcal{D}^{(H)}$. Clearly, the sub-instance with the higher expected maximum for $\mathcal{R}$ obtains at least $1/2$ of the expected maximum of the original instance.

The High-/Low-Algorithm chooses either φ_L or φ_H, depending on which acceptance scheme provides the higher expected utility for $\mathcal{R}$. This guarantees $\mathcal{R}$ an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation of the optimal expected utility when performing an optimal one-dimensional (online) search. $\square_{\text{Theorem 5.12}}$

Both AlgoLow and AlgoHigh use the procedure RestrictTypes (Algorithm 20). It is given a parameter $\mu \leq 1$ and identifies a set Q with the best types for $\mathcal{R}$ such that $\sum_{(i,j) \in Q} q_{ij} \cdot \varrho_{ij} \geq \mu/2 \cdot \text{OPT}$, where OPT denotes the expected maximum for $\mathcal{R}$ among all n actions. Additionally, the combined probability mass of the types in Q is less than μ, i.e., $\sum_{(i,j) \in Q} q_{ij} < \mu$, or all types in Q can only be drawn in a single round. The algorithm works similar to the start of our Binning Algorithm, where a set of good types was identified. The main difference to RestrictTypes is that in the Binning Algorithm, types were considered individually. This is no longer possible, all types with the same value for $\mathcal{R}$ in a single round have to be considered at the same time.

Lemma 5.15

For distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and a parameter $0 < \mu \leq 1$, RestrictTypes$(\mathcal{D}_1, \ldots, \mathcal{D}_n, \mu)$

Algorithm 20: RestrictTypes

Input: n distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$, value μ restricting the mass
Output: Set Q of good options for $\mathcal{R}$

1 Set $Q = \emptyset$, $q = q^* = 0$, $U = [n] \times [m]$.
2 **while** $q < \mu$ **do**
3   Set $U^* = \emptyset$, $q^* = 0$.
4   **for** $\ell = 1, \ldots, n$ **do**
5   Let $U^*_\ell = \{(\ell, j) \in U \mid \varrho_{\ell j} \geq \varrho_{i'j'}$ for all $(i', j') \in U\}$ be the types in round ℓ from the set of all remaining types in all rounds with the best utility for $\mathcal{R}$.
6   Set $q_\ell = \sum_{(i,j) \in U^*_\ell} q_{ij}$.
7   **if** $q^* + q_\ell < \mu$ **then** Add $U^* = U^* \cup U^*_\ell$, update $q^* = q^* + q_\ell$.
8   **else**
9   **if** $q_\ell > q^*$ **then** Set $U^* = U^*_\ell$.
10   **break for-loop**
11   Set $q^* = \sum_{(i,j) \in U^*} q_{ij}$ and $\varrho^* = \varrho_{ij}$ for $(i, j) \in U^*$.
12   **if** $q + q^* > \mu$ **then** Set $B = U^*$.
13   **else** Add $Q = Q \cup U^*$.
14   Remove $U = U \setminus U^*$, update $q = q + q^*$.
15 Set $\varrho_Q = \sum_{(i,j) \in Q} q_{ij} \cdot \varrho_{ij}$ and $\varrho_B = q^* \cdot \varrho^*$.
16 **if** $\varrho_Q < \varrho_B$ **then** Set $Q = B$.
17 **return** Q

identifies Q, the best set of types for $\mathcal{R}$, such that

$$\sum_{(i,j) \in Q} q_{ij} \cdot \varrho_{ij} \geq \frac{\mu}{2} \cdot \mathbb{E}_{\theta_i \sim \mathcal{D}_i} \left[\max_{i \in [n]} \varrho(\theta_i) \right]$$

and either (1) the combined mass $\sum_{(i,j) \in Q} q_{ij} < \mu$ or (2) Q contains only types from a single round. RestrictTypes can be implemented to run in a time polynomial in the input size.

Proof of Lemma 5.15. Similar to the Binning Algorithm, RestrictTypes considers the types by descending value for $\mathcal{R}$. Since types with the same value for $\mathcal{R}$ in a single round cannot be distinguished, they are considered together. In the while-loop in lines 2-14, the procedure identifies two sets Q and B which contain the best types for $\mathcal{R}$. As long as the combined mass of types considered is less than μ, the types are added to set Q. Once this threshold is exceeded by a set of types, the types from this final set are put into set B. Hence, the following holds. The combined utility mass of Q is less than μ, but the combined mass of $Q \cup B$ is at least μ. Note that B only contains types from a single round, which guarantees that if Q or B are returned, either (1) or (2) must hold.

Since the set $Q \cup B$ contains the best types for $\mathcal{R}$, by line 15 we have

$$\frac{1}{\mu} \cdot \sum_{(i,j) \in Q \cup B} q_{ij} \cdot \varrho_{ij} \geq \mathbb{E}_{\theta_i \sim \mathcal{D}_i} \left[\max_{i \in [n]} \varrho(\theta_i) \right] .$$

In lines 15 and 16, the individual values obtained by Q and B are compared. If B guarantees a better expected utility for $\mathcal{R}$, the set Q is overwritten by B. Hence, the set Q returned by RestrictTypes guarantees

$$\frac{2}{\mu} \cdot \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij} \geq \mathbb{E}_{\theta_i \sim \mathcal{D}_i}\left[\max_{i\in[n]} \varrho(\theta_i)\right] .$$

When determining the set Q, in each round, at least one type is added to Q and removed from U. Hence, we have at most $n \cdot m$ rounds. In each round, the best remaining type among all n actions needs to be identified, which can easily be done in time n when types are sorted by descending receiver utility. Sorting all m types for each distribution takes time $n \cdot m \cdot \log m$. Clearly, this results in an overall running time polynomial in the input size. $\square_{\text{Lemma 5.15}}$

Before showing the proof of Lemma 5.13, we give a short intuitive description of AlgoLow. The set Q returned by RestrictTypes is split into $O(\sqrt{\alpha})$ many bins. Each bin consist of consecutive rounds and contains all types from the set Q that are in the respective rounds. The bins are constructed such that $\mathcal{S}$ is incentivized to propose the first action with a type from a bin instead of waiting for an action with a better type from the same bin. The resulting scheme φ then deterministically accepts only types from the best bin.

Proof of Lemma 5.13. By Lemma 5.15, RestrictTypes($\mathcal{D}_1, \dots, \mathcal{D}_n, 1/2$) in line 1 of the algorithm returns a set Q such that $4 \cdot \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij} \geq \mathbb{E}_{\theta_i\sim\mathcal{D}_i}\left[\max_{i\in[n]} \varrho(\theta_i)\right] = \text{OPT}$, the expected maximum for $\mathcal{R}$. In the for-loop in lines 3-6, a new bin is opened if the combined mass of the current bin and the set of types from the currently considered round i is greater than $1/\sqrt{\alpha}$. Hence, either the previous bin or the new bin have a probability mass of at least $\frac{1}{2\sqrt{\alpha}}$. Since the set Q has a total mass of at most 1 this means that at most $2\sqrt{\alpha}$ many bins are opened.

Now, assume that AlgoLow chooses bin B, i.e., B provides the highest value for $\mathcal{R}$. Consider the case that $\mathcal{S}$ has not proposed any actions and observes an action i with type $\theta \in B$. We know $\xi(\theta) \geq 1$ by the assumption that sender-utilities are α-bounded. Additionally, B either has a combined mass of at most $\frac{1}{\sqrt{\alpha}}$ or B only spans a single round. Hence, the probability that another type from bin B is observed in a future round is at most $\frac{1}{\sqrt{\alpha}}$. Since all types in B have low sender expectation, the conditional sender expectation of a type in B is at most $\sqrt{\alpha}$. This implies that $\mathcal{S}$ proposes action i.

Similar to our Binning Algorithm, for each $(i, j) \in B$, the probability that θ_{ij} in round i is proposed requires the following two independent events to occur.

1. No type $\theta_{i'j'}$ with $(i', j') \in B$ was proposed in some round $i' < i$, and
2. θ_{ij} is realized in round i.

Using the same arguments as in the proof of Lemma 5.5, we see that round i is reached without a proposal with probability at least $1/2$. Since θ_{ij} is drawn with probability q_{ij}, this means that $\mathcal{R}$ achieves an expected utility of at least $1/2 \cdot \sum_{(i,j)\in B} q_{ij} \cdot \varrho_{ij}$.

Since the number of bins is at most $2\sqrt{\alpha}$ and the one with the highest value for $\mathcal{R}$ is chosen, this means that $\frac{1}{2} \cdot \sum_{(i,j)\in B} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{4\sqrt{\alpha}} \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij}$. Hence, $\mathcal{R}$ gets an expected utility of at least

$$\frac{1}{2} \cdot \sum_{(i,j)\in B} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{4\sqrt{\alpha}} \cdot \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{16\sqrt{\alpha}} \cdot \mathrm{OPT} = \Omega\left(\frac{1}{\sqrt{\alpha}}\right) \cdot \mathrm{OPT} \quad .$$

Since RestrictTypes only requires a polynomial running time and the remaining steps are using only the types in Q a constant number of times, Algorithm 18 can be run in polynomial time with respect to the input size. $\square_{\text{Lemma 5.13}}$

In contrast to the grouping of good types by the rounds in which they can be observed in AlgoLow, AlgoHigh classifies the types by their expected utility for $\mathcal{S}$. Similar to our Binning Algorithm, in a class of types constructed by AlgoHigh, the expected utility for $\mathcal{S}$ differs by at most a factor 2. This means that there are $O(\log \alpha)$ many classes from which the best one is chosen. A notable difference to the previous algorithms is that due to the high sender expectation of the types, $\mathcal{S}$ cannot be incentivized to propose the first type from the chosen class with probability 1. Rather, we show that the probability is at least $\frac{1}{2\sqrt{\alpha}}$. In combination, this gives the approximation guarantee of $\Omega\left(\frac{1}{\sqrt{\alpha}\cdot\log\alpha}\right)$ of AlgoHigh.

Proof of Lemma 5.14. Again, we use $\mathrm{OPT} = \mathbb{E}_{\theta_i\sim\mathcal{D}_i}\left[\max_{i\in[n]} \varrho(\theta_i)\right]$ to denote the expected maximum value for $\mathcal{R}$ among the n actions. By Lemma 5.15, we know that for the set Q returned by RestrictTypes$(\mathcal{D}_1, \ldots, \mathcal{D}_n, 1/4)$ in line 1, it holds that $8 \cdot \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij} \geq \mathrm{OPT}$. The algorithm then partitions the set Q into $z = \lceil \log_2 \sqrt{\alpha} \rceil$ classes. The types inside each class differ by at most a factor 2 in their conditional expectation for $\mathcal{S}$. The best class for $\mathcal{R}$ is chosen, i.e., a class $\mathcal{C}$ such that

$$\sum_{(i,j)\in\mathcal{C}} q_{ij} \cdot \varrho_{ij} = \max_{\ell\in[z]} \sum_{(i,j)\in\mathcal{C}_\ell} q_{ij} \cdot \varrho_{ij} \quad .$$

For the receiver expectation, this implies

$$z \cdot \sum_{(i,j)\in\mathcal{C}} q_{ij} \cdot \varrho_{ij} \geq \sum_{(i,j)\in Q} q_{ij} \cdot \varrho_{ij} \geq \frac{1}{8} \cdot \mathrm{OPT} \quad .$$

Let E denote the lower bound on the expected sender utility for $\mathcal{C}$. Recall that we consider α-bounded sender utilities. As such, all sender values are in the interval $[1, \alpha]$. Hence, an expected utility of at least E implies that with probability at least $\frac{E}{2\alpha - E}$, a random element from $\mathcal{C}$ has a sender utility value of at least $E/2$. If that were not the case, the overall expected utility would necessarily be less than E, which is a contradiction to the definition of $\mathcal{C}$.

We choose parameter $\mu = 1/4$ in the call to RestrictTypes. Hence, the probability that another type from the same class is observed in a later round is at most $1/4$ by similar arguments to those in the proof of Lemma 5.5. In combination with the conditional expectation of a future type from $\mathcal{C}$ being upper bounded by $2E$, this means that $\mathcal{S}$ always proposes an action which provides a sender utility of at least $E/2$. Therefore, $\mathcal{S}$ always proposes the first action with an acceptable type with a probability of at least $\frac{E}{2\alpha - E} \geq \frac{E}{2\alpha}$. Since we consider only types with high sender expectation, we have $E \geq \sqrt{\alpha}$ and thus we have a proposal probability of at least $\frac{1}{2\sqrt{\alpha}}$ for the first type from class $\mathcal{C}$ that is observed.

Hence, by using the acceptance scheme computed by AlgoHigh, $\mathcal{R}$ gets an expected utility of at least

$$\begin{aligned} \frac{1}{2\sqrt{\alpha}} \cdot \sum_{(i,j)\in\mathcal{C}} q_{ij} \cdot \varrho_{ij} &\geq \frac{1}{2\sqrt{\alpha}} \cdot \frac{1}{8 \cdot z} \cdot \mathrm{OPT} \\ &= \frac{1}{16\sqrt{\alpha} \cdot \lceil \log_2 \sqrt{\alpha} \rceil} \cdot \mathrm{OPT} \\ &= \Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right) \mathrm{OPT} \quad . \end{aligned}$$

We have already seen that the running time of RestrictTypes is polynomial in the input size. The set Q resulting from the call to the subroutine consists of a subset of the input. Classifying the types in Q by sender utility and determining the best class only considers each type in Q a constant number of times. Overall, this means that AlgoHigh can be implemented to run in a time polynomial in the input size. $\square_{\text{Lemma 5.14}}$

This concludes the section on α-bounded sender utility values. In the next section, we consider a different parameter β, which bounds the ratios between utility values of individual types for $\mathcal{S}$ and $\mathcal{R}$.

5.3 β-bounded Ratios of Utility Values

In the previous section, we considered the case that the sender's values are bounded by parameter α. The receiver's utility values on the other hand can be arbitrary. In this section, we bound the ratios of different types for $\mathcal{S}$ and for $\mathcal{R}$. Intuitively, this means that – up to factor β – a type θ_{ij} that is better for $\mathcal{S}$ than another type $\theta_{i'j'}$ is also better for $\mathcal{R}$. Formally, we consider the following. Let $\beta \geq 1$ be the smallest number such that

$$\frac{1}{\beta} \cdot \frac{\xi_{ij}}{\xi_{i'j'}} \leq \frac{\varrho_{ij}}{\varrho_{i'j'}} \leq \beta \cdot \frac{\xi_{ij}}{\xi_{i'j'}} \quad \text{for all } i, i' \in [n], \text{ and all } j, j' \in [m]. \tag{5.4}$$

We say that an instance satisfying (5.4) has *β-bounded utility ratios.*

Clearly, (5.4) requires the utility values for both $\mathcal{S}$ and $\mathcal{R}$ to be strictly positive. As we did in the previous section on α-bounded sender utilities, we will remark on instances where there exist types which have a sender utility value or a receiver utility value of 0. For now, we assume that all utility values are positive.

Note that we can normalize the input by choosing an arbitrary type $\theta_{i'j'}$ as reference. Dividing all receiver-values by $\varrho_{i'j'}$ and all sender-values by $\xi_{i'j'}$, we get

$$\frac{1}{\beta} \cdot \frac{\varrho_{ij}}{\xi_{ij}} \leq 1 \leq \beta \cdot \frac{\varrho_{ij}}{\xi_{ij}} \quad ,$$

or, put differently,

$$\frac{1}{\beta} \leq \frac{\xi_{ij}}{\varrho_{ij}} \leq \beta \quad \text{for all } i \in [n], \text{ and all } j \in [m] \quad . \tag{5.5}$$

Note that (5.5) always holds for an instance with β-bounded utility ratios but it is not a sufficient condition. To see this, consider the following two types with $\beta > 1$. θ_1 has $\varrho_1 = 1$, $\xi_1 = \beta$ and θ_2 has $\varrho_2 = 1$, $\xi_2 = 1/\beta$. Clearly, both θ_1 and θ_2 satisfy (5.5) but $\frac{\varrho_1}{\varrho_2} = 1 < \frac{1}{\beta} \cdot \frac{\xi_1}{\xi_2} = \frac{1}{\beta} \cdot \frac{\beta}{1/\beta} = \beta$, which violates (5.4).

As a first acceptance scheme, consider the following *Lazy Acceptance Scheme* which does not restrict the sender in any way, i.e., it sets $\varphi_{ij} = 1$ for all $i \in [n]$ and all $j \in [m]$. Clearly, this leads to $\mathcal{S}$ performing an optimal one-dimensional online search with respect to the values $\xi(\theta_i)$.

Proposition 5.16
If the instance has β-bounded utility ratios, the Lazy Acceptance Scheme guarantees $\mathcal{R}$ an $\Omega(1/\beta)$-approximation of the expected utility for optimal (online) search.

Proof. Consider the following random variables. We denote by $\theta_{\mathcal{X}}$ the type of the action $\mathcal{X} \in \{\mathcal{S}, \mathcal{R}\}$ chooses in an optimal one-dimensional online search for their respective best utility. By definition of β-bounded utility ratios, we have

$$\beta \cdot \frac{\xi(\theta_{\mathcal{R}})}{\xi(\theta_{\mathcal{S}})} \geq \frac{\varrho(\theta_{\mathcal{R}})}{\varrho(\theta_{\mathcal{S}})} \quad .$$

Since this holds pointwise for every realization, it also holds when going over to the expectation. Hence, we get

$$\beta \cdot \frac{\mathbb{E}[\xi(\theta_{\mathcal{R}})]}{\mathbb{E}[\xi(\theta_{\mathcal{S}})]} \geq \frac{\mathbb{E}[\varrho(\theta_{\mathcal{R}})]}{\mathbb{E}[\varrho(\theta_{\mathcal{S}})]} \quad .$$

Clearly $\mathbb{E}[\xi(\theta_{\mathcal{S}})] \geq \mathbb{E}[\xi(\theta_{\mathcal{R}})]$, which implies

$$\mathbb{E}[\varrho(\theta_{\mathcal{S}})] \geq \frac{1}{\beta} \cdot \frac{\mathbb{E}[\xi(\theta_{\mathcal{S}})]}{\mathbb{E}[\xi(\theta_{\mathcal{R}})]} \cdot \mathbb{E}[\varrho(\theta_{\mathcal{R}})] \geq \frac{1}{\beta} \cdot \mathbb{E}[\varrho(\theta_{\mathcal{R}})] \quad .$$

Hence, the expected utility for $\mathcal{R}$ when $\mathcal{S}$ performs a one-dimensional online search is at least $1/\beta$ of the expected utility $\mathcal{R}$ would obtain when performing the search. □

Clearly, the Lazy Acceptance Scheme does not require any knowledge of the utility values, its approximation guarantee stems from the β-bounded utility ratios. This implies that for $\mathcal{S}$-oblivious proposals, the Lazy Acceptance Scheme can be used to obtain an $\Omega(1/\beta)$-approximation. Note that the upper bound of Theorem 5.10 can easily be adapted to β-bounded utility ratios, even for sender-oblivious proposals. We can set $\beta = \alpha$. To ensure that all values are strictly positive, we can also increase $\varrho(\theta_1)$ to $1/\beta$. This gives us the following corollary.

Corollary 5.17
There is a class of instances of online delegation with IID options, β-bounded utility ratios, and sender-oblivious proposals, in which every acceptance scheme φ obtains at most an $O(1/\sqrt{\beta})$-approximation of the expected utility for optimal (online) search.

Proof. The proof uses the same argumentation as the proof of Theorem 5.10. We use the following adaptation of the distribution $\mathcal{D}$.

Algorithm 21: β-Classification Algorithm

Input: n distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$
Output: Acceptance Scheme φ

Set $U = [n] \times [m]$ and $z = \lceil \log_2 \beta \rceil$.
for $\ell = -z+1$ *to* $z-1$ **do**
  Set $\mathcal{C}_\ell = \left\{ (i,j) \in U \;\middle|\; \frac{\varrho_{ij}}{\xi_{ij}} \in \left[2^{\ell-1}, 2^{\ell}\right) \right\}$.
Set $\mathcal{C}_z = \left\{ (i,j) \in U \;\middle|\; \frac{\varrho_{ij}}{\xi_{ij}} \in [2^{z-1}, \beta] \right\}$.
Choose $\ell^* = \arg\max_{\ell \in \{-z+1,\dots,z\}} \varrho(\mathcal{S}, \ell)$, where $\varrho(\mathcal{S}, \ell)$ is the expected utility for $\mathcal{R}$ when $\mathcal{S}$ performs an optimal online search on class $\mathcal{C}_\ell$.
Set $\varphi_{ij} = 1$ for all $(i,j) \in \mathcal{C}_{\ell^*}$.
return φ

Type	θ_1	θ_2	θ_3
ϱ	$1/\beta$	1	1
ξ	1	2	β
Probability	$1 - \frac{1}{n}$	$\frac{1}{n} - \frac{1}{n \cdot \sqrt{\beta}}$	$\frac{1}{n \cdot \sqrt{\beta}}$

Table 10: Distribution $\mathcal{D}$ for the upper bound on β-bounded utility ratios with sender-oblivious proposals.

Note that in contrast to the proof of Theorem 5.10, in this adapted instance, θ_1 does provide some positive utility to $\mathcal{R}$. Still, by assuming that $\varphi_{i1} = 0$ for all $i \in [n]$, $\mathcal{R}$ loses at most an additive utility of $\frac{1}{\beta} = o\left(1/\sqrt{\beta}\right)$. Hence, the overall upper bound of $O\left(1/\sqrt{\beta}\right)$ still holds. □

Remark 5.18
Note that the upper bound of Remark 5.11 can be applied literally by replacing α by β. Hence, for sender-oblivious proposals and types with utility 0 for $\mathcal{S}$, there is an upper bound of $O\left(1/n\right)$, even for $\beta = 1$ as bound on the ratios of types with strictly positive utility values.

If all types with strictly positive utility values have β-bounded ratios and there are no types with utility 0 for $\mathcal{S}$, the Lazy Acceptance Scheme can be applied with a slight modification, i.e., all types θ with $\varrho(\theta) = 0$ are not acceptable.

Finally, let us discuss the case that $\mathcal{R}$ learns the values for $\mathcal{S}$ when receiving a proposal.

5.3.1 Sender-Aware Proposals

For the case of sender-aware proposals, we improve the Lazy Acceptance Scheme using our β-Classification Algorithm (Algorithm 21). The main idea is still to let $\mathcal{S}$ perform an optimal online search. Instead of accepting all types, $\mathcal{R}$ first classifies them by their utility ratios and then accepts only types from the best class. Since there are $2 \cdot \lceil \log_2 \beta \rceil = O(\log \beta)$ many classes, the best one among them guarantees an $\Omega\left(\frac{1}{\log \beta}\right)$-approximation. We formalize the result in the following theorem.

Theorem 5.19
For instances with β-bounded utility ratios, there is a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{1}{\log\beta}\right)$-approximation of the expected utility for optimal (online) search. The scheme can be computed in a time polynomial in the input size.

Proof. Clearly, the algorithm considers every type only a constant number of times. Hence, it runs in time polynomial in the input size.

For the proof of the approximation ratio, consider the case that only types from a class $\mathcal{C}_\ell$ are accepted by $\mathcal{R}$. We denote by $\varrho(\mathcal{S},\ell)$ and $\xi(\mathcal{S},\ell)$ the expected utility for $\mathcal{R}$ and $\mathcal{S}$, respectively. Note that $\mathcal{S}$ uses an optimal online search with respect to the types which $\mathcal{R}$ accepts. Similarly, we denote by $\varrho(\mathcal{R},\ell)$ and $\xi(\mathcal{R},\ell)$ the expected utility for $\mathcal{R}$ and $\mathcal{S}$ of the best type for $\mathcal{R}$ from class $\mathcal{C}_\ell$ – if such a type exists. Clearly, $\varrho(\mathcal{R},\ell)$ is the result for $\mathcal{R}$ in an optimal offline search restricted to class $\mathcal{C}_\ell$.

This implies $\varrho(\mathcal{R},\ell) \geq \varrho(\mathcal{S},\ell)$ and $\xi(\mathcal{S},\ell) \geq \xi(\mathcal{R},\ell)$. Furthermore, since $\frac{\varrho_{ij}}{\xi_{ij}} \in \left[2^{\ell-1}, 2^{\ell}\right]$ for all $(i,j) \in \mathcal{C}_\ell$, we have

$$\varrho(\mathcal{S},\ell) \geq \xi(\mathcal{S},\ell) \cdot 2^{\ell-1} \geq \xi(\mathcal{R},\ell) \cdot 2^{\ell-1} \geq \frac{\varrho(\mathcal{R},\ell)}{2} \quad .$$

Now, consider OPT, the best value for $\mathcal{R}$ among all n actions, not restricted to a single class. Clearly, the best type for $\mathcal{R}$ from a class $\mathcal{C}_\ell$ does not have to be the best type overall. Thus, we have

$$\text{OPT} \leq \sum_{\ell=-z+1}^{z} \varrho(\mathcal{R},\ell) \leq 2 \cdot \sum_{\ell=-z+1}^{z} \varrho(\mathcal{S},\ell) \ ,$$

where we used $z = \lceil \log_2 \beta \rceil$ as defined in the algorithm. Our algorithm chooses the class $\mathcal{C}_{\ell^*}$ that maximizes $\varrho(\mathcal{S},\ell^*)$. This implies

$$\varrho(\mathcal{S},\ell^*) \geq \frac{1}{4\log_2\beta} = \Omega\left(\frac{1}{\log\beta}\right),$$

which concludes the proof of the theorem. □

We have already seen that types with utility 0 for $\mathcal{S}$ are problematic in the case of sender-oblivious proposals. Let us now discuss sender-aware proposals and the implications of types with utility 0 for $\mathcal{S}$ (or $\mathcal{R}$).

Remark 5.20
Clearly, each type θ with $\varrho(\theta) = 0$ should be ignored and rejected by $\mathcal{R}$. Hence, they do not pose a problem.

In the sender-aware setting, $\mathcal{R}$ is able to distinguish all individual types. This allows $\mathcal{R}$ to group all types θ with $\xi(\theta) = 0$ in an additional class $\mathcal{C}_\perp$. If only that particular class is accepted, $\mathcal{S}$ is indifferent among the possible types. Due to our tie-breaking assumption, this means that $\mathcal{S}$ will perform an optimal online search for $\mathcal{R}$. Similar to the other classes, this means that $\mathcal{R}$ loses at most a factor of 2 with respect to the optimal utility from $\mathcal{C}_\perp$ in hindsight. For the original classes, the analysis remains unchanged. In total, the number of classes increases by 1, i.e., there are still $O(\log\beta)$ many classes. Choosing the best one with respect to the utility for $\mathcal{R}$ still guarantees the receiver an $\Omega\left(\frac{1}{\log\beta}\right)$-approximation.

Hence, we get the following two corollaries. The first one generalizes the result of Theorem 5.19 to instances with types that have utility 0 for $\mathcal{R}$, the second one shows the general upper bound of Theorem 5.3.

Corollary 5.21
If receiver and sender have β-bounded utilities for the set of types with only strictly positive utilities, there is a deterministic acceptance scheme such that $\mathcal{R}$ obtains an $\Omega\left(\frac{1}{\log \beta}\right)$-approximation of the expected utility for optimal (online) search.

Corollary 5.22
There is a class of instances with β-bounded utility ratios in which every acceptance scheme φ obtains at most an $O\left(\frac{\log \log \beta}{\log \beta}\right)$-approximation of the expected utility for optimal (online) search.

This concludes our discussion of online delegated search.

Chapter 6

Conclusion

In the thesis, we discuss different settings of strategic communication with commitment power between two self-interested, rational agents, a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. $\mathcal{S}$ is able to observe the state of nature and thus has an informational advantage. $\mathcal{R}$, on the other hand, only sees the signals sent by $\mathcal{S}$ and eventually chooses one out of n actions without knowing the exact type. The type of the chosen action then determines the utility for both $\mathcal{S}$ and $\mathcal{R}$. The algorithms we discuss are always designed from the perspective of the agent with commitment power.

6.1 Bayesian Persuasion

In the first part, the sender has the power to commit to a signaling scheme φ. Such a scheme determines the probability for each signal $\sigma \in \Sigma$, depending on the realized state of nature. More formally, for each state of nature $\boldsymbol{\theta}$, $\varphi(\boldsymbol{\theta}, \sigma)$ denotes the probability that σ is sent. This allows $\mathcal{R}$ to perform a Bayesian update and infer information on $\boldsymbol{\theta}$.

6.1.1 Offline Bayesian Persuasion

We start our discussion of Bayesian persuasion with an offline setting, i.e., $\mathcal{S}$ can observe all n actions and their types simultaneously. We describe a polynomial-time algorithm based on geometric interpretations of the states of nature. The algorithm computes an optimal scheme for $\mathcal{S}$ for some classes of symmetric instances. The result also holds if the signal space has a limited size of $k < n$. Our algorithm requires an efficient probability oracle for the symmetric distribution. We show the existence of such oracles for two classes of symmetric instances, namely d-random order and prophet secretary instances. Our results also imply that Dughmi and Xu's solution for IID instances translates to the scenario of limited signals. It would be very interesting to see if there are further classes of symmetric instances for which optimal signaling schemes can be computed in polynomial time – either by showing that they allow for an efficient probability oracle or via a new approach.

For independent instances, we identify a condition of ϱ_E-optimality, where ϱ_E denotes the maximum a priori expectation for $\mathcal{R}$ among the n actions. An instance satisfies ϱ_E-optimality if there exists an optimal scheme that guarantees $\mathcal{R}$ a conditional expectation of at least ϱ_E for each signal $\sigma \in \Sigma$. For ϱ_E-optimal instances, we show a polynomial-time algorithm that computes our Independent Scheme φ_{IS} in two separate steps. The first step consists of identifying a good subset S of k actions to recommend using the k signals. In the second step, the signaling scheme is computed for the set S. Using φ_{IS}, $\mathcal{S}$ can obtain a constant-factor approximation of at least $(1 - 1/e)^2$ of the optimal utility. We are then further able to improve this approach using an FPTAS for step 1, which, for every constant $\varepsilon > 0$ is able to compute a $(1-\varepsilon)\cdot(1-1/e)$-approximately optimal scheme in polynomial time. The factor $(1-1/e)$ is tight using step 2 of our two-step approach. Essentially, this means that the constant factor achievable for ϱ_E-optimal independent instances cannot be improved using our strategy. Unfortunately, our approach cannot be extended to independent instances which do not satisfy ϱ_E-optimality. This leaves two interesting questions for further research: Does a different approach allow for a better approximation ratio for ϱ_E-optimal instances in polynomial time? For which additional classes of independent instances – if any – is a constant-factor approximation in polynomial time possible?

We close out the chapter on offline Bayesian persuasion with approximation results for instances with restricted signal spaces with respect to the unrestricted counterparts. For symmetric instances, we show a tight k/n-approximation when $\mathcal{S}$ has access to only k instead of n signals. For ϱ_E-optimal independent instances, we show asymptotically matching upper and lower bounds of $\Theta\left(k/n\right)$. Clearly, this means that access to very few signals can substantially hurt the sender's expected utility.

Besides open question with respect to constraints on the number of signals, the model of Bayesian persuasion can be extended in different interesting directions. Especially the question of tractability of (near-)optimal signaling schemes is yet to be answered for many settings, including, among others, scenarios with multiple receivers (with and without externalities) with whom a single sender interacts.

6.1.2 Online Bayesian Persuasion

The next chapter discusses an online variant of Bayesian persuasion. We measure the performance of the online sender with respect to that of an offline sender. In the first section, we consider independent actions similar to the offline setting. Although an optimal signaling scheme can be determined in polynomial time, there are instances in which such a scheme cannot achieve an approximation ratio greater than 0, i.e., the optimal offline scheme extracts some positive utility while any online scheme can only get a utility of 0. Therefore, we consider a restricted class of independent instances. This class requires two conditions, one of which is ϱ_E-optimality, the other one the existence of a canonical deviation option for $\mathcal{R}$. For such instances, we design a simple scheme that guarantees a $1/2$-approximation of the optimal utility. This matches the best possible bound for the one-dimensional prophet inequality problem for general independent instances.

The second section on online Bayesian persuasion assumes that types are drawn in uniform random order. We discuss 16 different variants with varying levels of information as well as two different objectives for both $\mathcal{S}$ and $\mathcal{R}$. In the scenarios with cardinal receiver utility, we describe the Pareto Mechanism which utilizes a geometric interpretation of the instance and requires polynomial time for its computation. Using this mechanism, $\mathcal{S}$ achieves the same expected utility as in the corresponding offline scenario. Hence, this can be used as a benchmark for the remaining scenarios. We denote the expected utility or the success probability in the basic case with OPT. When utility values are unknown, we use our Growing Pareto Mechanism with the growing set of types observed. This gives $\mathcal{S}$ an expected utility of at least $\mathrm{OPT} \cdot \left(\frac{1}{3\sqrt{3}} - o(1)\right)$ in the cardinal utility case and a success probability of $\mathrm{OPT} \cdot (1/4 - o(1))$ in the ordinal utility case. These bounds are not tight, the one-dimensional secretary problem gives an asymptotic upper bound of $1/e$ on the probability that the best type is chosen. In the basic setting with disclosure, we first show an algorithm to compute an optimal mechanism using an exponential number of linear programs. In terms of efficient algorithms, we describe another variant of the Pareto Mechanism which uses the shrinking set of remaining types. Using the Shrinking Pareto Mechanism, an ordinal sender achieves a success probability of $\mathrm{OPT} \cdot (1/2 - o(1))$, and we show that this is asymptotically tight. For a sender with cardinal objective, the upper bound of $\mathrm{OPT} \cdot (1/2 + o(1))$ still holds. Our Shrinking Pareto Mechanism achieves an approximation ratio of $1/3 - o(1)$ with respect to the utility obtained in the corresponding basic scenario. Unlike all previously discussed settings with constant approximation factors, in the secretary scenario with disclosure, we show that there are instances in which $\mathcal{S}$ can only achieve an approximation ratio in $O(1/n)$ with respect to the optimal success probability or expected utility, respectively. Using our Trivial Mechanism which always recommends the first action, we show an asymptotically matching lower bound of $\Omega(1/n)$.

In the scenarios with ordinal receiver utility, all bounds are asymptotically tight. In the basic scenario, $\mathcal{S}$ can get the best type with probability $1 - o(1)$, regardless of whether the instance is with or without disclosure. In the secretary scenarios, the utility for $\mathcal{S}$ decreases. Interestingly, we can describe a mechanism that asymptotically matches the performance of the one-dimensional secretary algorithm of $1/e$, i.e., the receiver does not substantially decrease the sender's expected utility or success probability. In the most challenging setting, namely the secretary scenario with disclosure, we show a success probability of $1/4 - o(1)$, drastically increasing the approximation ratio with respect to the corresponding cardinal receiver setting. Interestingly, this also guarantees $\mathcal{R}$ a success probability of $1/4 - o(1)$ since the scheme is entirely symmetric with respect to the agents' utility values. From the receiver's side, this is the only scenario in which the scheme actually increases the success probability. All other settings ensure only a success probability of $1/n$, matching the success probability of $\mathcal{R}$ blindly picking one of the actions. Similarly, in the cardinal receiver setting, all schemes only guarantee that $\mathcal{R}$ gets an expected utility of ϱ_E, i.e., no more than the expected utility $\mathcal{R}$ would get when choosing an action without additional information.

It would be an interesting extension of our research to see whether an optimal signaling scheme in the basic setting with disclosure and cardinal receiver utility can be computed efficiently. Clearly, closing the gaps between our lower and upper bounds would also solve an open question.

Extensions of the model provide another interesting direction for further research. Studying extensions of the classic secretary problem or the prophet inequality problem within the domain of Bayesian persuasion opens a plethora of interesting questions to answer. To give just a few examples, consider a receiver who wants to take multiple actions with respect to some packing constraint, or an online model with recourse, i.e., decisions are not necessarily irrevocable, but it might be costly to retract them. Additionally, already the one-dimensional variants without the added separation between information acquisition and decision-making provide interesting questions for future research.

6.2 Online Delegated Search

In the second part of the thesis, the receiver has commitment power and commits to an acceptance scheme φ. Such a scheme determines for each $(i, j) \in [n] \times [m]$ the probability φ_{ij} that $\mathcal{R}$ accepts θ_{ij} if an action with that type is proposed by $\mathcal{S}$. As our first result, we show that there are IID instances in which the receiver cannot hope for more than an $O(1/n)$-approximation of the utility when performing a one-dimensional search. This stands in striking contrast to the result of Kleinberg and Kleinberg [58], who showed a constant-factor approximation for IID instances in offline delegated search. We further show that this bound is tight, i.e., we give an acceptance scheme that always guarantees an $\Omega(1/n)$-approximation. The scheme simply accepts the action with the highest a priori expectation for $\mathcal{R}$.

Since the IID instance which shows the low approximation guarantee requires a ratio of sender utilities which is exponential in n, we study instances bounded by two natural parameters. In the first instance, the ratio of sender utility values is bounded by α. For such instances, we describe our Binning Algorithm which computes an acceptance scheme that guarantees an approximation ratio of $\Omega\left(\frac{\log \log \alpha}{\log \alpha}\right)$ with respect to the utility of the best option in hindsight in polynomial time. Since $\frac{\log \log \alpha}{\log \alpha} = \Theta\left(\frac{1}{n}\right)$ for $\alpha = n^{\Theta(n)}$, this bound is tight.

For parameter β, which bounds the ratio of utility values for $\mathcal{S}$ in terms of the ratio of utility values for $\mathcal{R}$, we describe an algorithm that computes an acceptance scheme guaranteeing an $\Omega\left(\frac{1}{\log \beta}\right)$-approximation. This matches to upper bound of $O\left(\frac{\log \log \beta}{\log \beta}\right)$ up to a factor of $\log \log \beta$.

Additionally, we study the above settings with less information for $\mathcal{R}$, namely using sender-oblivious proposals. In such a scenario, $\mathcal{R}$ does not learn the sender's utility value when an action is proposed. This implies that types in a single round which have the same value for $\mathcal{R}$ cannot be distinguished. Hence, such types are required to have the same acceptance probability. In this setting, the bound of $\Theta(1/n)$ still holds. For parameter α, we describe an algorithm that achieves an $\Omega\left(\frac{1}{\sqrt{\alpha} \cdot \log \alpha}\right)$-approximation, matching the upper bound of $O\left(1/\sqrt{\alpha}\right)$ up to a factor of $\log \alpha$. For parameter β, the discrepancy is greater. The instance showing the upper bound of $O\left(1/\sqrt{\alpha}\right)$ easily translates, hence, we get an upper bound of $O\left(1/\sqrt{\beta}\right)$. For our lower bound, the receiver is guaranteed an $\Omega\left(1/\beta\right)$-approximation when accepting every single type with probability 1. For both parameters, the loss of information results in an exponential deterioration of approximation ratios, i.e., from logarithmic guarantees to polynomial ones.

Overall, the results for the parameterized instances provide (near-)optimal approximation ratios and can be computed in polynomial time. Still, it would be very interesting to close the open gaps, especially for the polynomial gap of $\sqrt{\beta}$ for sender-oblivious proposals when instances are bounded by parameter β.

A natural extension to our sender-oblivious model is to consider an even stronger assumption, namely, $\mathcal{R}$ not having any information on the values for $\mathcal{S}$. Clearly, for general instances, our bound of $\Theta(1/n)$ holds. Is the receiver still able to obtain a good approximation with respect to some – possibly new – parameter on the instance?

Clearly, our assumption that the process ends after the first proposal – even if the action is rejected – is rather strict. It would be interesting to see what $\mathcal{R}$ and $\mathcal{S}$ can do in more general models. We leave this question open for future work.

Additionally, although the sender always tries and performs an online search for the best type, the receiver can drastically reduce the value of this best option for $\mathcal{S}$ via the acceptance scheme and no general guarantees can be made with respect to the expected utility for $\mathcal{S}$. The sender might not be motivated to work very hard when the expected utility is very low. This leads to very interesting questions of how to effectively incentivize $\mathcal{S}$ to do work for $\mathcal{R}$. A potential remedy for $\mathcal{S}$ and a possible direction of research is an extension of the model to multiple receivers that all solicit information from a single sender. This way, the sender only has to gather information once and can use the information on multiple occasions. This gives rise to a host of questions regarding possible competition between receivers, i.e., whether the actions to take are unique, or the possible objective functions of the sender.

In contrast to this, a single receiver might want to introduce competition on the sender side. It would be interesting to see how much – if at all – $\mathcal{R}$ can profit from consulting several senders with their own individual interests.

Clearly, the open questions discussed above for online Bayesian persuasion with respect to extensions of classic online problems can also be studied from the perspective of delegated search, i.e., where the receiver has commitment power.

Bibliography

[1] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi HajiAghayi, Robert Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. In *Proc. 49th Symp. Theory Comput. (STOC)*, STOC 2017, pages 61–71, New York, NY, USA, 2017. Association for Computing Machinery.

[2] Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.*, 43(2):930–972, 2014.

[3] Saed Alizamir, Francis de Véricourt, and Shouqiang Wang. Warning against recurring risks: An information design approach. *Manag. Sci.*, 66(10):4359–4919, 2020.

[4] Ricardo Alonso and Odilon Câmara. Persuading voters. *Amer. Econ. Rev.*, 106(11):3590–3605, 2016.

[5] Ricardo Alonso and Niko Matouschek. Optimal delegation. *Rev. Econ. Stud.*, 75(1):259–293, 2008.

[6] Manuel Amador and Kyle Bagwell. The theory of optimal delegation with an application to tariff caps. *Econometrica*, 81(4):1541–1599, 2013.

[7] Attila Ambrus and Georgy Egorov. Delegation and nonmonetary incentives. *J. Econ. Theory*, 171:101 – 135, 2017.

[8] Jerry Anunrojwong, Krishnamurthy Iyer, and Vahideh Manshadi. Information design for congested social services: Optimal need-based persuasion. In *Proc. 21st Conf. Econ. Comput. (EC)*, EC '20, page 349–350, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Itai Arieli and Yakov Babichenko. Private Bayesian persuasion. *J. Econ. Theory*, 182:185–217, 2019.

[10] Mark Armstrong and John Vickers. A model of delegated project choice. *Econometrica*, 78(1):213–244, 2010.

[11] Pak Hung Au. Dynamic information disclosure. *RAND J. Econ.*, 46(4):791–823, 2015.

[12] Robert Aumann and Michael Maschler. Game theoretic aspects of gradual disarmament. *Report of the US Arms Control and Disarmament Agency*, 80:1–55, 1966.

[13] Yunus Aybas and Eray Turkel. Persuasion with coarse communication. *arXiv*, abs/1910.13547, 2021.

[14] Yakov Babichenko and Siddharth Barman. Algorithmic aspects of private Bayesian persuasion. In *Proc. 8th Innov. Theoret. Comput. Sci. (ITCS)*, pages 34:1–34:16, 2017.

[15] Yakov Babichenko, Inbal Talgam-Cohen, Haifeng Xu, and Konstantin Zabarnyi. Multi-channel Bayesian persuasion. *CoRR*, abs/2111.09789, 2021.

[16] Yakov Babichenko, Inbal Talgam-Cohen, Haifeng Xu, and Konstantin Zabarnyi. Regret-minimizing Bayesian persuasion. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *Proc. 22nd Conf. Econ. Comput. (EC)*, page 128. ACM, 2021.

[17] Ashwinkumar Badanidiyuru, Kshipra Bhawalkar, and Haifeng Xu. Targeting and signaling in ad auctions. In *Proc. 29th Symp. Discret. Algorithms (SODA)*, pages 2545–2563, 2018.

[18] Thomas Bayes and Richard Price. An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions (1683-1775)*, 53:370–418, 1763.

[19] Curtis Bechtel and Shaddin Dughmi. Delegated stochastic probing. In James R. Lee, editor, *Proc. 12th Innov. Theoret. Comput. Sci. (ITCS)*, pages 37:1–37:19, 2021.

[20] Martin Beckmann. Dynamic programming and the secretary problem. *Comput. Math. Appl.*, 19(11):25–28, 1990.

[21] Dirk Bergemann, Benjamin Brooks, and Stephen Morris. The limits of price discrimination. *Amer. Econ. Rev.*, 105(3):921–57, March 2015.

[22] Dirk Bergemann and Stephen Morris. Information design: A unified perspective. *J. Econ. Literature*, 57(1):44–95, 2019.

[23] Umang Bhaskar, Yu Cheng, Young Kun Ko, and Chaitanya Swamy. Hardness results for signaling in Bayesian zero-sum and network routing games. In *Proc. 17th Conf. Econ. Comput. (EC)*, pages 479–496, 2016.

[24] Ozan Candogan. Information design in operations. *Informs Tutorials*, pages 176–201, 2020.

[25] Matteo Castiglioni, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Online Bayesian persuasion. In *Proc. 34th Conf. Adv. Neural Inf. Processing Syst. (NeurIPS)*, pages 16188–16198, 2020.

[26] Matteo Castiglioni, Alberto Marchesi, Andrea Celli, and Nicola Gatti. Multi-receiver online Bayesian persuasion. In *Proc. 38th Int. Conf. Machine Learning (ICML)*, pages 1314–1323, 2021.

[27] Archishman Chakraborty and Rick Harbaugh. Persuasive puffery. *Marketing Science*, 33(3):382–400, 2014.

[28] Shuchi Chawla, Jason Hartline, David Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proc. 42nd Symp. Theory Comput. (STOC)*, pages 311–320, 2010.

[29] José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms for a random stream of customers. In *Proc. 18th Conf. Econ. Comput. (EC)*, pages 169–186, 2017.

[30] Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50(6), 1982.

[31] Francis de Véricourt, Huseyin Gurkan, and Shouqiang Wang. Informing the public about a pandemic. *Manag. Sci.*, 67(10):6350–6357, 2021.

[32] Kimon Drakopoulos, Shobhit Jain, and Ramandeep Randhawa. Persuading customers to buy early: The value of personalized information provisioning. *Manag. Sci.*, 67(2):828–853, 2021.

[33] Shaddin Dughmi. Algorithmic information structure design: a survey. *SIGecom Exchanges*, 15(2):2–24, 2017.

[34] Shaddin Dughmi, David Kempe, and Ruixin Qiang. Persuasion with limited communication. In *Proc. 17th Conf. Econ. Comput. (EC)*, pages 663–680, 2016.

[35] Shaddin Dughmi, Rad Niazadeh, Alexandros Psomas, and S. Matthew Weinberg. Persuasion and incentives through the lens of duality. In *Proc. 15th Conf. Web and Internet Econ. (WINE)*, pages 142–155, 2019.

[36] Shaddin Dughmi and Haifeng Xu. Algorithmic Bayesian persuasion. In *Proc. 48th Symp. Theory Comput. (STOC)*, pages 412–425, 2016.

[37] Shaddin Dughmi and Haifeng Xu. Algorithmic persuasion with no externalities. In *Proc. 18th Conf. Econ. Comput. (EC)*, pages 351–368, 2017.

[38] Eugene Dynkin. The optimum choice of the instant for stopping a Markov process. In *Sov. Math. Dokl*, volume 4, pages 627–629, 1963.

[39] Jeffrey Ely. Beeps. *Amer. Econ. Rev.*, 107(1):31–53, 2017.

[40] Jeffrey Ely, Alexander Frankel, and Emir Kamenica. Suspense and surprise. *J. Political Econ.*, 123(1):215–260, 2015.

[41] Yuval Emek, Michal Feldman, Iftah Gamzu, Renato Paes Leme, and Moshe Tennenholtz. Signaling schemes for revenue maximization. *ACM Trans. Econ. Comput.*, 2(2):5:1–5:19, 2014.

[42] Françoise Forges. Games with incomplete information: From repetition to cheap talk and persuasion. *Ann. Econ. Stat.*, 137:3–30, 2020.

[43] Ronen Gradwohl, Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. Algorithms for persuasion with limited communication. In *Proc. 32nd Symp. Discret. Algorithms (SODA)*, pages 637–652, 2021.

[44] Ronen Gradwohl, Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. Reaping the informational surplus in Bayesian persuasion. *Amer. Econ. J. Micro.*, Forthcoming.

[45] Jerry R. Green and Nancy Stokey. A two-person game of information transmission. Harvard Institute of Economic Research Discussion Paper No. 751, 1980.

[46] Jerry R. Green and Nancy L. Stokey. A two-person game of information transmission. *J. Econ. Theory*, 135:90–104, 2007.

[47] Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. Prophet inequalities for Bayesian persuasion. In *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, pages 175–181, 2020.

[48] Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. The secretary recommendation problem. In *Proc. 21st Conf. Econ. Comput. (EC)*, page 189, 2020.

[49] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[50] Theodore P Hill and Robert P Kertz. A survey of prophet inequalities in optimal stopping theory. *Contemp. Math*, 125:191–207, 1992.

[51] Martin Hoefer, Pasin Manurangsi, and Alexandros Psomas. Algorithmic persuasion with evidence. In *Proc. 12th Innov. Theoret. Comput. Sci. (ITCS)*, pages 3:1–3:20, 2021.

[52] Bengt Robert Holmström. *On Incentives and Control in Organizations.* PhD thesis, Stanford University, 1977.

[53] Bengt Robert Holmström. On the theory of delegation. In *Bayesian Models in Economic Theory*, pages 115–141. Elsevier, 1984.

[54] Emir Kamenica. Bayesian persuasion and information design. *Ann. Rev. Econ.*, 11(1):249–272, 2019.

[55] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *Amer. Econ. Rev.*, 101(6):2590–2615, 2011.

[56] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems.* Springer, 2013.

[57] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Proc. 21st European Symp. Algorithms (ESA)*, pages 589–600, 2013.

[58] Jon Kleinberg and Robert Kleinberg. Delegated search approximates efficient search. In *Proc. 19th Conf. Econ. Comput. (EC)*, pages 287–302, 2018.

[59] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, chapter 3, pages 71–104. Cambridge University Press, 2014.

[60] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bull. Amer. Math. Soc*, 83:745–747, 1977.

[61] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts and processes with finite value. *Adv. Prob.*, 4:197–266, 1978.

[62] Can Küçükgül, Özalp Özer, and Shouqiang Wang. Engineering social learning: Information design of time-locked sales campaigns for online platforms. *Manag. Sci.*, Forthcoming.

[63] Maël Le Treust and Tristan Tomala. Persuasion with limited communication capacity. *J. Econ. Theory*, 184:104940, 2019.

[64] David Lingenbrink and Krishnamurthy Iyer. Optimal signaling mechanisms in unobservable queues. *Oper. Res.*, 67(5):1397–1416, 2019.

[65] Nahum D. Melumad and Toshiyuki Shibano. Communication in settings with no transfers. *The RAND Journal of Economics*, 22(2):173–198, 1991.

[66] George Nemhauser, Laurence Wolsey, and Marshall Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Prog.*, 14:265–294, 1978.

[67] Yiangos papanastasiou, Kostas Bimpikis, and Nicos Savva. Crowdsourcing exploration. *Manag. Sci.*, 64(4):1727–1746, 2018.

[68] Zinovi Rabinovich, Albert Xin Jiang, Manish Jain, and Haifeng Xu. Information disclosure as a means to security. In *Proc. 14th Conf. Auton. Agents and Multi-Agent Syst. (AAMAS)*, pages 645–653, 2015.

[69] Jérôme Renault, Eilon Solan, and Nicolas Vieille. Optimal Dynamic Information Provision. *Games Econ. Behav.*, 104:329–349, 2017.

[70] Aviad Rubinstein. Honest Signaling in Zero-Sum Games Is Hard, and Lying Is Even Harder. In *Proc. 44th Int. Colloq. Autom. Lang. Programming (ICALP)*, pages 77:1–77:13, 2017.

[71] Ester Samuel-Cahn. Comparison of Threshold Stop Rules and Maximum for Independent Nonnegative Random Variables. *The Annals of Probability*, 12(4):1213 – 1216, 1984.

[72] Haifeng Xu. On the tractability of public persuasion with no externalities. In *Proc. 30th Symp. Discret. Algorithms (SODA)*, pages 2708–2727, 2020.

[73] Haifeng Xu, Rupert Freeman, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. Signaling in Bayesian Stackelberg games. In *Proc. 15th Conf. Auton. Agents and Multi-Agent Syst. (AAMAS)*, pages 150–158, 2016.

[74] Haifeng Xu, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe. Exploring information asymmetry in two-stage security games. In *Proc. 29th Conf. Artif. Intell. (AAAI)*, pages 1057–1063, 2015.

[75] Robert Young. Euler's constant. *Math. Gaz*, pages 187–190, 1991.

[76] You Zu, Krishnamurthy Iyer, and Haifeng Xu. Learning to persuade on the fly: Robustness against ignorance. In *Proc. 22nd Conf. Econ. Comput. (EC)*, page 927–928, 2021.