Artificial Intelligence 334 (2024) 104171

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Delegated online search

Pirmin Braun [a], Niklas Hahn [b,*], Martin Hoefer [c,*], Conrad Schecker [a,*]

[a] Goethe University Frankfurt, Germany
[b] ISTA (Institute of Science and Technology Austria), Austria
[c] Department of Computer Science, RWTH Aachen University, Germany

ARTICLE INFO

Keywords:
Delegated search
Online algorithms
Approximation algorithms
Algorithmic game theory
Information design
Principal-agent problem

ABSTRACT

In a delegation problem, a *principal* $\mathcal{P}$ with commitment power tries to pick one out of n options. Each option is drawn independently from a known distribution. Instead of inspecting the options herself, $\mathcal{P}$ delegates the information acquisition to a rational and self-interested *agent* $\mathcal{A}$. After inspection, $\mathcal{A}$ proposes one of the options, and $\mathcal{P}$ can accept or reject.

Delegation is a classic setting in economic information design with many prominent applications, but the computational problems are only poorly understood. In this paper, we study a natural *online* variant of delegation, in which the agent searches through the options in an online fashion. For each option, he has to irrevocably decide if he wants to propose the current option or discard it, before seeing information on the next option(s). How can we design algorithms for $\mathcal{P}$ that approximate the utility of her best option in hindsight?

We show that in general $\mathcal{P}$ can obtain a $\Theta(1/n)$-approximation and extend this result to ratios of $\Theta(k/n)$ in case (1) $\mathcal{A}$ has a lookahead of k rounds, or (2) $\mathcal{A}$ can propose up to k different options. We provide fine-grained bounds independent of n based on three parameters. If the ratio of maximum and minimum utility for $\mathcal{A}$ is bounded by a factor α, we obtain an $\Omega(\log\log\alpha/\log\alpha)$-approximation algorithm, and we show that this is best possible. Additionally, if $\mathcal{P}$ cannot distinguish options with the same value for herself, we show that ratios polynomial in $1/\alpha$ cannot be avoided. If there are at most β different utility values for $\mathcal{A}$, we show a $\Theta(1/\beta)$-approximation. If the utilities of $\mathcal{P}$ and $\mathcal{A}$ for each option are related by a factor γ, we obtain an $\Omega(1/\log\gamma)$-approximation, where $O(\log\log\gamma/\log\gamma)$ is best possible.

1. Introduction

The study of delegation problems is a prominent area with numerous applications. There are two parties – a decision maker (called *principal*) $\mathcal{P}$ and an *agent* $\mathcal{A}$. n actions or *options* are available to $\mathcal{P}$. Each option has a utility for $\mathcal{P}$ and a (possibly different) utility for $\mathcal{A}$, which are drawn from a known distribution $\mathcal{D}$. Instead of inspecting options herself, $\mathcal{P}$ delegates the search for a good option to $\mathcal{A}$. $\mathcal{A}$ sees all realized utility values and sends a signal to $\mathcal{P}$. Based on this signal (and $\mathcal{D}$), $\mathcal{P}$ chooses an option. Both parties play this game in order to maximize their respective utility from the chosen option.

Many interesting applications can be captured within this framework. For example, consider a company that is trying to hire an expert in a critical area. Instead of searching the market, the company delegates the search to a head-hunting agency that searches the

[*] Corresponding authors.
E-mail addresses: pirminbraun16@gmail.com (P. Braun), niklas.hahn@ist.ac.at (N. Hahn), mhoefer@cs.rwth-aachen.de (M. Hoefer), schecker@em.uni-frankfurt.de (C. Schecker).

https://doi.org/10.1016/j.artint.2024.104171
Received 12 June 2023; Received in revised form 10 January 2024; Accepted 10 June 2024

Available online 20 June 2024

market for suitable candidates. Alternatively, consider an investor, who hires a financial consultant to seek out suitable investment opportunities. Clearly, principal and agent might not always have aligned preferences. While the investor might prefer investments with high interest rates, the financial consultant prefers selling the products for which he gets a provision.

Problems of this kind are recently arising as part of the deployment of AI in financial markets. Trading and investment decisions are increasingly made by automated agents. Delegation problems emerge prominently as part of the interplay of financial agents that search, analyze, and suggest potential investments for investors. More generally, AI development has been celebrated for the success in designing and building automated agents for board games such as chess or go. More recent challenges involve the development of automated strategies for advanced game-theoretic scenarios involving strategic communication (such as bluffing strategies in poker, or playing games like diplomacy). Strategic communication is also the distinguishing feature of principal-agent problems like delegation. Algorithmic understanding of this domain can be instrumental to push the frontier of AI development for games with strategic communication.

A key feature of applications such as searching for job candidates or financial investments is that the availability of options often changes over time, and the pair of agents needs to solve a stopping problem. For example, many lucrative financial investment opportunities arise only within short notice and expire quickly. Therefore, a financial service has to decide whether or not to recommend an investment without exactly knowing what future investment options might become available. Here $\mathcal{A}$ faces an online search problem, in which the n options are realized in a sequential fashion. After seeing the realization of option i, he has to decide whether to propose the option to $\mathcal{P}$ or discard it. In the elementary variant, if the option is proposed, $\mathcal{P}$ decides to accept or reject this option and the process ends. Otherwise, the process continues with option $i+1$. We also provide results for the more elaborate variant, in which $\mathcal{A}$ can make up to $k > 1$ proposals, and the process ends when $\mathcal{P}$ accepts a proposal, or as soon as she has rejected all k proposals.

In the study of delegation problems, $\mathcal{P}$ usually has commitment power, i.e., $\mathcal{P}$ specifies in advance her decision for each possible signal, taking into account the subsequent best response of $\mathcal{A}$. This is reasonable in many applications (e.g., an investor can initially restrict the investment options she is interested in, or the company fixes in advance the required qualifications for the new employee). Interestingly, although $\mathcal{P}$ commits and restricts herself in advance, this behavior is usually in her favor. The induced best response of $\mathcal{A}$ can lead to better utility for $\mathcal{P}$ than in any equilibrium, where both parties mutually best respond. Using a revelation-principle style argument, the communication between $\mathcal{P}$ and $\mathcal{A}$ can be reduced to $\mathcal{A}$ revealing the utilities of a single option and $\mathcal{P}$ deciding to accept or reject that option (for a discussion, see, e.g. [27]).

The combination of online search and delegation has been examined before, albeit from a purely technical angle. Kleinberg and Kleinberg [27] designed approximation algorithms for delegation, showing that $\mathcal{P}$ can obtain a constant-factor approximation to the expected utility of her best option in hindsight. Their algorithms heavily rely on techniques and tools developed in the domain of prophet inequalities. However, they are applied to an *offline* delegation problem. Instead, we consider the natural extension of [27] to online search. Interestingly, we exhibit a notable contrast – in online delegation a constant-factor approximation might be impossible to achieve. In fact, the approximation ratio can be as low as $O(1/n)$, and $\Omega(1/n)$ can always be achieved. Motivated by this sharp contrast, we provide a fine-grained analysis based on three natural problem parameters: (1) the discrepancy of utility for the agent, (2) the number of different utility values for the agent, and (3) the misalignment of agent and principal utilities.

1.1. Model

We study *online delegation* between principal $\mathcal{P}$ and agent $\mathcal{A}$ in (up to) n rounds. In every round i, an option is drawn independently from a known distribution $\mathcal{D}_i$ with finite support Ω_i of size s_i. We denote the options of $\mathcal{D}_i$ by $\Omega_i = \{\omega_{i1}, \dots, \omega_{i,s_i}\}$ and the random variable of the draw by O_i. For every $i \in [n]$ and $j \in [s_i]$, the option ω_{ij} has probability p_{ij} to be drawn from $\mathcal{D}_i$. If this option is proposed by $\mathcal{A}$ and chosen by $\mathcal{P}$, then $\mathcal{A}$ has utility $a_{ij} \ge 0$ and $\mathcal{P}$ utility $b_{ij} \ge 0$.

We assume that $\mathcal{P}$ has commitment power. Before the start of the game, she commits to an *action scheme* φ with a value $\varphi_{ij} \in [0, 1]$ for each option ω_{ij}. φ_{ij} is the probability that $\mathcal{P}$ accepts option ω_{ij} when it is proposed by $\mathcal{A}$ in round i. For a *deterministic* scheme with all $\varphi_{ij} \in \{0, 1\}$, we define the sets $E_i = \{\omega_{ij} \mid \varphi_{ij} = 1\}$ of *acceptable options* in each round i.

In contrast to $\mathcal{P}$, $\mathcal{A}$ gets to see the n random draws from the distributions in an online fashion. He decides in each round i whether he proposes the current option O_i to $\mathcal{P}$ or not. If he decides to propose it, then $\mathcal{P}$ sees the option and decides based on φ whether or not she accepts it. If $\mathcal{P}$ accepts, the utility values of the option are realized; if not, both players get utility 0. In either case, the game ends after $\mathcal{P}$ decides. Both players strive to maximize their expected utility.

Initially, both players know the distribution $\mathcal{D}_i$ for every round $i \in [n]$. The sequence of actions then is as follows: (1) $\mathcal{P}$ decides φ and communicates this to $\mathcal{A}$; (2) in each round i, $\mathcal{A}$ sees $O_i \sim \mathcal{D}_i$ and irrevocably decides to propose or discard O_i; (3) when $\mathcal{A}$ decides to propose some option $O_i = \omega_{ij}$, then $\mathcal{P}$ accepts it with probability φ_{ij}, and the game ends.

In Section 2.2 we also touch upon an (arguably more natural) extension in which $\mathcal{A}$ can sequentially propose up to k different options, for some given $k \in \{2, \dots, n\}$. In this case, the game ends after either $\mathcal{P}$ accepts a proposal, or $\mathcal{P}$ rejected all k proposals. In this case the action scheme φ becomes much more complex, since it needs to depend on the proposal history. Multiple sequential proposals allow $\mathcal{P}$ to monitor the behavior of $\mathcal{A}$ more closely and thereby reduce or even remove the delegation property (see Section 2.2 for details). As such, we focus on the conceptually simpler and technically more challenging variant with a single proposal throughout.

$\mathcal{A}$ knows the distributions and the action scheme φ of upcoming rounds. This determines his expected utility from proposed options. Hence, $\mathcal{A}$ essentially faces an online stopping problem that can be solved via backwards induction. We can assume without loss of generality that all decisions (not) to propose an option by $\mathcal{A}$ are deterministic: If the expected utility from the realization in

Table 1
An example instance.

	Round 1		Round 2	
option ω_{ij}	ω_{11}	ω_{12}	ω_{21}	ω_{22}
value-pair (a_{ij}, b_{ij})	(3,1)	(3,8)	(2,4)	(16,4)
probability p_{ij}	0.75	0.25	0.75	0.25

the current round is greater than the expected utility that can be obtained by optimal play in the remaining rounds, it is an optimal decision for $\mathcal{A}$ to propose the current option (otherwise not). To avoid technicalities, we assume that $\mathcal{A}$ breaks ties in favor of $\mathcal{P}$.

Our goal is to design action schemes φ with high expected utility for $\mathcal{P}$. In our analysis, we prove approximation ratios[1] w.r.t. the expected utility of $\mathcal{P}$ in the non-delegated (online) problem, i.e., when $\mathcal{P}$ searches through the n realized options herself. The latter is a classic stopping problem, for which the expected utilities of optimal online and offline search differ by at most a factor of 2 (due to the basic prophet inequality [28,29]).

We also analyze scenarios with *oblivious* and *semi-oblivious proposals*. In both these scenarios, $\mathcal{A}$ reveals only the utility value b_{ij} for $\mathcal{P}$ when proposing an option (but not his own value a_{ij}). In contrast, when $\mathcal{P}$ gets to see the utility values of both agents, we term this *conscious proposals*. The difference between the oblivious variants lies in the prior knowledge of $\mathcal{P}$:

Conscious: In the conscious scenario, $\mathcal{P}$ is fully aware of the distributions, including all possible utility values b_{ij}, a_{ij} for both parties. When an option ω_{ij} gets proposed in some round i, $\mathcal{P}$ sees both utility values b_{ij} and a_{ij}.

Semi-Oblivious: In the semi-oblivious scenario, $\mathcal{P}$ is fully aware of the distributions, including all potential utility values b_{ij} and a_{ij} for both parties. However, when an option ω_{ij} gets proposed in some round i, $\mathcal{P}$ sees only her utility value b_{ij}.

Oblivious: In the oblivious scenario, $\mathcal{P}$ initially observes the probabilities of all options with her utility values b_{ij}. However, all utility values a_{ij} of $\mathcal{A}$ remain unknown to $\mathcal{P}$ throughout. As such, when an option ω_{ij} gets proposed in some round i, $\mathcal{P}$ sees only her utility value b_{ij}.

In the scenarios with restricted discrepancy (in Section 3.1), we assume $\mathcal{P}$ is aware of the bound $\alpha = \max_{i,j} a_{ij} / \min_{i,j} a_{ij}$.

Example 1. We discuss a simple example to illustrate the definitions. We consider deterministic strategies by $\mathcal{P}$ and conscious proposals. There are two rounds with the options distributed according to Table 1.

For the benchmark, we assume that $\mathcal{P}$ can see and choose the options herself. The best option is ω_{12}. If this is not realized in round 1, the option realized in round 2 is the best choice. Note that this optimal choice for $\mathcal{P}$ can be executed even in an online scenario, where she first sees round 1 and gets to see round 2 only after deciding about round 1. The expected utility of this best (online) choice for $\mathcal{P}$ is 5.

Now in the delegated scenario, suppose $\mathcal{P}$ accepts option ω_{22}. Then $\mathcal{A}$ would always wait for round 2 and hope for a realization of ω_{22}, even if ω_{21} would not be accepted by $\mathcal{P}$. Hence, accepting ω_{22} leads to an expected utility for $\mathcal{P}$ of at most 4. In contrast, the optimal decision scheme for $\mathcal{P}$ is to accept only ω_{12} and ω_{21} with an expected utility of 4.25. For the (semi-)oblivious scenario, $\mathcal{P}$ cannot distinguish the options in round 2, and her expected utility is at most 4.

Clearly, $\mathcal{P}$ has to strike a careful balance between (1) accepting a sufficient number of high-profit options to obtain a high expected utility overall and (2) rejecting options to motivate $\mathcal{A}$ to propose better options for $\mathcal{P}$ in earlier rounds. ♦

1.2. Contribution and outline

In Section 2 we show that the worst-case approximation ratio for online delegation is $O(1/n)$ and a ratio of $\Omega(1/n)$ can always be achieved. Intuitively, $\mathcal{A}$ waits too long and forgoes many profitable options for $\mathcal{P}$. $\mathcal{P}$ can only force $\mathcal{A}$ to propose earlier if she refuses to accept later options – this, however, also hurts the utility of $\mathcal{P}$. The instances require a ratio of maximum and minimum utility values for $\mathcal{A}$ that is in the order of $n^{\Theta(n)}$.

We further show that the bounds extend to more general variants in which (1) $\mathcal{A}$ has a lookahead of k rounds, or (2) $\mathcal{A}$ can propose up to k options, resulting in tight approximation ratios of $\Theta(k/n)$. Note that all impossibility results in this paper apply already for IID instances, while all algorithmic results apply for general instances.

Intuitively, in the extended variants the problem becomes "less online" (with lookahead) or "less delegated" (with k proposals). In the latter variant, when there are up to n proposals, $\mathcal{P}$ can force the agent to propose every single option (by threatening to reject all remaining subsequent options). Essentially, this reduces the problem to non-delegated online search. Hence, the challenges of online delegation arise in the single-proposal variant without lookahead in their most concise form. This provides further motivation to concentrate on this variant in the subsequent sections.

In Section 3, we examine the effect of the utility values of for $\mathcal{A}$ using two different parameters: (1) the discrepancy of utility for $\mathcal{A}$ using the ratio α of maximum and minimum utility values, and (2) the number β of different possible utility values for $\mathcal{A}$.

[1] Approximation ratios of this type have also been termed *delegation gap* in [8].

Table 2
Asymptotic bounds on the approximation ratio in terms of different instance parameters.

Proposals	General utilities	α-bounded utilities for $\mathcal{A}$	β different values for $\mathcal{A}$	γ-bounded $\mathcal{P}/\mathcal{A}$-misalignment
Conscious	$\Theta\left(\frac{1}{n}\right)$	$\Theta\left(\frac{\log\log\alpha}{\log\alpha}\right)$	$\Theta\left(\frac{1}{\beta}\right)$	$\Omega\left(\frac{1}{\log\gamma}\right)$ $O\left(\frac{\log\log\gamma}{\log\gamma}\right)$
Semi-Oblivious	$\Theta\left(\frac{1}{n}\right)$	$\Omega\left(\frac{1}{\sqrt{\alpha}\log\alpha}\right)$ $O\left(\frac{1}{\sqrt{\alpha}}\right)$	$\Theta\left(\frac{1}{n}\right)$	$\Omega\left(\frac{1}{\gamma}\right)$ $O\left(\frac{1}{\sqrt{\gamma}}\right)$
Oblivious	$\Theta\left(\frac{1}{n}\right)$	$\Theta\left(\frac{1}{\alpha}\right)$	$\Theta\left(\frac{1}{n}\right)$	$\Theta\left(\frac{1}{\gamma}\right)$

First, we consider α-bounded utilities in Section 3.1 and obtain an $\Omega(\log\log\alpha/\log\alpha)$-approximation of the optimal (online) search for $\mathcal{P}$. This result is tight. The algorithm limits the acceptable options of $\mathcal{P}$, partitions them into different bins, and then restricts $\mathcal{A}$'s search space to the best possible bin for $\mathcal{P}$. The challenge is to design a profitable set of options inside a bin that should be accepted by $\mathcal{P}$ without giving $\mathcal{A}$ an incentive to forgo proposing many of them. Using our algorithm $\mathcal{P}$ obtains a good approximation even if the differences in utility of $\mathcal{A}$ are polynomial in n.

Additionally, we consider the more challenging *semi-oblivious* and *oblivious* scenarios in which $\mathcal{P}$ does not get to see the agent's utility of the proposed option. In the (fully) oblivious case, $\mathcal{P}$ is even a priori unaware of the utility values for $\mathcal{A}$ for all options (and thus remains so throughout). Our algorithm for this scenario achieves an $\Omega(1/\alpha)$-approximation. This is contrasted with a set of instances for which no action scheme can extract more than an $O(1/\alpha)$–approximation. In the *semi-oblivious scenario,* $\mathcal{P}$ has a priori knowledge of the prior, but still does not see the agent's utility for proposed options. For this setting, our algorithm achieves an $\Omega(1/(\sqrt{\alpha}\log\alpha))$-approximation, and in general $O(1/\sqrt{\alpha})$ is best-possible. The results highlight the effect of hiding $\mathcal{A}$'s utilities from $\mathcal{P}$ (in the proposal, or in the proposal and the prior) – the achievable approximation ratios increase from logarithmic to polynomial ratios in α.

Second, for β many different utility values for $\mathcal{A}$ we extend the techniques developed in the previous subsection to achieve a $\Theta(1/\beta)$-approximation for conscious proposals. In the semi-oblivious scenario, the ratio w.r.t. β can become unbounded. This extends to the oblivious scenario.

In Section 4, we consider the misalignment of agent and principal utilities via a parameter $\gamma \geq 1$, which is the smallest value such that all utilities of $\mathcal{P}$ and $\mathcal{A}$ are related by a factor in $[1/\gamma, \gamma]$. Limited misalignment also leads to improved approximation results for $\mathcal{P}$. We show an $\Omega(1/\log\gamma)$-approximation of the optimal (online) search for $\mathcal{P}$. Moreover, every algorithm must have a ratio in $O(\log\log\gamma/\log\gamma)$. For both (semi-)oblivious scenarios, we obtain an $\Omega(1/\gamma)$-approximation. In the oblivious scenario, this is tight. In the semi-oblivious scenario, every algorithm must have a ratio in $O(1/\sqrt{\gamma})$.

In Table 2, we summarize our lower and upper bounds on the approximation ratio in terms of the different instance parameters.

Overall, our results reveal that the impact of delegation in online search can be severe – in comparison to the undelegated online search problem, the achieved value can deteriorate by a $\Theta(n)$-factor. We highlight that such a drastic deterioration relies on a number of pathological worst-case properties, whose absence can significantly improve the performance. Lookahead or multiple proposals by $\mathcal{A}$ lead to a substantially smaller deterioration. Intuitively, lookahead reduces the degree to which the scenario is "online". Multiple proposals can be used to closely monitor and penalize the behavior of $\mathcal{A}$ and, thus, reduce the degree to which the scenario is "delegated". Hence, these adjustments lead to improvements by changing key properties of the underlying search problem.

Bounded utility values for $\mathcal{A}$ or bounded misalignment of the utilities of $\mathcal{A}$ and $\mathcal{P}$ can significantly improve the performance of delegated online search. The resulting ratios only depend logarithmically on the involved parameters. Interestingly, we also observe a significant dependence on the information that $\mathcal{P}$ receives as part of the proposal. Overall, our results tightly characterize conditions under which the impact of delegation on the performance is mild. Note that we do not (directly) address the problem of approximating an *optimal scheme* for delegated online search, which represents a fascinating direction for future work.

1.3. Related work

Holmstrom [25,26] initiated the study of delegation as a bilevel optimization between an uninformed principal and a privately informed agent. The principal delegates the decision to the agent who himself has an interest in the choice of decision. Since then, there have been numerous works on various aspects of delegation. For example, [30,1] studied the impact of (mis)alignment of the agent's and the principal's interests on the optimal delegation sets. Armstrong and Vickers [2] studied the delegation problem over discrete sets of random cardinality with elements drawn from some distribution. They identify sufficient conditions for the search problem to have an optimal solution.

A similar model to ours was considered in computer science by Kleinberg and Kleinberg [27], where the option set searched by the agent consists of n IID draws from a known distribution. They show constant-factor approximations of the optimal expected principal utility when performing the search herself rather than delegating it to the agent. For their analysis, they rely on tools from online stopping theory. The key difference between their work and our paper is that – albeit using tools from online optimization – they study an *offline* problem while we focus on an *online* version.

Bechtel and Dughmi [7] recently extended this line of research by combining delegation with stochastic probing. Here a subset of elements can be observed by the agent (subject to some constraints), and several options can be chosen (subject to a different set of constraints). Similarly, Bechtel et al. [8] study connections between delegation and a generalized Pandora's Box problem.

A related but different area is contract theory, which considers principal-agent settings with uncertainty and monetary transfers. An early formalization was introduced by Grossmann and Hart [22]. Computational aspects of this problem are recently starting to attract interest, see, e.g., Dütting et al. [18,19], as well as Babaioff et al. [3,4] for earlier work (on a slightly different model).

The study of persuasion, another model of strategic communication, has gained a lot of traction at the intersection between economics and computation in recent years. Here, the informed party (the "sender") is the one with commitment power, trying to influence the behavior of the uninformed agent (the "receiver"). Persuasion has proven to be a popular topic in AI. Castiglioni et al. [14] studied Bayesian posted price auctions where buyers arrive sequentially and receive signals from the revenue maximizing seller. Moreover, signaling may be used in other settings, e.g., persuading voters [9,12], or for reducing social cost in congestion games with uncertain delays [11,21]. Closer to our paper is the study of persuasion in the context of stopping problems [24,23]. These works study persuasion problems in a prophet inequality [23] as well as in a secretary setting [24].

Other notable algorithmic results on persuasion problems concern optimal policies, hardness, and approximation algorithms in the general case [17] as well as in different variations, e.g., with multiple receivers [5,6,16,31,32], with limited communication complexity [15,20], or relations to online learning [10,13,33]

2. Impossibility

2.1. A tight bound

As a first simple observation, note that $\mathcal{P}$ can always achieve an n-approximation with a deterministic action scheme, even in scenarios with oblivious proposals. $\mathcal{P}$ accepts exactly all options in a single round i^* with optimal expected utility, i.e., $E_{i^*} = \{\omega_{i^*,j} \mid j \in [s_{i^*}]\}$ for $i^* = \arg\max_{i\in[n]} \mathbb{E}[b_{ij}]$, and $E_j = \emptyset$ otherwise. This motivates $\mathcal{A}$ to always propose the option from round i^*, and $\mathcal{P}$ gets expected utility $\mathbb{E}[b_{i^*,j}]$. By a union bound, the optimal utility from searching through all options herself is upper bounded by $\mathbb{E}\left[\sum_i b_{ij}\right] \le n \cdot \mathbb{E}[b_{i^*,j}]$.

Proposition 1. *For online delegation there is a deterministic action scheme φ such that $\mathcal{P}$ obtains at least a $1/n$-approximation of the expected utility for optimal (online) search.*

We show a matching impossibility result, even in the IID setting with $\mathcal{D}_i = \mathcal{D}$ for all rounds $i \in [n]$, and when $\mathcal{P}$ gets to see the full utility pair of any proposed option. There are instances in which $\mathcal{P}$ suffers a deterioration in the order of $\Theta(n)$ over the expected utility achieved by searching through the options herself.

For the proof, consider the following class of instances. The distribution $\mathcal{D}$ can be cast as an independent composition, i.e., we independently draw the utility values for $\mathcal{P}$ and $\mathcal{A}$. For $\mathcal{P}$ there are two possibilities, either utility 1 with probability $1/n$, or utility 0 with probability $1 - 1/n$. For $\mathcal{A}$, there are n possibilities with agent utility of $n^{4\ell}$, for $\ell = 1, \dots, n$, where each one has probability $1/n$. In combination, we can view $\mathcal{D}$ as a distribution over $j = 1, \dots, 2n$ options. Options ω_j for $j = 1, \dots, n$ have probability $1/n^2$ and utilities $(b_j, a_j) = (1, n^{4j})$, for $j = n+1, \dots, 2n$ they have probability $1/n - 1/n^2$ and utilities $(b_j, a_j) = (0, n^{4(j-n)})$.

Theorem 1. *There is a class of instances of online delegation in the IID setting, in which every action scheme φ obtains at most an $O(1/n)$-approximation of the expected utility for optimal (online) search.*

Proof. For simplicity, we first show the result for schemes φ with $\varphi_{ij} = 0$ for all rounds $i \in [n]$ and all $j = n+1, \dots, 2n$. In the end of the proof we discuss why this can be assumed for an optimal scheme.

Since all options $j \in [n]$ have the same utility for $\mathcal{P}$, she wants to accept one of them as soon as it appears. If she searches through the options herself, the probability that there is an option of value 1 is $1 - (1 - 1/n)^n \ge 1 - 1/e$. Her expected utility is a constant. In contrast, when delegating the search to $\mathcal{A}$, the drastic utility increase motivates him to wait for the latest round in which a better option is still acceptable by $\mathcal{P}$. As a result, $\mathcal{A}$ waits too long, and removing acceptable options in later rounds does not remedy this problem for $\mathcal{P}$.

More formally, interpret an optimal scheme φ as an $n \times n$ matrix, for rounds $i \in [n]$ and options $j \in [n]$. We outline some adjustments that preserve the optimality of matrix φ.

Consider the set S of all entries with $\varphi_{ij} \le 1/n$. For each $(i, j) \in S$, the probability that option j is realized in round i is $1/n^2$. When it gets proposed by $\mathcal{A}$, then it is accepted by $\mathcal{P}$ with probability at most $1/n$. By a union bound, the utility obtained from all these options is at most $|S|/n^2 \cdot 1/n \le 1/n$.

Suppose we change the scheme by decreasing φ_{ij} to 0 for each $(i, j) \in S$. Then each entry in φ is either 0 or at least $1/n$. If $\mathcal{A}$ makes the same proposals as before, the change decreases the utility of $\mathcal{P}$ by at most $1/n$. Then again, in the new scheme $\mathcal{A}$ can have an incentive to propose other options in earlier rounds. Since all options with $\varphi_{ij} \ne 0$ have utility 1 for $\mathcal{P}$, this only leads to an increase of utility for $\mathcal{P}$. Moreover, in round 1 we increase all acceptance probabilities to $\varphi_{1j} = 1$ for $j \in [n]$. Then, upon arrival of such an option ω_j in round 1, the change can incentivize $\mathcal{A}$ to propose this option – which is clearly optimal for $\mathcal{P}$, since this is an optimal option for her. Since the change is in round 1, it introduces no incentive to wait for $\mathcal{A}$. As such, it can only increase the utility for $\mathcal{P}$. Now consider any entry $\varphi_{ij} \ge 1/n$. We observe two properties:

Table 3
Adjustments from the proof of Theorem 1 for an example scheme φ for $n = 4$. Left: Entries $\varphi_{ij} < 1/n$ (bold) get set to 0. The expected utility for $\mathcal{P}$ decreases by at most $1/n$. Middle: Italic entries show options that never get proposed due to options with bold entries (cf. property 1.). Italic entries can be set to 0. Right: Bold entries have been raised according to property 2. A Manhattan path of $2n - 1$ entries evolves.

Rnd.	ω_1	ω_2	ω_3	ω_4		Rnd.	ω_1	ω_2	ω_3	ω_4		Rnd.	ω_1	ω_2	ω_3	ω_4
1	0.3	**0.2**	0.6	0.9		1	*0.3*	*0*	*0.6*	0.9		1	0	0	0	**1**
2	0.4	0.3	0.8	1	→	2	*0.4*	*0.3*	0.8	**1**	→	2	0	0	**1**	1
3	**0.2**	0.4	0.6	**0.2**		3	*0*	*0.4*	0.6	0		3	0	0	**0.7**	0
4	0.5	**0.1**	0.7	0		4	0.5	0	**0.7**	0		4	**0.7**	**0.7**	0.7	0

1. Suppose $\varphi_{i'j'} \geq 1/n$ for $i' < i$ and $j' < j$. Then $\mathcal{P}$ accepts realization $\omega_{j'}$ in round i' with positive probability, but she will also accept the better (for $\mathcal{A}$) realization ω_j in a later round i with positive probability. $\mathcal{A}$ will not propose $\omega_{j'}$ in round i' but wait for round i, since the expected utility in the later round i is at least $n^{4j} \cdot 1/n^2 \cdot \varphi_{ij} \geq n^{4j-3} > n^{4(j-1)} \geq n^{4j'} \cdot \varphi_{i'j'}$, the utility in round i'. As such, w.l.o.g. we set $\varphi_{i'j'} = 0$ for all $i' < i$ and $j' < j$.
2. Suppose $\varphi_{i'j} < \varphi_{ij}$ for $i' < i$. By property 1., all realizations $\omega_{j'}$ with $j' < j$ are not accepted in rounds $1, \ldots, i-1$. Hence, setting $\varphi_{i'j} = \varphi_{ij}$ does not change the incentives for $\mathcal{A}$ w.r.t. other options, and thus only (weakly) increases the expected utility of $\mathcal{P}$. By the same arguments, we set $\varphi_{ij'} = \max\{\varphi_{ij'}, \varphi_{ij}\}$ for all inferior options $j' < j$ in the same round i.

We apply the adjustments implied by the two properties repeatedly, starting for the entries φ_{in} in the n-th column for option ω_n, then in column $n-1$, etc. By 1., every positive entry $\varphi_{ij} \geq 1/n$ leads to entries of 0 in all "dominated" entries $\varphi_{i'j'}$ with $i' < i$ and $j' < j$. As a consequence, the remaining positive entries form a Manhattan path in the matrix φ. The path starts at φ_{1n}, ends at φ_{n1}, and for each $\varphi_{ij} \geq 1/n$ it continues either at $\varphi_{i+1,j} \geq 1/n$ or $\varphi_{i,j-1} \geq 1/n$. See Table 3 for an example.

We can upper bound the expected utility of $\mathcal{P}$ by assuming that all $2n-1$ entries on the Manhattan path are 1 (i.e., φ is deterministic) and $\mathcal{A}$ proposes an acceptable option whenever possible. The probability that this happens is at most $(2n-1)/n^2 = O(1/n)$ by a union bound. This is an upper bound on the expected utility of $\mathcal{P}$ and proves the theorem for schemes with $\varphi_{ij} = 0$ for all $i \in [n]$ and $j \geq n+1$.

Finally, suppose $\varphi_{ij} > 0$ for some $j \geq n+1$. Clearly, option ω_j adds no value to the expected utility of $\mathcal{P}$. Moreover, the fact that ω_j has positive probability to be accepted in round i can only motivate $\mathcal{A}$ to refrain from proposing inferior options in earlier rounds. As such, setting $\varphi_{ij} = 0$ only (weakly) increases the utility of $\mathcal{P}$. □

2.2. *Extensions*

We discuss two generalizations for which a slight adaptation of our lower bound of $\Theta(1/n)$ established in Section 2 is sufficient. First, we assume $\mathcal{A}$ has a lookahead. Second, we allow several proposals to be made by $\mathcal{A}$.

2.2.1. *Agent with lookahead*

Consider online delegation when $\mathcal{A}$ has a lookahead of k rounds. In round i, $\mathcal{A}$ gets to see all realized options of rounds $i, i+1, \ldots, \min\{n, i+k\}$. For simplicity, our benchmark here is the expected value of $\mathcal{P}$ for optimal (non-delegated) *offline* search (i.e., online search with lookahead $n-1$). Note that the expected value for optimal online search is at least 1/2 of this [28,29]. Hence, asymptotically all benchmarks of expected utility for optimal offline or online search, with or without lookahead, are the same.

Proposition 2. *For online delegation with lookahead k there is an action scheme φ such that $\mathcal{P}$ obtains an $\Omega(k/n)$-approximation of the expected utility for optimal (offline) search.*

Partition the n rounds into $\lceil n/(k+1) \rceil$ parts with at most $k+1$ consecutive rounds each. Suppose we apply (non-delegated) offline search on each part individually. The expected value of offline search on the best of the $O(n/k)$ parts yields an $\Omega(k/n)$-approximation of the expected value of offline search on all n rounds.

To obtain an $\Omega(k/n)$-approximation for the online delegated version, apply online search with $\mathcal{A}$ and lookahead of k to the best part of at most $k+1$ consecutive rounds. Due to the lookahead, this results in offline delegated search. In terms of utility for $\mathcal{P}$, offline delegated search using prophet-inequality techniques [27] approximates optimal offline search by at least a factor of 1/2. Hence, applying the offline delegation algorithm of [27] on the best set of $k+1$ consecutive rounds yields an $\Omega(k/n)$-approximation.

Let us show that this guarantee is asymptotically optimal. The argument largely follows the proof of Theorem 1. The class of instances is the same. We only explain which parts of the proof must be adapted.

Corollary 1. *There is a class of instances of online delegation with lookahead k in the IID setting, in which every action scheme φ obtains at most an $O(k/n)$-approximation of the expected utility for optimal (offline) search.*

Proof. Using similar observations as in the proof of Theorem 1, we can again (a) assume w.l.o.g. that $\varphi_{ij} = 0$ for all $j = n+1, \ldots, 2n$, and (b) assume that $\varphi_{ij} = 0$ or $\varphi_{ij} \geq 1/n$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, 2n$, which deteriorates the expected utility for $\mathcal{P}$ by at most $O(1/n)$.

Consider the two properties in the proof of Theorem 1. For property (1), we extend the idea to entries $\varphi_{i'j'} \geq 1/n$ with $j' < j$ and $i' + k < i$. In particular, $\mathcal{A}$ will decide to wait and not propose option $\omega_{j'}$ in round i' if there is a round $i > i' + k$ with a better option ω_j being acceptable (with probability at least $\varphi_{ij} \geq 1/n$). As such, we drop $\varphi_{i'j'}$ to 0 whenever such a constellation arises. Then, whenever an entry remains $\varphi_{i'j'} \geq 1/n$, this means that all entries for better options $j > j'$ in rounds $i = i' + k + 1, \dots, n$ must be $\varphi_{ij} = 0$.

Now for a given option $\omega_{j'}$, consider round $i_{j'} = \arg\min\{i \mid \varphi_{ij'} \geq 1/n\}$. Then, for all better options with $j > j'$, property (1) requires that $\varphi_{ij} = 0$ for all $i \geq i_{j'} + k + 1$. As such, for each option ω_j, there can be at most k positive entries "beyond the Manhattan path", i.e., k rounds in which ω_j remains acceptable (with prob. at least $1/n$) after the first round when any lower-valued $\omega_{j'}$ becomes acceptable (with prob. at least $1/n$).

Property (2) applies similarly as before. As such, we obtain a Manhattan path with $2n - 1$ entries, and in addition there can be up to nk entries with $\varphi_{ij} \geq 1/n$, i.e., a total of at most $(k+2)n - 1$ entries. We again upper bound the expected utility of $\mathcal{P}$ by assuming that all these entries are 1 and $\mathcal{A}$ proposes an acceptable option whenever possible. The probability that this happens is at most $((k+2)n-1)/n^2 = O(k/n)$ by a union bound, and this implies the upper bound on the expected utility of $\mathcal{P}$. □

2.2.2. Agent with k proposals

Now consider the case when $\mathcal{A}$ can propose up to k options in k different rounds. In this case, the definition of an action scheme becomes more complex – rather than a single matrix, φ turns into a decision tree. For each round i, consider the *history* $H_i = (h_1, \dots, h_{i-1})$. For every round $j = 1, \dots, i-1$, the entry h_j indicates whether or not there was a proposal by $\mathcal{A}$ in round j, and if so, which option was proposed. Now given a round i and a history H_i, an action scheme yields a value $\varphi_{ij}(H_i) \in \{0, 1\}$ indicating whether or not $\mathcal{P}$ accepts option ω_j when being proposed in round i after history H_i. As before, $\mathcal{P}$ commits to an action scheme anticipating the induced rational behavior of $\mathcal{A}$. A simple backward induction shows that there is always an optimal proposal strategy for $\mathcal{A}$ which is deterministic. For simplicity, we also restrict attention to deterministic action schemes for $\mathcal{P}$.

Proposition 3. *For online delegation with k proposals there is a deterministic action scheme φ such that $\mathcal{P}$ obtains an $\Omega(k/n)$-approximation of the expected utility for optimal (offline) search.*

The scheme is related to the approach in the previous section. Select the best interval $\ell, \dots, \ell + k - 1$ of k consecutive rounds that maximize the expected value of offline search for $\mathcal{P}$ over these rounds. We observed in the previous section that optimal offline search in these k rounds yields an $\Omega(k/n)$-approximation of optimal offline search over n rounds. We design an action scheme that incentivizes $\mathcal{A}$ to propose exactly the k options in rounds $\ell, \dots, \ell + k - 1$, thereby reducing the scenario to (non-delegated) online search for $\mathcal{P}$ over these rounds. Since the performance of online and offline search are related by a factor of 2, asymptotically we achieve the same performance as offline search over the k rounds.

We set $\varphi_{ij}(H_i) = 0$ for rounds $i < \ell$ and all j and H_i, as well as for rounds $i > \ell + k - 1$ and all j and H_i. For each round $\ell \leq i \leq \ell + k - 1$, we set $\varphi_{ij}(H_i) = 0$ for all options j whenever the history reveals that in at least one of the rounds $\ell, \dots, i-1$ there was no proposal from $\mathcal{A}$. Otherwise, if H_i reveals that there was a proposal in each of these rounds, we set $\varphi_{ij}(H_i)$ as in an optimal online (non-delegated) search over rounds $\ell, \dots, \ell + k - 1$.

In this action scheme, $\mathcal{P}$ immediately terminates the search whenever she did not receive a proposal from $\mathcal{A}$ in one of the k rounds, leaving both agents with a utility of 0. This creates an incentive for $\mathcal{A}$ to submit a proposal in each of the k rounds, since this is the only possibility to obtain a positive utility value.

Let us show that the approximation guarantee is asymptotically optimal. The argument uses and extends the result of Theorem 1. The class of instances is the same.

Theorem 2. *There is a class of instances of online delegation with k proposals in the IID setting, in which every deterministic action scheme φ obtains at most an $O(k/n)$-approximation of the expected utility for optimal (offline) search.*

Proof. We analyze the process by splitting the evolution of the process into at most k non-overlapping *phases*. Let i_ℓ be the (random) round in which $\mathcal{A}$ makes the ℓ-th proposal, for $\ell = 1, \dots, k$. For completeness, we set $i_0 = 0$. Phase ℓ is the set of rounds $\{i_{\ell-1} + 1, \dots, i_\ell\}$. $\mathcal{P}$ can accept an option in at most one of the phases. Thus, by linearity of expectation, the expected utility of $\mathcal{P}$ is upper bounded by the sum of expected utilities that $\mathcal{P}$ obtains in each phase. In the rest of the proof, we will show that in each phase, the expected utility for $\mathcal{P}$ is at most $O(1/n)$. Hence, the total expected utility of $\mathcal{P}$ is $O(k/n)$, which proves the approximation guarantee.

Towards this end, consider a single phase ℓ. We condition on the *full history* of the process before phase ℓ, i.e., we fix all options drawn as well as decisions of $\mathcal{P}$ and $\mathcal{A}$ that have led to the $(\ell - 1)$-th proposal in round $i_{\ell-1}$. We denote this full history by H^f. During phase ℓ (conditioned on H^f), we want to interpret the process as a single-proposal scenario analyzed in Theorem 1. In particular, by fixing the history and the starting round of phase ℓ, the histories H_i within phase ℓ are also fixed. As such, during phase ℓ, the scheme φ can be seen as an action scheme for a single-proposal scenario with $n - i_{\ell-1}$ rounds.

Now let us define an auxiliary single-proposal instance with n rounds. In this instance, we assume $\mathcal{P}$ sets $\varphi'_{ij} = 0$ for all options $j = 1, \dots, 2n$ in the first $i = 1, \dots, i_{\ell-1}$ rounds and then in rounds $i = i_{\ell-1} + 1, \dots, n$ applies $\varphi'_{ij} = \varphi_{ij}(H_i)$ (where H_i is composed of H^f and no proposal in rounds $i_{\ell-1} + 1, \dots, i - 1$). Then $\mathcal{P}$ behaves in the auxiliary instance exactly as in phase ℓ of the k-proposal instance. In contrast, $\mathcal{A}$ does not necessarily show the same behavior. In the auxiliary instance, $\mathcal{A}$ gets utility 0 if the proposal is rejected. In phase ℓ of the k-proposal instance, however, proposing an option that gets rejected can be profitable for $\mathcal{A}$. After rejection, phase $\ell + 1$ is reached and better expected utility for $\mathcal{A}$ might be achievable in upcoming rounds (since the scheme could result in more favorable behavior of $\mathcal{P}$ when the ℓ-th reject decision happens in round i).

In the auxiliary instance, we model this property by a *reject bonus* for $\mathcal{A}$ – whenever a proposal is rejected in any round $i \geq i_{\ell-1} + 1$, then ($\mathcal{P}$ receives no utility and) $\mathcal{A}$ receives the conditional expected utility from optimal play in the remaining rounds of the k-proposal instance, conditioned on φ and history H_{i+1} composed of H_i and the rejected proposal in round i. It is straightforward to see that in the auxiliary instance with reject bonus, the interaction between $\mathcal{P}$ and $\mathcal{A}$ exactly proceeds as in phase ℓ of the k-proposal instance.

Consider the auxiliary single-proposal instance with reject bonus for any given phase ℓ. We prove that the expected utility for $\mathcal{P}$ does not decrease when the following *adjustments* are made: (1) the reject bonus is reduced to 0, and (2) we set $\varphi'_{ij} = 0$ for all rounds i and options $j \geq n + 1$ (i.e., the ones with $b_j = 0$). We prove the statement by induction over the rounds.

Clearly w.l.o.g. there are no proposals in rounds $i \leq i_{\ell-1}$. We can assume that (1) and (2) hold for all these rounds. Now consider round $i = i_{\ell-1} + 1$. When rejecting an option, or when accepting an option of value 0, the utility for $\mathcal{P}$ is 0. For these options, adjustments (1) and (2) in round i change the utility for $\mathcal{A}$ to 0, as well. When facing such an option in round i, the adjustments incentivize $\mathcal{A}$ to wait for potentially better subsequent options. It weakly increases the expected utility of $\mathcal{P}$.

Towards an induction, suppose the statement is true after the adjustments (1) and (2) in all rounds $i_{\ell-1} + 1 \leq i' \leq i$. Now consider round $i + 1$. First, condition on the event that in both instances (with and without adjustments on round $i + 1$), we reach round $i + 1$. As argued above, the adjustments in round $i + 1$ imply that $\mathcal{A}$ has less incentive to propose an option of value 0 for $\mathcal{P}$ in round $i + 1$ and more incentive to wait for subsequent rounds. Hence, the utility for $\mathcal{P}$ (conditioned on reaching round $i + 1$) does not decrease.

Note, however, that the probability of reaching round $i + 1$ also changes by the adjustments. For every $i_{\ell-1} < i' < i + 1$, removing the reject bonus and reducing the set of acceptable options in round $i + 1$ lead to a reduction in expected utility for $\mathcal{A}$ from the rounds *after* round i'. This increases the probability that $\mathcal{A}$ will propose an option in some round i' *before* $i + 1$. It decreases the probability of reaching round $i + 1$. Nevertheless, this is again good news for $\mathcal{P}$: Since by hypothesis $\mathcal{P}$ accepts only options of utility 1 and there is no reject bonus in all rounds $i' \leq i$, any proposal in these rounds is accepted and results in optimal utility for $\mathcal{P}$. Overall, $\mathcal{P}$ only profits from the adjustments (1) and (2) in round $i + 1$. By induction, this holds when the adjustments are made in all rounds.

After the adjustments, the auxiliary instance is a standard single-proposal instance studied in the context of Theorem 1. This shows that the expected utility obtained by $\mathcal{P}$ is in $O(1/n)$.

As a consequence, the conditional expected utility for $\mathcal{P}$ in phase ℓ (conditioned on each H^f) is upper bounded by $O(1/n)$. Hence, the overall expected utility from phase ℓ is at most $O(1/n)$. The expected utility from k phases is upper bounded by $O(k/n)$. This proves the theorem. □

3. Agent utilities

In this section, we examine the effects of two parameters on the approximation ratio: (1) the discrepancy of utility values of $\mathcal{A}$, or (2) the number of different utility values of $\mathcal{A}$. The lower bound instance in Theorem 1 uses n different utility values for $\mathcal{A}$, ranging from 1 to $n^{\Theta(n)}$. Is such a drastic discrepancy necessary to show a substantial lower bound? Can we obtain better approximation results for instances with a smaller ratio of the maximum and minimum utility values for $\mathcal{A}$? Similarly, can a bound on the number of utility values for $\mathcal{A}$ help to improve the approximation for $\mathcal{P}$?

3.1. Discrepancy of agent utilities

In this subsection, we study the discrepancy of agent utility values, i.e., the ratio between the maximum and the minimum utility value for $\mathcal{A}$. In the next subsection, we begin by studying the setting of conscious proposals.

3.1.1. Conscious proposals

Let us first assume that $a_{ij} > 0$ for all options (see Remark 1 below how to extend our results to the case when $a_{ij} = 0$ is allowed). Let $\alpha = \max\{a_{ij} \mid i \in [n], j \in [s_i]\} / \min\{a_{ij} \mid i \in [n], j \in [s_i]\}$. W.l.o.g. we scale all utility values to $a_{ij} \in [1, \alpha]$, where both boundaries α and 1 are attained by at least one option. Then $\mathcal{A}$ has *α-bounded utilities*.

We use Algorithm 1 to compute a good action scheme with respect to parameter α. Intuitively, we partition the best options for $\mathcal{P}$ that add up a total probability mass of roughly 1/2 into $O(\log\alpha / \log\log\alpha)$ many bins. Each bin is constructed in a way such that $\mathcal{A}$ is incentivized to propose the first option he encounters from that particular bin, assuming that $\mathcal{P}$ only accepts options from that bin. The algorithm determines the best bin for $\mathcal{P}$ and deterministically restricts the acceptable options to the ones from this bin.

Let us discuss the algorithm in more detail. As a first step, the algorithm uses a procedure RestrictOptions($\mathcal{D}_1, \ldots, \mathcal{D}_n, m$) (Algorithm 2) with parameter $m = 1/2$ as subroutine. The procedure considers all options in descending order of principal utility until a total probability mass m is reached. Starting out with $\hat{Q} = \emptyset$, options are added to $\hat{Q} = \{(i, j) \mid b_{ij} \geq b_{i',j'} \ \forall (i', j') \notin \hat{Q}\}$ as long as $\sum_{(i,j)\in\hat{Q}} p_{ij} < m$. The first option (i^*, j^*) that would reach or surpass the combined mass of m (i.e., such that $\sum_{(i,j)\in\hat{Q}\cup\{(i^*,j^*)\}} p_{ij} \geq m$) is considered separately. The procedure RestrictOptions then returns either $Q = \hat{Q}$ or $Q = \{(i^*, j^*)\}$, whichever set provides a better expected utility for $\mathcal{P}$. As a consequence, $\sum_{(i,j)\in Q} p_{ij} \cdot b_{ij} \geq m/2 \cdot \mathbb{E}_{\omega_{ij}\sim\mathcal{D}_i}[\max_{i\in[n]} b_{ij}]$.

Lemma 1 summarizes the claim.

Lemma 1. *The subroutine RestrictOptions($\mathcal{D}_1, \ldots, \mathcal{D}_n, m$) with distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and a parameter $0 < m \leq 1$ as input identifies Q, the best set of options for $\mathcal{P}$, such that*

Algorithm 1: $\Omega(\log\log\alpha/\log\alpha)$-approximation.

```
Input: $n$ distributions $\mathcal{D}_1,\dots,\mathcal{D}_n$
Output: Action Scheme $\varphi$

1  Let $Q = \text{RestrictOptions}(\mathcal{D}_1,\dots,\mathcal{D}_n, 1/2)$.
2  if $Q$ spans only a single round then Set $B_1 = Q$.
3  else
4      Construct $c = \lceil \log_2 \alpha \rceil$ classes $C_1,\dots,C_c$ such that $C_k = \{(i,j) \in Q \mid a_{ij} \in [2^{k-1}, 2^k)\}$ for all $k = 1,\dots,c-1$ and
         $C_c = \{(i,j) \in Q \mid a_{ij} \in [2^{c-1}, 2^c]\}$.
5      Set $b = 1$, and $s = c$. Open bin 1 and set $B_1 = \emptyset$.
6      for $k = c$ down to 1 do
7          if $2^{k-1} < 2^s \cdot \sum_{(i,j)\in B_b \cup C_k} p_{ij}$ then
8              set $b = b+1$ and $s = k$.          // $\sum_{(i,j)\in Q} p_{ij} < 1/2$, so no open bin stays empty
9              Open the new bin $b$ and set $B_b = \emptyset$
10         Add class $C_k$ to bin $B_b = B_b \cup C_k$.
11 $b^* = \arg\max_{b=1,\dots} \sum_{(i,j)\in B_b} p_{ij} b_{ij}$, the bin with highest utility for $\mathcal{P}$.
12 Set $\varphi_{ij} = 1$ for all $(i,j) \in B_{b^*}$ and $\varphi_{ij} = 0$ otherwise.
13 return $\varphi$
```

Algorithm 2: RestrictOptions.

```
Input: $n$ distributions $\mathcal{D}_1,\dots,\mathcal{D}_n$, value $m$ restricting the mass
Output: Set $Q$ of good options for $\mathcal{P}$
1  Set $\hat{Q} = \emptyset$, $p = p^* = 0$, $U = \bigcup_{i=1}^n \bigcup_{j=1}^{s_i} \{(i,j)\}$.
2  while $p < m$ do
3      $U^* = \emptyset$, $p^* = 0$
4      for $k = 1,\dots,n$ do
5          Let $U_k^* = \{(k,j) \in U \mid b_{kj} \ge b_{i'j'}$ for all $(i',j') \in U\}$ be the options in round $k$ from the set of all remaining options with the best utility
             for $\mathcal{P}$
6          Set $p_k = \sum_{(i,j)\in U_k^*} p_{ij}$
7          if $p^* + p_k < m$ then add $U^* = U^* \cup U_k^*$, update $p^* = p^* + p_k$
8          else
9              if $p_k > p^*$ then set $U^* = U_k^*$
10             break for-loop
11     Set $p^* = \sum_{(i,j)\in U^*} p_{ij}$ and $b^* = b_{ij}$ for $(i,j) \in U^*$
                                                  /* All options in $U^*$ have the same value for $\mathcal{P}$. */
12     if $p + p^* > m$ then set $B = U^*$; else add $\hat{Q} = \hat{Q} \cup U^*$
13     Remove $U = U \setminus U^*$, update $p = p + p^*$.
14 Set $b_{\hat{Q}} = \sum_{(i,j)\in\hat{Q}} p_{ij} b_{ij}$ and $b_B = p^* b^*$
15 if $b_{\hat{Q}} < b_B$ then set $Q = B$ else set $Q = \hat{Q}$
16 return $Q$
```

$$\sum_{(i,j)\in Q} p_{ij} \cdot b_{ij} \ge m/2 \cdot \mathbb{E}_{\omega_{ij}\sim\mathcal{D}_i}[\max_{i\in[n]} b_{ij}]$$

and either (1) the combined mass $\sum_{(i,j)\in Q} p_{ij} < m$ or (2) all options in Q arrive in the same round.

Proof. RestrictOptions first restricts the possible set of options to $\hat{Q} \cup B$ consisting of the best options for $\mathcal{P}$, with a union probability mass of at least m in the while-loop in lines 2-13. Inside this while-loop, the options with the highest utility for $\mathcal{P}$ are identified using the for-loop in lines 4-10. This loop ensures that no more than a combined mass of m is considered for a set of options from different rounds with the same (currently highest) utility for $\mathcal{P}$. Should such a set with a higher combined mass than m exist, the if/else-statement in lines 7-10 picks the better part of this set while ensuring that either the combined mass of the set is at most m or only options from a single round are considered.

Hence, by line 13 it holds that

$$1/m \cdot \sum_{(i,j)\in\hat{Q}\cup B} p_{ij} b_{ij} \ge \mathbb{E}_{\omega_{ij}\sim\mathcal{D}_i}[\max_{i\in[n]} b_{ij}] \ .$$

In line 14, the utility for $\mathcal{P}$ from the sets $\hat{Q}$ and B is calculated, in line 15, the better one for $\mathcal{P}$ is chosen. This means that at most another factor of 2 is lost. In total, the set Q returned by RestrictOptions guarantees

$$2/m \cdot \sum_{(i,j)\in Q} p_{ij} b_{ij} \geq \mathbb{E}_{\omega_{ij}\sim\mathcal{D}_i}[\max_{i\in[n]} b_{ij}] \ . \quad \square$$

If the set Q returned by RestrictOptions only spans a single round i, the agent will always be incentivized to propose an acceptable option in round i. For this scenario, the algorithm only creates a single bin B_1. Otherwise, it continues with the second and third step described in the following.

In the second step of the algorithm, the options identified by RestrictOptions are classified by their utility for $\mathcal{A}$. The algorithm divides Q into $c = \lceil \log_2 \alpha \rceil$ classes depending on the agent utility such that the lowest and highest agent utilities in any given class differ by at most a factor of 2. More precisely, classes $C_1, \dots, C_c$ are constructed such that $C_k = \{(i,j) \in Q \mid a_{ij} \in [2^{k-1}, 2^k)\}$ for $k = 1, \dots, c-1$ and $C_c = \{(i,j) \in Q \mid a_{ij} \in [2^{c-1}, 2^c]\}$.

For the third step, subsequent classes (by their agent utility value) are combined into bins such that (1) the bins are as big as possible and (2) $\mathcal{A}$ optimizes his own expected utility by proposing the first option he encounters from any bin – assuming that only options from this bin are allowed. Classes are either added to a bin completely or not at all. Let s be the index of the class with the highest agent utilities currently considered, i.e., the first class to be added to the current bin B_b. We consider the classes by decreasing agent utility values, i.e., with indices $k = s, s-1, \dots$. While $2^{k-1} \geq 2^s \cdot \sum_{(i,j)\in B_b \cup C_k} p_{ij}$, a rational $\mathcal{A}$ will always propose the first option available from the current bin if that is the only allowed bin as it has a higher utility than what $\mathcal{A}$ can expect from later rounds. Hence, the class currently under consideration can be added to the current bin.

Finally, having constructed all bins, the algorithm simply chooses the best one for $\mathcal{P}$.

Lemma 2 shows that our algorithm achieves an approximation ratio which is linear in the number of bins opened. The subsequent Lemma 3 bounds the number of bins opened, showing that it is in the order of $O(\log\alpha/\log\log\alpha)$. Together, the lemmas prove Theorem 3, our main result of the section.

Lemma 2. *Let ℓ be the number of bins opened by the algorithm. Then the scheme computed by the algorithm obtains at least an $1/(8\ell)$-approximation of the expected utility of the best option for $\mathcal{P}$ in hindsight.*

Proof. We know that Q satisfies $4 \cdot \sum_{(i,j)\in Q} p_{ij} \cdot b_{ij} \geq \mathbb{E}_{\omega_{ij}\sim\mathcal{D}_i}[\max_{i\in[n]} b_{ij}] = \mathrm{OPT}$ by Lemma 1.

Now consider the construction of the bins. Suppose we split Q into ℓ bins $B_1, B_2, \dots, B_\ell$. We pick the best one B_{b^*} from the ℓ bins $B_1, \dots, B_\ell$, so

$$\sum_{(i,j)\in B_{b^*}} p_{ij} b_{ij} \geq \frac{1}{\ell} \sum_{(i,j)\in Q} p_{ij} b_{ij} \geq \frac{1}{4\ell} \cdot \mathrm{OPT} \ .$$

The action scheme restricts attention to B_{b^*} and accepts each proposed option ω_{ij} from the bin with probability 1. Let $k^- = \min\{k \mid C_k \subseteq B_{b^*}\}$ be the class of smallest index in B_{b^*}, and k^+ the one with largest index, respectively. Now suppose the agent learns in round i that an option ω_{ij} with $(i,j) \in B_{b^*}$ arrives in this round. We claim that $\mathcal{A}$ will then decide to propose this option. This is obvious if all options in B_{b^*} are only realized in round i. Otherwise, the agent might want to wait for an option in a later round. If $\mathcal{A}$ proposes, then his utility is a_{ij}. Otherwise, if he waits for another option from B_{b^*} in a later round, then a union bound shows that the expected utility is at most

$$\sum_{\substack{(i',j')\in B_{b^*}\\ i'>i}} p_{i'j'} \cdot a_{i'j'} \leq \sum_{\substack{(i',j')\in B_{b^*}\\ i'>i}} p_{i'j'} \cdot 2^{k^+} < 2^{k^+} \cdot \sum_{(i',j')\in B_{b^*}} p_{i'j'} \leq 2^{k^- -1} \leq a_{ij} \ ,$$

where the second-to-last inequality is a consequence from the construction of the bin. Hence, the first option from the bin that is realized also gets proposed by $\mathcal{A}$ and accepted by $\mathcal{P}$.

Now for each option $(i,j) \in B_{b^*}$, the probability that this option is proposed and accepted is the combination of two independent events: (1) no other option from B_{b^*} was realized in any of the rounds $i' < i$, (2) option ω_{ij} is realized in round i. The probability for event (2) is p_{ij}. For event (1), we define $m_i = \sum_{(i,j)\in B_{b^*}} p_{ij}$. With probability $\prod_{i'<i}(1 - m_{i'})$, no option from B_{b^*} is realized in rounds $i' < i$. Note that $\sum_{i=1}^n m_i \leq 1/2$. The term $\prod_{i=1}^n (1-m_i)$ is minimized for $m_1 = 1/2$ and $m_{i'} = 0$ for $1 < i' < i$. Thus $\prod_{i=1}^n (1-m_i) \geq 1/2$, i.e., the probability of event (1) is at least $1/2$.

Overall, by linearity of expectation, the expected utility of $\mathcal{P}$ when using φ is at least

$$\sum_{(i,j)\in B_{b^*}} \frac{1}{2} \cdot p_{ij} \cdot b_{ij} \geq \frac{1}{2\ell} \cdot \sum_{(i,j)\in Q} p_{ij} \cdot b_{ij} \geq \frac{1}{8\ell} \cdot \mathrm{OPT} . \quad \square$$

Lemma 3. *Let ℓ be the number of bins opened by the algorithm. It holds that $\ell = O(\log\alpha/\log\log\alpha)$.*

Proof. Consider a bin B and its mass $p_B = \sum_{(i,j)\in B} p_{ij}$. We want to argue that at most $O(c/\log c)$ bins are opened. To do so, we first condition on having ℓ open bins and strive to lower bound the number of classes in these ℓ bins.

Consider a bin B starting at C_s. The algorithm adds classes to B until $2^{k-1} < 2^s p_B$. Thus, $s - k + 1 > \log_2(1/p_B)$, i.e., the number of classes in B_i is lower bounded by $\log_2(1/p_B)$.

Now consider two bins B_i and B_{i+1} and condition on $q = p_{B_i} + p_{B_{i+1}}$. Together the bins contain at least $\log_2(1/p_{B_i}) + \log_2(1/(q - p_{B_i}))$ classes. Taking the derivative for p_{B_i}, we see that this lower bound is minimized when $p_{B_i} = q/2 = p_{B_{i+1}}$. Applying this balancing step repeatedly, the lower bound on the number of classes in all bins is minimized when $p_{B_i} = p_{B_j}$ for all bins B_i, B_j. Thus, when opening ℓ bins, we obtain the smallest lower bound on the number of classes in these bins by setting $p_{B_i} = 1/\ell \cdot \sum_{(i,j)\in Q} p_{ij} < 1/(2\ell)$ for all bins B_i. Conversely, when opening ℓ bins, we need to have at least $\ell \log_2(2\ell)$ classes in these bins.

Now, since we need to put c classes into the bins, we need to ensure that for the number ℓ of open bins we have $\ell(\log_2 \ell + 1) \le c$, since otherwise the ℓ bins would require more than c classes in total. This implies that $c = \Omega(\ell \log_2 \ell)$ and, hence, $\ell = O(c/\log c) = O(\log\alpha/\log\log\alpha)$. □

Using Lemmas 2 and 3, we can prove the following theorem:

Theorem 3. *If the agent has α-bounded utilities, there is a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(\log\log\alpha/\log\alpha)$-approximation of the expected utility for optimal (online) search.*

Observe that the approximation ratio of this algorithm is tight in general. Consider the instances in Theorem 1 with $\alpha = n^{\Theta(n)}$. The theorem shows that every scheme can obtain at most a ratio of $O(1/n) = O(\log\log\alpha/\log\alpha)$.

Remark 1. If there are options with utility 0 for $\mathcal{A}$, the maximum ratio between the lowest and highest utility for $\mathcal{A}$ becomes unbounded. Still, if the maximum ratio between the lowest and highest *non-zero* utility for $\mathcal{A}$ is bounded by α, an $\Omega(\log\log\alpha/\log\alpha)$-approximation can be achieved with a slight modification of Algorithm 1. Suppose there are any options with $a_{ij} = 0$ in Q, then construct another bin B_0 which consists of all options with 0 utility for $\mathcal{A}$ in the set Q. If B_0 is the bin that is chosen as the best bin in the algorithm, the agent will not receive any utility and, due to tie-breaking in favor of $\mathcal{P}$, can be assumed to execute an online search for $\mathcal{P}$. Using standard prophet inequality results, this yields a 1/2-approximation for $\mathcal{P}$ within this bin. If bin B_0 is not the best bin, the analysis from the theorem can be applied.

Remark 1 shows that we can indeed allow options with $a_{ij} = 0$. If all other options with strictly positive utility are α-bounded, we obtain the same approximation factor (asymptotically).

Corollary 2. *If the agent has α-bounded utilities for all options with strictly positive utility, there is a deterministic action scheme such that the principal obtains an $\Omega(\log\log\alpha/\log\alpha)$-approximation of the expected utility for optimal (online) search.*

3.1.2. Oblivious proposals

In the previous section, we considered algorithms for $\mathcal{P}$ when she learns the utility pair for the proposed option. In this section, we show that (fully) oblivious proposals can be a substantial drawback for $\mathcal{P}$. Obviously, the lower bound in Theorem 1 remains intact even for oblivious proposals, when $\mathcal{P}$ does not learn the utility value of the proposed option for $\mathcal{A}$. For oblivious proposals and α-bounded agent utilities, we can significantly strengthen the lower bound. In contrast to the logarithmic approximation guarantee above, we provide a linear lower bound in α for oblivious proposals.

Theorem 4. *There is a class of instances of online delegation with α-bounded utilities for the agent and oblivious proposals, in which every action scheme φ obtains at most an $O(1/\alpha)$-approximation of the expected utility for optimal (online) search.*

Proof. Consider the following class of instances. In $\mathcal{D}_i$, there are two options with the following probabilities and utilities: ω_{i1} with $p_{i1} = 1 - 1/n$ and $(b_{i1}, a_{i1}) = (0, 1)$, as well as ω_{i2} with $p_{i2} = 1/n$ and $(b_{i2}, a_{i2}) = (1, x_i)$, where $x_i \in \{1, \alpha\}$ and $\alpha \in [1, n]$. In the first rounds $i = 1, \dots, i^* - 1$ we have $x_i = 1$, then $x_i = \alpha$ for rounds $i = i^*, \dots, n$. The expected utility when $\mathcal{P}$ performs (undelegated) online search is $1 - (1 - 1/n)^n \ge 1 - 1/e$.

$\mathcal{P}$ wants that $\mathcal{A}$ proposes any profitable option ω_{i2} as soon as possible. As in the proof of Theorem 1, we can assume that all $\varphi_{i1} = 0$ in an optimal scheme – this option has no value for $\mathcal{P}$ and can only raise the incentive to wait for $\mathcal{A}$.

Due to oblivious proposals, $\mathcal{P}$ has to choose φ without being aware of the value of i^*. For our impossibility result, we adjust i^* to the scheme φ chosen by $\mathcal{P}$: Set $i^* \in \{1, \dots, n\}$ to the largest number such that $\sum_{i=i^*}^n \varphi_{i2} \ge e \cdot n/\alpha$, or $i^* = 1$ if no such number exists.

First, suppose that $i^* = 1$. Then, even if we force $\mathcal{A}$ to propose *every* option ω_{i2} as soon as it arises, a union bound shows that the expected utility of $\mathcal{P}$ is upper bounded by $\sum_{i=1}^n \frac{1}{n} \cdot \varphi_{i2} \le \frac{e}{\alpha} + \frac{1}{n}$. Hence, $\mathcal{P}$ obtains only an $O(1/\alpha)$-approximation, for any $\alpha \in [1, n]$.

Now suppose that $i^* > 1$. Consider an optimal scheme φ for $\mathcal{P}$. If ω_{i2} arises in round i, $\mathcal{A}$ decides if it is more profitable to propose i or wait for a later round. Indeed, we show that $\mathcal{A}$ never proposes ω_{i2} in any round $i < i^*$. Consider the expected utility from proposing the first option ω_{k2} arising in rounds $k = i^*, \dots, n$. This is

$$\alpha \cdot \left(\sum_{k=i^*}^{n} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{k-i^*} \varphi_{k2} \right) = \alpha \cdot \frac{1}{n} \cdot \sum_{k=i^*}^{n} \left(1 - \frac{1}{n}\right)^{k-i^*} \varphi_{k2} > \frac{\alpha}{n} \cdot \frac{1}{e} \cdot \sum_{k=i^*}^{n} \varphi_{k2} \ge \frac{\alpha}{en} \cdot \frac{en}{\alpha} = 1 \ge \varphi_{i2} \ ,$$

i.e., strictly larger than the expected utility φ_{i2} from proposing ω_{i2} in round $i < i^*$. Hence, $\mathcal{A}$ only proposes in rounds $k = i^*, \dots, n$. Even if $\mathcal{A}$ would be able to propose *every* option ω_{k2} in rounds $k = i^*, \dots, n$, a union bound implies that the expected utility of $\mathcal{P}$ from these rounds is upper bounded by $\sum_{k=i^*}^n \frac{1}{n} \cdot \varphi_{k2} \le \frac{e}{\alpha} + \frac{1}{n}$. For any $\alpha \in [1, n]$, $\mathcal{P}$ obtains an $O(1/\alpha)$-approximation. □

Algorithm 3: $\Omega(1/\alpha)$-approximation for oblivious proposals.

```
Input: n distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$
Output: Action Scheme $\varphi$
Let $Q$ = RestrictOptions($\mathcal{D}_1, \dots, \mathcal{D}_n, 1/(2\alpha)$).
Set $\varphi_{ij} = 1$ for all $(i, j) \in Q$.
return $\varphi$
```

This class of instances is actually a worst-case. In the following, we show that an $\Omega(1/\alpha)$-approximation can always be achieved for this setting. To this end, we provide Algorithm 3. It uses the subroutine RestrictOptions (Algorithm 2) with parameter $m = 1/(2\alpha)$ to determine a set of acceptable options Q. Intuitively, $\mathcal{P}$ uses this subroutine to find a small subset of her best options such that $\mathcal{A}$ is incentivized to propose the first realized option from this subset. This is similar to what $\mathcal{P}$ did in the conscious setting. The key difference here is that she cannot use bins (that depend on $\mathcal{A}$'s utilities, of which $\mathcal{P}$ is oblivious) to make further selections that prevent $\mathcal{A}$ from waiting. However, as the total probability mass of acceptable options is only a $1/(2\alpha)$-fraction and $\mathcal{A}$ has α-bounded utilities, $\mathcal{A}$ is incentivized to propose an option from Q as soon as it is realized. Thus, $\mathcal{P}$ can indeed accept all options from Q. We formalize this claim in Theorem 5.

Theorem 5. *If the agent has α-bounded utilities and makes oblivious proposals, there is a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(1/\alpha)$-approximation of the expected utility for optimal (online) search.*

Proof. Consider Algorithm 3. The proof follows along the lines of Lemma 2. By Lemma 1, we have $4\alpha \cdot \sum_{(i,j)\in Q} p_{ij} b_{ij} \geq \text{OPT}$, the expected value of the best option in hindsight.

The action scheme accepts each proposed option ω_{ij} from the set Q with probability 1. Note that Q satisfies either that $\sum_{(i,j)\in Q} p_{ij} < 1/(2\alpha)$ or all options in Q arrive in the same round i.

In the latter case, $\mathcal{A}$ will propose any option ω_{ij} with $(i,j) \in Q$ he encounters in round i. In a later round $i' > i$, $\mathcal{P}$ will not accept any option.

Hence, let us consider the former case that Q satisfies $\sum_{(i,j)\in Q} p_{ij} < 1/(2\alpha)$. Suppose the agent learns in round i that an option ω_{ij} with $(i,j) \in Q$ arrives. We claim that $\mathcal{A}$ will propose this option. If $\mathcal{A}$ proposes, then the expected utility is a_{ij}. Otherwise, if he waits for another option from Q in a later round, then a union bound shows that the expected utility is at most

$$\sum_{(i',j')\in Q,\ i'>i} p_{i'j'} \cdot a_{i'j'} \leq \sum_{(i',j')\in Q,\ i'>i} p_{i'j'} \cdot \alpha \cdot a_{ij} \leq a_{ij} \ ,$$

where the first inequality is due to α-bounded utilities, and the second inequality follows since $\sum_{(i,j)\in Q} p_{ij} \leq 1/(2\alpha)$ by construction. Hence, the first option from Q that is realized also gets proposed by $\mathcal{A}$ and accepted by $\mathcal{P}$.

Now, for each option $(i,j) \in Q$, the probability that this option is proposed and accepted is the combination of two independent events: (1) no other option from Q was realized in any of the rounds $i' < i$, (2) option ω_{ij} is realized in round i. The probability of event (2) is p_{ij}. For the probability of event (1), we define $m_i = \sum_{(i,j)\in Q} p_{ij}$. With probability $\prod_{i'<i}(1 - m_i)$ no option from Q is realized in rounds $i' < i$. Note that $\sum_{i=1}^n m_i \leq 1/(2\alpha)$. The term $\prod_{i=1}^n (1 - m_i)$ is minimized for $m_1 = 1/(2\alpha)$ and $m_{i'} = 0$ for $1 < i' < i$. Thus $\prod_{i=1}^n (1 - m_i) \geq 1 - 1/(2\alpha)$, i.e., the probability of event (1) is at least $1 - 1/(2\alpha) \geq 1/2$.

By linearity of expectation, the expected utility of $\mathcal{P}$ when using φ based on Q is at least

$$\sum_{(i,j)\in Q} \frac{1}{2} \cdot p_{ij} \cdot b_{ij} \geq \frac{1}{8\alpha} \cdot \text{OPT} \ . \quad \square$$

In contrast to Corollary 2, the result of Theorem 5 does *not* generalize to the case when $\mathcal{A}$ has options with utility 0, and α is the ratio of maximum and minimum non-zero utility. Even in the semi-oblivious scenario (discussed in the next section), all algorithms must have a ratio in $O(1/n)$, even when all utilities for $\mathcal{A}$ are $a_{ij} \in \{0, 1\}$.

3.1.3. Semi-oblivious proposals

In this section, we analyze semi-oblivious proposals. $\mathcal{P}$ has full a priori information about the prior, but she does not learn the utility value of $\mathcal{A}$ upon a proposal. The information about the prior can indeed help to improve the approximation ratio from $\Theta(1/\alpha)$ to $\Omega(1/(\sqrt{\alpha}\log\alpha))$, but not to a logarithmic bound as shown for conscious proposals in Theorem 3. In particular, we start by showing the following limit on the approximation ratio.

Theorem 6. *There is a class of instances of online delegation with IID options, α-bounded utilities for the agent, and semi-oblivious proposals, in which every action scheme φ obtains at most an $O(1/\sqrt{\alpha})$-approximation of the expected utility for optimal (online) search.*

Proof. The lower bound applies in an IID instance with three different options. One option is bad for both $\mathcal{P}$ and $\mathcal{A}$, but has a very high probability of $1 - 1/n$. The other two options provide the same (good) utility for $\mathcal{P}$, one of which is good and the other one bad for $\mathcal{A}$. The combined probability of both options is $1/n$. Since $\mathcal{P}$ cannot distinguish between the two good options, in each round she has to decide to either accept both or reject both. While $\mathcal{P}$ would like to accept any of the good options, $\mathcal{A}$ has an incentive to wait and propose only the option that is good for both. Overall, this approach ensures that every achievable approximation ratio for $\mathcal{P}$ must be in $O(1/\sqrt{\alpha})$.

More formally, consider the following class of IID instances with $\mathcal{D}_i = \mathcal{D}_j = \mathcal{D}$. In $\mathcal{D}$, there are three options with the following probabilities and utilities: ω_1 with $p_1 = 1 - 1/n$ and $(b_1, a_1) = (0, 1)$, ω_2 with $p_2 = 1/n - 1/(n\sqrt{\alpha})$ and $(b_2, a_2) = (1, 2)$, and ω_3 with $p_3 = 1/(n\sqrt{\alpha})$ and $(b_3, a_3) = (1, \alpha)$, for any $\alpha \in [2, n^2]$.

Note that $\mathcal{P}$ cannot distinguish between the latter options when they are proposed. Thus, in each round i, $\mathcal{P}$ accepts option ω_1 with probability φ_{i1} and options ω_2 and ω_3 with φ_{i2}. As in the proof of Theorem 1, we can assume that all $\varphi_{i1} = 0$ in an optimal scheme – this option yields no value for $\mathcal{P}$ and could only raise the incentive to wait for $\mathcal{A}$.

Consider any optimal scheme φ for $\mathcal{P}$. To obtain an upper bound on the utility of $\mathcal{P}$, we assume that $\mathcal{A}$ always proposes ω_3 whenever it is realized.[2] For ω_2, he evaluates whether or not it is profitable to wait for a later round. Suppose $\mathcal{A}$ proposes ω_2 in round i. A necessary condition for this is that the expected utility from proposing ω_3 in subsequent rounds is smaller, i.e.,

$$2\varphi_{i2} \ge \alpha \cdot \sum_{k=i+1}^{n} \frac{\varphi_{k2}}{n\sqrt{\alpha}} \left(1 - \frac{1}{n\sqrt{\alpha}}\right)^{k-i-1} = \frac{\sqrt{\alpha}}{n} \cdot \sum_{k=i+1}^{n} \left(1 - \frac{1}{n\sqrt{\alpha}}\right)^{k-i-1} \varphi_{k2} \ . \tag{1}$$

If this condition is fulfilled, we set $\Delta_i = 1$. Otherwise, we set $\Delta_i = 0$. Then, using a union bound, the utility of $\mathcal{P}$ from φ can be upper bounded by

$$\sum_{i=1}^{n} \varphi_{i2} \left(\frac{1}{n\sqrt{\alpha}} + \Delta_i \left(\frac{1}{n} - \frac{1}{n\sqrt{\alpha}} \right) \right) \ . \tag{2}$$

Consider the first round i_s in which $\Delta_{i_s} = 1$.

Combining (1) with the fact that $\varphi_{i_s,2} \le 1$, this means that

$$2 \ge \frac{\sqrt{\alpha}}{n} \cdot \sum_{k=i_s+1}^{n} \left(1 - \frac{1}{n\sqrt{\alpha}}\right)^{k-i_s-1} \varphi_{k2} \ ,$$

which implies

$$\sum_{k=i_s+1}^{n} \varphi_{k2} < \frac{2n}{\sqrt{\alpha}} \cdot \left(1 - \frac{1}{n\sqrt{\alpha}}\right)^{-n} \le \frac{2n}{\sqrt{\alpha}} \cdot \left(1 - \frac{1}{\sqrt{\alpha}}\right)^{-1} < \frac{7n}{\sqrt{\alpha}} \ .$$

For the second inequality, we used Bernoulli's inequality $(1+x)^n \ge 1 + xn$ for all $x \ge -1$ and all $n \in \mathbb{N}$. Finally, $(1 - 1/\sqrt{\alpha})^{-1} < 7/2$ for $\alpha \in [2, n^2]$ as its maximum is attained at $\alpha = 2$ and $(1 - 1/\sqrt{2})^{-1} < (1 - 5/7)^{-1} = 7/2$.

Using (2) and our assumption that $\alpha \in [2, n^2]$, the utility of $\mathcal{P}$ is upper bounded by

$$\frac{1}{n} \sum_{i=1}^{n} \varphi_{i2} \left(\frac{1}{\sqrt{\alpha}} + \Delta_i \left(1 - \frac{1}{\sqrt{\alpha}}\right)\right) \le \frac{1}{n} \left(\frac{i_s - 1}{\sqrt{\alpha}} + 1 + \frac{7n}{\sqrt{\alpha}} \right) = O\left(\frac{1}{\sqrt{\alpha}}\right) \ . \quad \square$$

For the setting of semi-oblivious proposals, we design a more elaborate algorithm. It uses one of two subroutines, depending on the expected utility for $\mathcal{A}$ (for pseudocode see Algorithm 6). The resulting action scheme provides an $\Omega(1/(\sqrt{\alpha}\log\alpha))$-approximation for $\mathcal{P}$.

Consider all options with the same utility for $\mathcal{P}$ in a single round. This set of options has *low agent expectation* if the conditional expected utility for $\mathcal{A}$ in this set of options is less than $\sqrt{\alpha}$. Otherwise, it has *high agent expectation*. For the first subroutine, we concentrate on all options with low agent expectation. Hence, this subroutine is called AlgoLow (Algorithm 4).

Other options are considered to receive a utility of 0 for $\mathcal{P}$ and, thus, are excluded from consideration. The scheme φ_L achieves an $\Omega(1/\sqrt{\alpha})$-approximation in the instance $\mathcal{D}^{(L)}$, where only options with low agent expectation generate value for $\mathcal{P}$. Similarly, for options with high agent expectation we describe procedure AlgoHigh (Algorithm 5).

The scheme φ_H achieves an $\Omega(1/(\sqrt{\alpha}\log_2\alpha))$-approximation in the instance $\mathcal{D}^{(H)}$, where only options with high agent expectation generate value for $\mathcal{P}$. In the end, we choose the better scheme for $\mathcal{P}$, thereby forfeiting at most another factor 2 of her optimal expected utility. Overall, our Algorithm obtains a ratio of $\Omega(1/(\sqrt{\alpha}\log\alpha))$.

Theorem 7. *If the agent has α-bounded utilities and makes semi-oblivious proposals, there is a deterministic action scheme such that $\mathcal{P}$ obtains at least an $\Omega(1/(\sqrt{\alpha}\log\alpha))$-approximation of the expected utility for optimal (online) search.*

[2] Note that due to the differences in acceptance probabilities φ_{i2}, he might actually have an incentive to wait for a later round, in which the probability that $\mathcal{P}$ accepts is higher.

Algorithm 4: 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 $\mathcal{P}$ has an expectation for
        $\mathcal{A}$ of less than $\sqrt{\alpha}$
Output: Action Scheme $\varphi$
Set $Q = RestrictOptions(\mathcal{D}_1, \dots, \mathcal{D}_n, 1/2)$.
Set $\ell = 1$, $b_1 = p_1 = 0$, $C_1 = \emptyset$
for $k = 1, \dots, n$ do
    Set $p^* = \sum_{(k,j) \in Q} p_{kj}$
    if $p_\ell + p^* > 1/\sqrt{\alpha}$ then set $\ell = \ell + 1, C_\ell = \{(k, j) \in Q\}, p_\ell = p^*$;
      else add $C_\ell = C_\ell \cup \{(k, j) \in Q\}$
Set $b'_\ell = \sum_{(i,j) \in C_{\ell'}} p_{ij} b_{ij}$ for all $1 \le \ell' \le \ell$.
Choose $\ell^*$ such that $b_{\ell^*} \ge b_{\ell'}$ for all $1 \le \ell' \le \ell$.
Set $\varphi_{ij} = 1$ for all $(i, j) \in C_{\ell^*}$.
return $\varphi$
```

Algorithm 5: AlgoHigh.

```
Input: $n$ distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$, where in every distribution individually, the set of options with the same value for $\mathcal{P}$ has an expectation for
        $\mathcal{A}$ of at least $\sqrt{\alpha}$
Output: Action Scheme $\varphi$
Set $Q = RestrictOptions(\mathcal{D}_1, \dots, \mathcal{D}_n, 1/4)$.
for $k = 0, \dots, \lfloor \log_2 \sqrt{\alpha} \rfloor - 1$ do
    Set $C_k = \{(i, j) \in Q \mid \frac{\sum_{(i,j') \in Q, b_{ij} = b_{ij'}} p_{ij} a_{ij}}{\sum_{(i,j') \in Q, b_{ij} = b_{ij'}} p_{ij}} \in \left[\sqrt{\alpha} \cdot 2^k, \sqrt{\alpha} \cdot 2^{k+1}\right)\}$
    Set $b_k = \sum_{(i,j) \in C_k} p_{ij} b_{ij}$.
Set $C_{\lfloor \log_2 \sqrt{\alpha} \rfloor} = \{(i, j) \in Q \mid \frac{\sum_{(i,j') \in Q, b_{ij} = b_{ij'}} p_{ij} a_{ij}}{\sum_{(i,j') \in Q, b_{ij} = b_{ij'}} p_{ij}} \in \left[\sqrt{\alpha} \cdot 2^{\lfloor \log_2 \sqrt{\alpha} \rfloor}, \alpha\right]\}$
Set $b_k = \sum_{(i,j) \in C_k} p_{ij} b_{ij}$.
Choose $k$ such that $b_k \ge b_{k'}$ for all $k' = 0, \dots, \lfloor \log_2 \sqrt{\alpha} \rfloor$. Set $\varphi_{ij} = 1$ for all $(i, j) \in C_k$.
return $\varphi$
```

Let us give a brief intuition for AlgoLow. The algorithm leverages the low expectation for $\mathcal{A}$ by restricting the number of rounds from which options are accepted. More precisely, it partitions the set Q computed by RestrictOptions with parameter $1/2$ into $O(\sqrt{\alpha})$ many classes according to contiguous time intervals of rounds. The action scheme φ then accepts only options from the best class for $\mathcal{P}$. The overall probability that any acceptable option arrives turns out to be high enough (to obtain an $\Omega(1/\sqrt{\alpha})$-approximation for $\mathcal{P}$) and low enough (such that $\mathcal{A}$ wants to propose the first acceptable option rather than wait for a better one later on).

Lemma 4. *If the agent has α-bounded utilities, makes semi-oblivious proposals, and all options have low agent expectation, AlgoLow (Algorithm 4) constructs a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(1/\sqrt{\alpha})$-approximation of the expected utility for optimal (online) search.*

Proof. The set Q returned by $RestrictOptions(\mathcal{D}_1, \dots, \mathcal{D}_n, 1/2)$ guarantees $4 \cdot \sum_{(i,j) \in Q} p_{ij} b_{ij} \ge \text{OPT}$ by Lemma 1.

When splitting the set Q into classes in the beginning of the algorithm, it is guaranteed that no class spanning more than a single round has a combined probability mass greater than $1/\sqrt{\alpha}$. This means that whenever a new class is opened, the mass of the previous and the current one combined are greater than $1/\sqrt{\alpha}$. Hence, there are at most $2 \cdot \sqrt{\alpha}$ many classes in total.

Now assume class C is chosen by the algorithm and some acceptable option arrives in round i. From the assumption that utilities are α-bounded, we know that this option has an agent value of at least 1. By a union bound, the probability that any additional acceptable option from C arrives in a future round is at most $1/\sqrt{\alpha}$ (as all classes that consist of a higher mass than $1/\sqrt{\alpha}$ only have options from a single round). The conditional expectation for $\mathcal{A}$ for any acceptable option in a future round is at most $\sqrt{\alpha}$. Hence, $\mathcal{A}$ proposes the option in round i.

Similar to Algorithm 1, the probability that an action (i, j) from the chosen class is proposed is the combination of two independent events: (1) no other option from this class was proposed in a prior round $i' < i$ and (2) (i, j) is realized in round i. If the chosen class only consists of a single round, the probability for (1) is trivially 1, otherwise, we can use the same argumentation as in the proof of Lemma 2 to see that the probability that round i is reached is at least $1/2$. This means that $\mathcal{P}$ achieves an expected utility of at least $1/2 \cdot \sum_{(i,j) \in C} p_{ij} b_{ij}$.

As there are at most $2 \cdot \sqrt{\alpha}$ many classes and the algorithm chooses the best one for $\mathcal{P}$, by running AlgoLow, she will achieve an expected utility of

Algorithm 6: $\Omega(1/(\sqrt{\alpha}\log\alpha))$-approximation for semi-oblivious proposals.

```
Input: $n$ distributions $\mathcal{D}_1,\dots,\mathcal{D}_n$
Output: Action Scheme $\varphi$
Set $U = \bigcup_{i=1}^{n}\bigcup_{j=1}^{s_i}\{(i,j)\}$.
Partition $U$ into $U_L = \{(i,j)\in U \mid \sum_{k=1,\, b_{ik}=b_{ij}}^{s_i} p_{ik}a_{ik} < \sqrt{\alpha}\sum_{k=1,\, b_{ik}=b_{ij}}^{s_i} p_{ik}\}$ and $U_H = U \setminus U_L$.
for $k = 1,\dots,n$ do
    $\mathcal{D}_k^{(L)} \leftarrow \mathcal{D}_k$, $\mathcal{D}_k^{(H)} \leftarrow \mathcal{D}_k$
    In $\mathcal{D}_k^{(L)}$ set utilities of every option $(k,j)\in U_H$ to 0 for $\mathcal{P}$ and 1 for $\mathcal{A}$
    In $\mathcal{D}_k^{(H)}$ set utilities of every option $(k,j)\in U_L$ to 0 for $\mathcal{P}$ and $\sqrt{\alpha}$ for $\mathcal{A}$.
Set $\varphi_L = AlgoLow(\mathcal{D}_1^{(L)},\dots,\mathcal{D}_n^{(L)})$,   $\varphi_H = AlgoHigh(\mathcal{D}_1^{(H)},\dots,\mathcal{D}_n^{(H)})$.
return $\varphi_L$ or $\varphi_H$, whichever yields better expected utility for $\mathcal{P}$
```

$$\frac{1}{2}\cdot\sum_{(i,j)\in C} p_{ij}b_{ij} \ge \frac{1}{4\sqrt{\alpha}}\cdot\sum_{(i,j)\in Q} p_{ij}b_{ij} \ge \frac{1}{16\sqrt{\alpha}}\cdot \mathrm{OPT} = \Omega\left(\frac{1}{\sqrt{\alpha}}\right)\cdot\mathrm{OPT}\ . \quad \square$$

AlgoLow classifies options only based on utility for $\mathcal{P}$ and time intervals. AlgoHigh instead uses an approach similar to Algorithm 1, namely classifying good options for $\mathcal{P}$ by their utility for $\mathcal{A}$. Since in the semi-oblivious scenario, options from a single round i with the same utility for $\mathcal{P}$ cannot be distinguished, the algorithm classifies options by their expected utility for $\mathcal{A}$ such that the expectation for $\mathcal{A}$ of all options in a single class differs by no more than a factor 2. Finally, the algorithm identifies the best one of these $O(\log\alpha)$ many classes. Unlike in previous scenarios, the individual options in a single class can have very different utility values for the agent, up to factor α. Only the expected utility is bounded by a factor 2. To show that the resulting scheme guarantees a good approximation ratio of $\Omega(1/\sqrt{\alpha}\log\alpha)$, we show that $\mathcal{A}$ proposes the first option he encounters with a sufficiently high probability of at least $\Omega(1/\sqrt{\alpha})$.

Lemma 5. *If the agent has α-bounded utilities, makes semi-oblivious proposals, and all options have high agent expectation, AlgoHigh (Algorithm 5) constructs a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(1/(\sqrt{\alpha}\log\alpha))$-approximation of the expected utility for optimal (online) search.*

Proof. Using $RestrictOptions(\mathcal{D}_1,\dots,\mathcal{D}_n, 1/4)$, the algorithm first identifies the best options for $\mathcal{P}$. By Lemma 1, it holds that $8\cdot\sum_{(i,j)\in Q} p_{ij}b_{ij} \ge \mathrm{OPT}$.

The set Q is then further partitioned into $\lfloor\log_2\sqrt{\alpha}\rfloor + 1$ smaller classes depending on their conditional expectation for $\mathcal{A}$, namely such that the conditional expectation for $\mathcal{A}$ of the elements in a class differs by at most a factor 2. Then, the class C such that $\sum_{(i,j)\in C} p_{ij}b_{ij} \ge \sum_{(i,j)\in C'} p_{ij}b_{ij}$ for all classes C' is chosen. This means that

$$(\lfloor\log_2\sqrt{\alpha}\rfloor + 1)\cdot\sum_{(i,j)\in C} p_{ij}b_{ij} \ge \sum_{(i,j)\in Q} p_{ij}b_{ij} \ge 1/8\cdot\mathrm{OPT}\ .$$

We denote by E the lower bound for the expected $\mathcal{A}$ utility of the interval of the chosen class C. Recall that all utilities for $\mathcal{A}$ are in the interval $[1,\alpha]$. This means that with a probability of at least $E/(2\alpha - E) \ge E/(2\alpha)$, a random element from C has an agent utility of at least $E/2$ – otherwise, an expected utility of at least E for $\mathcal{A}$ would not be possible. Since the probability that another allowed option in a later round arrives is at most $1/4$ due to the choice of $m = 1/4$ for the call to *RestrictOptions* and the expectation conditional on arrival of an allowed option is at most $2E$, $\mathcal{A}$ always proposes the first option with a utility of at least $E/2$. This in turn means that the agent will propose the first element from C he encounters with a probability of at least $E/(2\alpha)$. Since $E \ge \sqrt{\alpha}$, the probability that $\mathcal{A}$ proposes the first allowed element is at least $1/(2\sqrt{\alpha})$.

In total, this means that $\mathcal{P}$ achieves an expected utility of at least

$$\frac{1}{2\sqrt{\alpha}}\cdot\sum_{(i,j)\in C} p_{ij}b_{ij} \ge \frac{1}{2\sqrt{\alpha}}\cdot\frac{\mathrm{OPT}}{8\cdot(\lfloor\log_2\sqrt{\alpha}\rfloor + 1)} = \Omega\left(\frac{1}{\sqrt{\alpha}\log_2\alpha}\right)\cdot\mathrm{OPT}\ . \quad \square$$

Note that the approximation result of Theorem 7 does *not* generalize to the case when $\mathcal{A}$ has options with utility 0, and α is the ratio of maximum and minimum non-zero utility. We observe that in this case all algorithms must have a ratio in $O(1/n)$, even when all utilities for $\mathcal{A}$ are $a_{ij}\in\{0,1\}$.

To show this, we adapt the instance from the proof of Theorem 6 as follows. We set $p_1 = 1 - 1/n$ and $(b_1,a_1) = (0,0)$, $p_2 = 1/n - 1/n^2$ and $(b_2,a_2) = (1,0)$, $p_3 = 1/n^2$ and $(b_3,a_3) = (1,1)$. Note that $\alpha = 1$ here, as there is only a single non-zero utility value for $\mathcal{A}$.

Consider any deterministic scheme for $\mathcal{P}$. Clearly, $\mathcal{A}$ does not want to propose any option of value 0 for him until the last round in which options p_2 and p_3 are acceptable. By a union bound, the overall probability to propose an option of value 1 for $\mathcal{P}$ is at most $(n-1)\cdot 1/n^2 + 1/n < 2/n$, so the expected utility of $\mathcal{P}$ is in $O(1/n)$. By searching through the options herself, $\mathcal{P}$ obtains a value of at least $1-(1-1/n)^n \geq 1-1/e$. Hence, every deterministic scheme is $O(1/n)$-approximate, even in this case with $\alpha = 1$. A similar argument shows this result also for randomized schemes.

3.2. Number of agent utility values

In this subsection, we consider scenarios in which the number of different utility values for $\mathcal{A}$ is bounded by β. We only discuss conscious proposals, because we cannot hope for more than a $\Theta(1/n)$-approximation in the (semi-)oblivious scenarios, even for a small constant β. The lower bound is a trivial consequence by Proposition 1. To see the upper bound, consider the instance to prove Theorem 6 in the semi-oblivious scenario. It involves only three different options and accordingly only three different utility values for $\mathcal{A}$ to show an upper bound of $O(1/\sqrt{\alpha})$.

Corollary 3. *If the options only have β different utility values for the agent, there is a deterministic action scheme such that the principal obtains an $\Omega(1/\beta)$-approximation of the expected utility for optimal (online) search.*

Proof. We use an approach that is very similar to Algorithm 1. The algorithm starts by identifying the best options for $\mathcal{P}$ and collects them in the set Q using RestrictOptions$(D_1,\dots,D_n,1/2)$. Then, it constructs different classes $C_1,\dots,C_\beta$ from the set Q, with one class for each individual agent value. Finally, C^* is chosen as $C^* = \arg\max_{C\in\{C_1,\dots,C_\beta\}} \sum_{(i,j)\in C} p_{ij}b_{ij}$, and the algorithm sets $\varphi_{ij}=1$ for all options $(i,j)\in C^*$ and $\varphi_{ij}=0$ otherwise.

By Lemma 1, we have $\sum_{(i,j)\in Q} p_{ij}b_{ij} \geq \mathrm{OPT}/4$. By the pigeon-hole principle, the class C^* satisfies $\sum_{(i,j)\in C^*} p_{ij}b_{ij} \geq 1/\beta \cdot \sum_{(i,j)\in Q} p_{ij}b_{ij} \geq \mathrm{OPT}/(4\beta)$.

By the construction of the classes, the agent is indifferent between any of the options in a single class, as all of them provide the same value for $\mathcal{A}$. Consequently, $\mathcal{A}$ proposes the first option that is acceptable. Our arguments above (e.g., Lemma 2 or Theorem 5) imply that the combined probability that option $(i,j)\in C^*$ is realized and proposed is at least $1/2\cdot p_{ij}$. Hence, the expected utility of $\mathcal{P}$ when using C^* is at least $\frac{1}{8\beta}\cdot\mathrm{OPT}$ and this approach guarantees an $\Omega(1/\beta)$-approximation. □

Observe that the approximation ratio of this algorithm is tight in general. Consider the instances in Theorem 1 with $\beta = n$. The theorem shows that every scheme can obtain at most a ratio of $O(1/n) = O(1/\beta)$.

4. Misalignment of principal and agent utility

In this section, we consider performance guarantees based on the amount of misalignment of principal and agent utility. For most of the section, we assume that all utility values are strictly positive.

Consider the smallest $\gamma \geq 1$ such that

$$\frac{1}{\gamma}\cdot\frac{a_{ij}}{a_{i'j'}} \leq \frac{b_{ij}}{b_{i'j'}} \leq \gamma\cdot\frac{a_{ij}}{a_{i'j'}}$$

for any two options ω_{ij} and $\omega_{i'j'}$ in the instance. Then the preference of $\mathcal{P}$ between any pair $\omega_{ij},\omega_{i'j'}$ of options is shared by $\mathcal{A}$ – up to a factor of at most γ. We term this *γ-bounded utilities*.

Alternatively, it might be intuitive to consider $\delta \geq 1$ as a direct bound on the utility ratio

$$1/\delta\cdot a_{ij} \leq b_{ij} \leq \delta\cdot a_{ij}$$

for every single ω_{ij}. Then the utilities are δ^2-bounded in the sense defined above. For conscious proposals our results below yield asymptotically the same logarithmic bounds as for γ. For semi-oblivious and oblivious proposals, we obtain bounds that are polynomial in δ.

For the rest of the section, we consider an instance with γ-bounded utilities. Suppose we choose an arbitrary realization $\omega_{i'j'}$. Divide all utility values of $\mathcal{P}$ for all realizations by $b_{i'j'}$, and all utility values of $\mathcal{A}$ by $a_{i'j'}$. Note that this adjustment neither affects the incentives of the players nor the approximation ratios of our algorithms. Considering ω_{ij} with the adjusted utilities, we see that $1/\gamma\cdot b_{ij}/a_{ij} \leq 1 \leq \gamma\cdot b_{ij}/a_{ij}$, and thus $1/\gamma \leq b_{ij}/a_{ij} \leq \gamma$ for all $\omega_{i'j'}$. This condition turns out to be convenient for our analysis.

Our main idea is to use $O(\log\gamma)$ clusters C_k to group all the options that have a utility ratio between 2^k and 2^{k+1}, i.e.,

$$C_k = \{\omega_{ij}\in\Omega \mid 2^k \leq b_{ij}/a_{ij} < 2^{k+1}\}$$

for $k = \lfloor \log 1/\gamma \rfloor,\dots,\lceil \log\gamma \rceil$. Our deterministic scheme restricts the acceptable options to a single cluster C_{k^*}. Note that here $\mathcal{P}$ is assumed to see a_{ij} upon a proposal. The principal determines the cluster C_{k^*}, such that the best response by $\mathcal{A}$ (i.e., his optimal online algorithm applied with the options from that cluster) delivers the largest expected utility for $\mathcal{P}$.

Theorem 8. *If principal and agent have γ-bounded utilities, there is a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(1/\log\gamma)$-approximation of the expected utility for optimal (online) search.*

Proof. Consider any cluster C_k. We denote by $b(\mathcal{A}, k)$ and $a(\mathcal{A}, k)$ the expected utility for $\mathcal{P}$ and $\mathcal{A}$ when $\mathcal{P}$ uses C_k to determine φ. Now consider a hypothetical algorithm for $\mathcal{P}$ that observes all realizations and chooses the best option from C_k for $\mathcal{P}$ if possible. If there is no such option, it obtains a utility of 0. Let $b(\mathcal{P}, k)$ and $a(\mathcal{P}, k)$ be the expected utility of the hypothetical algorithm for $\mathcal{P}$ and $\mathcal{A}$, respectively. Clearly, $b(\mathcal{P}, k) \geq b(\mathcal{A}, k)$ and $a(\mathcal{A}, k) \geq a(\mathcal{P}, k)$, but also, by definition of C_k,

$$b(\mathcal{A}, k) \geq a(\mathcal{A}, k) \cdot 2^k \geq a(\mathcal{P}, k) \cdot 2^k \geq b(\mathcal{P}, k)/2 \ .$$

Now consider the best option for $\mathcal{P}$ in hindsight. The best-option-algorithm for cluster C_k picks the best option in hindsight if it comes from cluster C_k. Otherwise, it returns a value of 0. Let b_k^* be the expected utility of this algorithm for $\mathcal{P}$, and let OPT be the expected utility of the best option in hindsight for $\mathcal{P}$. Then

$$\mathrm{OPT} = \sum_{k=\lfloor \log 1/\gamma \rfloor}^{\lceil \log \gamma \rceil} b_k^* \leq \sum_{k=\lfloor \log 1/\gamma \rfloor}^{\lceil \log \gamma \rceil} b(\mathcal{P}, k) \leq \sum_{k=\lfloor \log 1/\gamma \rfloor}^{\lceil \log \gamma \rceil} b(\mathcal{A}, k) \cdot 2 \ .$$

Hence, since the scheme chooses the cluster k^* that maximizes $b(\mathcal{A}, k^*)$, we obtain an $\Omega(1/\log \gamma)$-approximation. □

By treating all options of utility 0 for $\mathcal{A}$ in a separate class and ignoring all options of utility 0 for $\mathcal{P}$, we can again adapt the performance guarantee also to instances in which all utility pairs of $\mathcal{A}$ and $\mathcal{P}$ with strictly positive entries are γ-bounded.

Corollary 4. *If principal and agent have γ-bounded utilities for the set of options with only strictly positive utilities, there is a deterministic action scheme such that $\mathcal{P}$ obtains an $\Omega(1/\log \gamma)$-approximation of the expected utility for optimal (online) search.*

The bound in Theorem 1 for conscious proposals can be applied rather directly to this case, i.e., when treating the 0-utility options for $\mathcal{P}$ in a separate class. Also the bounds for oblivious and semi-oblivious proposals in Theorems 4 and 6 apply directly, since in these instances $\gamma = \Theta(\alpha)$. This implies that any algorithm has a ratio in $O(\log\log \gamma / \log \gamma)$ for conscious proposals, in $O(1/\sqrt{\gamma})$ for semi-oblivious proposals, and in $O(1/\gamma)$ for oblivious proposals. Finally, it is trivial to obtain a $\Omega(1/\gamma)$-approximation for $\mathcal{P}$ in case of γ-bounded utilities and oblivious proposals – simply accept every option proposed by $\mathcal{A}$. The bound on the ratio is a simple consequence of γ-boundedness. As such, note that $\mathcal{P}$ is not required to know γ to obtain the approximation.

CRediT authorship contribution statement

Pirmin Braun: Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Niklas Hahn:** Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Martin Hoefer:** Formal analysis, Funding acquisition, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Conrad Schecker:** Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Hahn gratefully acknowledges the support of GIF grant I-1419-118.4/2017. Hoefer gratefully acknowledges the support of GIF grant I-1419-118.4/2017, DFG Research Unit ADYN (project number 411362735), and DFG grant Ho 3831/9-1 (project number 514505843).

References

[1] Ricardo Alonso, Niko Matouschek, Optimal delegation, Rev. Econ. Stud. 75 (1) (2008) 259–293.

[2] Mark Armstrong, John Vickers, A model of delegated project choice, Econometrica 78 (1) (2010) 213–244.

[3] Moshe Babaioff, Michal Feldman, Noam Nisan, Eyal Winter, Combinatorial agency, J. Econ. Theory 147 (3) (2012) 999–1034.

[4] Moshe Babaioff, Eyal Winter, Contract complexity, in: Proc. 15th Conf. Econ. Comput. (EC), 2014, p. 911.

[5] Yakov Babichenko, Siddharth Barman, Algorithmic aspects of private Bayesian persuasion, in: Proc. 8th Symp. Innov. Theoret. Comput. Sci. (ITCS), 2017, pp. 34:1–34:16.

[6] Ashwinkumar Badanidiyuru, Kshipra Bhawalkar, Haifeng Xu, Targeting and signaling in ad auctions, in: Proc. 29th Symp. Discret. Algorithms (SODA), 2018, pp. 2545–2563.

[7] Curtis Bechtel, Shaddin Dughmi, Delegated stochastic probing, in: James R. Lee (Ed.), Proc. 12th Symp. Innov. Theoret. Comput. Sci. (ITCS), 2021, pp. 37:1–37:19.

[8] Curtis Bechtel, Shaddin Dughmi, Neel Patel, Delegated Pandora's box, in: Proc. 23rd Conf. Econ. Comput. (EC), 2022, pp. 666–693.

[9] Matteo Castiglioni, Andrea Celli, Nicola Gatti, Persuading voters: it's easy to whisper, it's hard to speak loud, in: Proc. 34th Conf. Artif. Intell. (AAAI), 2020, pp. 1870–1877.

[10] Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Nicola Gatti, Online Bayesian persuasion, in: Proc. 34th Conf. Adv. Neural Inf. Processing Syst. (NeurIPS), 2020.

[11] Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Nicola Gatti, Signaling in Bayesian network congestion games: the subtle power of symmetry, in: Proc. 35th Conf. Artif. Intell. (AAAI), 2021, pp. 5252–5259.

[12] Matteo Castiglioni, Nicola Gatti, Persuading voters in district-based elections, in: Proc. 35th Conf. Artif. Intell. (AAAI), 2021, pp. 5244–5251.

[13] Matteo Castiglioni, Alberto Marchesi, Andrea Celli, Nicola Gatti, Multi-receiver online Bayesian persuasion, in: Proc. 38th Int. Conf. Machine Learning (ICML), 2021, pp. 1314–1323.

[14] Matteo Castiglioni, Giulia Romano, Alberto Marchesi, Nicola Gatti, Signaling in posted price auctions, in: Proc. 36th Conf. Artif. Intell. (AAAI), 2022, pp. 4941–4948.

[15] Shaddin Dughmi, David Kempe, Ruixin Qiang, Persuasion with limited communication, in: Proc. 17th Conf. Econ. Comput. (EC), 2016, pp. 663–680.

[16] Shaddin Dughmi, Haifeng Xu, Algorithmic persuasion with no externalities, in: Proc. 18th Conf. Econ. Comput. (EC), 2017, pp. 351–368.

[17] Shaddin Dughmi, Haifeng Xu, Algorithmic Bayesian persuasion, SIAM J. Comput. 50 (3) (2021).

[18] Paul Dütting, Tim Roughgarden, Inbal Talgam-Cohen, Simple versus optimal contracts, in: Proc. 20th Conf. Econ. Comput. (EC), 2019, pp. 369–387.

[19] Paul Dütting, Tim Roughgarden, Inbal Talgam-Cohen, The complexity of contracts, in: Proc. 31st Symp. Discret. Algorithms (SODA), 2020, pp. 2688–2707.

[20] Ronen Gradwohl, Niklas Hahn, Martin Hoefer, Rann Smorodinsky, Algorithms for persuasion with limited communication, Math. Oper. Res. 47 (3) (2022) 2520–2545.

[21] Svenja Griesbach, Martin Hoefer, Max Klimm, Tim Koglin, Public signals in network congestion games, in: Proc. 23rd Conf. Econ. Comput. (EC), 2022, p. 736.

[22] Sanford Grossman, Oliver Hart, An analysis of the principal-agent problem, Econometrica 51 (1) (1983) 7–45.

[23] Niklas Hahn, Martin Hoefer, Rann Smorodinsky, Prophet inequalities for Bayesian persuasion, in: Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI), 2020, pp. 175–181.

[24] Niklas Hahn, Martin Hoefer, Rann Smorodinsky, The secretary recommendation problem, Games Econ. Behav. 134 (2022) 199–228.

[25] Bengt Robert Holmstrom, On Incentives and Control in Organizations, PhD thesis, Stanford University, 1977.

[26] Bengt Robert Holmstrom, On the theory of delegation, in: Marcel Boyer, Richard Kihlstrom (Eds.), Bayesian Models in Economic Theory, Elsevier, 1984, pp. 115–141.

[27] Jon Kleinberg, Robert Kleinberg, Delegated search approximates efficient search, in: Proc. 19th Conf. Econ. Comput. (EC), 2018, pp. 287–302.

[28] Ulrich Krengel, Louis Sucheston, Semiamarts and finite values, Bull. Am. Math. Soc. 83 (1977) 745–747.

[29] Ulrich Krengel, Louis Sucheston, On semiamarts, amarts and processes with finite value, Adv. Probab. 4 (1978) 197–266.

[30] Nahum D. Melumad, Toshiyuki Shibano, Communication in settings with no transfers, Rand J. Econ. 22 (2) (1991) 173–198.

[31] Aviad Rubinstein, Honest signaling in zero-sum games is hard, and lying is even harder, in: Proc. 44th Int. Colloq. Autom. Lang. Programming (ICALP), 2017, pp. 77:1–77:13.

[32] Haifeng Xu, On the tractability of public persuasion with no externalities, in: Proc. 31st Symp. Discret. Algorithms (SODA), 2020, pp. 2708–2727.

[33] You Zu, Krishnamurthy Iyer, Haifeng Xu, Learning to persuade on the fly: robustness against ignorance, in: Proc. 22nd Conf. Econ. Comput. (EC), 2021, pp. 927–928.