Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Algorithms for Persuasion with Limited Communication

Ronen Gradwohl, Niklas Hahn, Martin Hoefer, Rann Smorodinsky

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–26
ISSN 0364-765X (print), ISSN 1526-5471 (online)
http://pubsonline.informs.org/journal/moor

Algorithms for Persuasion with Limited Communication

Ronen Gradwohl,[a] **Niklas Hahn,**[b,*] **Martin Hoefer,**[b] **Rann Smorodinsky**[c]

[a] Department of Economics and Business Administration, Ariel University, Frankfurt 60325, Frankfurt am Main, Germany; [b] Institute for Computer Science, Goethe University Frankfurt, 60325 Frankfurt am Main, Germany; [c] Faculty of Industrial Engineering & Management, Technion, Haifa 3200003, Israel
*Corresponding author

Contact: roneng@ariel.ac.il, https://orcid.org/0000-0001-6332-641X (RG); nhahn@em.uni-frankfurt.de, https://orcid.org/0000-0002-4929-0542 (NH); mhoefer@em.uni-frankfurt.de, https://orcid.org/0000-0003-0131-5605 (MH); rann@technion.ac.il, https://orcid.org/0000-0001-7447-5605 (RS)

Received: March 18, 2021
Revised: August 9, 2021
Accepted: September 6, 2021
Published Online in Articles in Advance: February 4, 2022

MSC2020 Subject Classification: Primary: 91A28

https://doi.org/10.1287/moor.2021.1218

Abstract. The Bayesian persuasion paradigm of strategic communication models interaction between a privately informed *sender* and an ignorant but rational *receiver*. The goal is typically to design a (near-)optimal communication (or *signaling*) scheme for the sender. It enables the sender to disclose information to the receiver in a way as to incentivize her to take an action that is preferred by the sender. Finding the optimal signaling scheme is known to be computationally difficult in general. This hardness is further exacerbated when the message space is constrained, leading to NP-hardness of approximating the optimal sender utility within any constant factor. In this paper, we show that in several natural and prominent cases the optimization problem is tractable even when the message space is limited. In particular, we study signaling under a symmetry or an independence assumption on the distribution of utility values for the actions. For symmetric distributions, we provide a novel characterization of the optimal signaling scheme. It results in a polynomial-time algorithm to compute an optimal scheme for many compactly represented symmetric distributions. In the independent case, we design a constant-factor approximation algorithm, which stands in marked contrast to the hardness of approximation in the general case.

Funding: This work was supported by the U.S.-Israel Binational Science Foundation [Grant 2016734]; the Ministry of Science and Technology, Israel [Grant 19400214]; the National Science Foundation [Award 1718670]; the German-Israeli Foundation for Scientific Research and Development [Grant I-1419-118.4/2017]; the Deutsche Forschungsgemeinschaft [Grants DFG Ho 3831/5-1, 6-1, and 7-1]; Technion VPR grants; the Technion Center for Security Science and Technology; and the Bernard M. Gordon Center for Systems Engineering at the Technion.

Keywords: persuasion • approximation algorithms

1. Introduction

Recommendations play a vital role in the modern information economy: online retailers make product recommendations, travel websites provide advice on hotels and attractions, navigation apps suggest driving routes, and so on. In all these examples, the designers of the recommendation systems have information that consumers do not, and both sides benefit from communication. However, the interests of the consumers and the recommenders are not always aligned. For example, whereas consumers may prefer to purchase products that constitute a better bargain, retailers may prefer to sell products for which they obtain higher margins. A natural goal is to optimize the use of the retailer's informational advantage such that recommendations result in consumer choices that maximize its own benefit. In doing so, the retailer must account for the consumers' interests to guarantee that recommendations are being followed.

This optimization problem fits into the Bayesian persuasion paradigm of Kamenica and Gentzkow [34], a fundamental model of strategic communication proposed in economics that has recently gained significant interest in algorithmic game theory. In this model, there are two players: a *sender* $\mathcal{S}$ with information about a so-called *state of nature*, and a *receiver* $\mathcal{R}$, who takes an action. Payoffs of the two players are determined both by the action chosen by $\mathcal{R}$ and by the state of nature. A priori, the players do not know the true state of nature, but rather share a common belief (i.e., a distribution) over the possible outcomes. However, $\mathcal{S}$ obtains information about the *realized* state of nature and then sends a message (called a *signal*) to $\mathcal{R}$. After receiving the signal, $\mathcal{R}$ takes an action, and payoffs are realized.

A distinguishing feature of the Bayesian persuasion model is that $\mathcal{S}$ commits to a *signaling scheme* before the state of nature is realized. A signaling scheme is a (possibly randomized) function from states of nature to signals. The action for $\mathcal{S}$ can be cast as choosing a signaling scheme that determines the signal once the state of nature is realized. This problem becomes interesting, above and beyond a standard optimization, when $\mathcal{S}$ and $\mathcal{R}$ have misaligned preferences with different optimal actions in various states. How can $\mathcal{S}$ make optimal use of her informational advantage in steering $\mathcal{R}$'s choice of action?

The problem of optimally designing recommendation systems fits neatly into this model. To illustrate, consider the following simple example: $\mathcal{S}$ is a retailer that makes a product recommendation to a consumer, $\mathcal{R}$, who must choose one of the products. The various products yield different utilities to each of the players, and whereas $\mathcal{S}$ knows which product yields which utilities, from $\mathcal{R}$'s perspective, the products are randomly ordered. The state of nature is the order in which the products appear, and the signaling scheme is the recommendation system implemented by the retailer.

To make the example concrete, suppose there are three products: one product is good for $\mathcal{S}$ and bad for $\mathcal{R}$, one is bad for $\mathcal{S}$ but good for $\mathcal{R}$, and one is bad for both. Denote these respective products by *GB*, *BG*, and *BB*, and suppose they yield sender–receiver utility pairs (1, 0), (0, 1), and (0, 0) when chosen. One signaling scheme for the sender is to always reveal which product is which. In this case, $\mathcal{R}$ will choose *BG*, and $\mathcal{S}$ will attain utility zero. A better scheme for $\mathcal{S}$ is to reveal no information. Here, the best $\mathcal{R}$ can do is choose randomly, in which case $\mathcal{S}$'s utility will be $1/3$. One might attempt to improve $\mathcal{S}$'s utility by always recommending *GB*. However, this policy is not *persuasive*: $\mathcal{R}$'s optimal reaction is to deviate to choosing one of the *other* two products at random, and again $\mathcal{S}$'s utility will be zero. Nonetheless, $\mathcal{S}$ can do better than the no-information scheme by choosing a scheme that recommends *GB* with probability $2/3$ and *BG* with probability $1/3$. A straightforward calculation using Bayes' rule shows that $\mathcal{R}$ cannot improve by deviating from this recommendation, and that following it leads to a sender utility of $2/3$. This, in fact, is the optimal signaling scheme for $\mathcal{S}$.

In this paper, we study potential barriers to optimal signaling, focusing on two constraints: limited communication and limited computational resources. First, in our example above, the optimal signaling scheme needs a signal space of size three, as each of the three products could potentially be *GB* or *BG*. But what if she was restricted to sending only one of two signals? More generally, suppose there are n products, but $\mathcal{S}$ is restricted to only k signals. These restrictions arise naturally, for example, when there is a limited attention span, or communication between the players is noisy and a limited number of bits can be transferred. Typically, designing optimal signaling schemes can be based on the popular toolset developed by Kamenica and Gentzkow [34]. However, these tools no longer apply when the number of available signals is limited.

Second, from a computational perspective, finding the optimal scheme might not be tractable. Suppose in the example above there are n products, for large n. For the restricted case in which the utility pairs of the n products are independent and identically distributed (i.i.d.) and given explicitly, Dughmi and Xu [19] develop a polynomial-time algorithm that computes the optimal scheme. Note that our example above, in which the utility pairs are known but their order is not, does not fall into this case. On the other hand, for general distributions over the utility pairs of each product (and even ones that are independent but not identical), they show that computing the optimal sender utility is #P-hard (Dughmi and Xu [19]).

Third, when computational concerns are combined with limited communication, the computational problem is exacerbated. Dughmi et al. [21] prove a substantially stronger hardness result and show that it is NP-hard to even approximate the optimal sender utility to within any constant factor.

1.1. Results and Contribution

We analyze optimal signaling schemes subject to communication and computation limits in the context of two specific classes of problems that we call symmetric instances and independent instances. The first class—symmetric instances—consists of problems in which the a priori probability of any vector of n utility pairs is the same as the a priori probability of any vector in which the n elements have been permuted. The optimization problems faced by recommendation systems often consist of symmetric instances. Our example above, in which products appear in a random order and the retailer makes a product recommendation, is a symmetric instance. Another example is a navigation app that suggests driving routes. Standard and oft-used models of congestion suggest that, in equilibrium, all comparable routes between two points have the same travel time (Wardrop [48]). In practice, however, there are fluctuations, which, from the point of a driver, are random and symmetric. The navigation app obtains information about traffic conditions and makes recommendations that take into account the driver's utility—such as minimizing travel time—and its own—such as learning about changes in congestion (see, e.g., Bahar et al. [10], Kremer et al. [38]). Finally, a third set of examples captured by symmetric instances are the i.i.d. instances highlighted by Dughmi and Xu [19], in which products' utility pairs are drawn i.i.d. from a single distribution. The class is more general than these examples; in Section 2, we describe some other cases that it captures.

In Section 3.1, we study the class of symmetric instances and develop a geometric characterization of the optimal signaling scheme. In Section 3.2, we use this characterization to design an algorithm that computes optimal schemes. Our algorithm runs in polynomial time given access to a *probability oracle* that computes certain probabilities related to the instance. We then prove that the probability oracle can be implemented in polynomial time in many prominent subclasses of instances studied in related literature, including but not limited to the i.i.d. and random-order cases. Our results significantly expand the set of instances for which optimal schemes can be computed efficiently beyond the i.i.d. case in Dughmi and Xu [19].

Interestingly, our results extend even to limited signal spaces. In Section 3.4, we show that for symmetric instances, the optimal scheme for n actions and $k < n$ signals is the same scheme as the one for k actions and k signals (and a suitably adjusted distribution over utility pairs). In addition to the geometric characterization of optimal signaling schemes with limited communication, our results also imply a polynomial-time algorithm for finding such a scheme. Moreover, in Section 3.5, we show how a bicriteria approximation can be obtained in polynomial time. Such a scheme is approximately optimal for the sender as well as approximately persuasive for the receiver.

The second class of problems we study—independent instances—consists of problems in which the utility pairs are independently but not identically distributed among the n actions. For example, the optimization problem faced by travel websites that provide advice on hotels can be viewed as an independent instance: a priori, the value one hotel provides travelers may be independent of other hotels, but the possible value distribution of five-star resorts is likely different from that of one-star hostels.

In Section 4, we develop polynomial-time algorithms for finding an approximately optimal signaling scheme in a class of such instances. For general independent instances, Dughmi and Xu [19] show that finding an optimal solution is #P-hard. We obtain a constant-factor approximation when the optimal scheme must guarantee for every signal at least the best a priori utility of any action for the receiver. This is the case, for example, when an action with a priori best utility for the receiver has deterministic utility for the receiver. Alternatively, this is the case when the receiver has an outside option of a priori optimal utility. Our first algorithm in Section 4.1 is simple to state and implement and guarantees a constant-factor approximation, even in the case in which the signal space is restricted to $k < n$ signals. The ratio is at least 0.375 for $k = 2$, and it approaches $(1 - 1/e)^2 \approx 0.3996$ for large k. With a significantly more elaborate procedure in Section 4.2, we improve the approximation ratio for large k to $(1 - 1/e - \varepsilon) \approx 0.632$, for any constant $\varepsilon > 0$. With the techniques used here, it is impossible to obtain a better ratio than $1 - 1/e$.

These results stand in marked contrast to the hardness result of Dughmi et al. [21] for general instances, where restrictions on the signaling space can make the optimization problem hard to approximate within any constant factor. Our results significantly broaden the class of instances for which good approximation algorithms are known to exist.

Finally, in Section 5, we show that restricting the number of signals from n to k hurts the optimal sender utility by a (tight) factor of $\Theta(k/n)$ in symmetric instances and in the subclass of independent instances we discuss in Section 4.[4]

1.2. Techniques

Our main results on symmetric instances in Sections 3.1 and 3.2 use a geometric characterization of the optimal signaling scheme. For every state of nature, we interpret the utility pairs of the n actions as a set of points in the two-dimensional plane. Given a state of nature, the expected utility of any signaling scheme can be interpreted as a *recommendation point* inside the convex hull of the point set. We show that the optimal scheme has a symmetry property and, for every state of nature, its recommendation point is located on the Pareto frontier of the point set. Their location is such that a single common slope lies tangent to the recommendation point for every state of nature. The symmetry property allows us to tightly capture the persuasiveness constraint as a linear inequality. Using these insights, we turn the computation to solving a polynomial number of linear programs (LPs). The coefficients are probabilities derived from the Pareto frontiers of point sets of the states of nature. In this way, computing the optimal signaling scheme reduces to computing certain probabilities. We show that for a variety of symmetric distributions, such as i.i.d., random-order, prophet-secretary, or explicitly represented ones (for formal definitions, see Section 2), computing these probabilities can be done in polynomial time.

Our results provide an alternative way to compute an optimal scheme for the i.i.d. case. The previous approach of Dughmi and Xu [19] uses symmetry to apply techniques from the literature on designing optimal auctions with money. These techniques crucially rely on independence among bidders/actions. In contrast, our characterization and algorithms directly exploit the structure of the persuasion problem. We can handle correlations in the utility pairs of the state of nature and obtain efficient algorithms for symmetric instances in full generality, even in the case with limited communication.

Our approximation algorithms for independent instances in Section 4 follow a two-step approach: (a) find a good subset of k actions and (b) use each of the k signals to recommend one action from the subset. By dropping and relaxing some constraints of the optimal signaling scheme, we devise an LP relaxation. For this relaxation, we prove that step (a) becomes a submodular optimization problem, for which we use the standard greedy algorithm. For (b), we develop an algorithm turning the optimal solution of the LP relaxation into a persuasive signaling scheme. This algorithm in Section 4.1 yields an approximation ratio of roughly $(1-1/e)$ (for large k) in each of these steps. Our improved analysis in Section 4.2 then shows that for large k, the greedy algorithm for step (a) can be replaced by an FPTAS, but the factor $1-1/e$ from step (b) remains. The latter factor turns out to be tight—a further improvement must bypass the use of the LP relaxation to upper bound the optimal sender utility.

1.3. Related Literature

Originating in Aumann and Maschler's [6] work on repeated games with incomplete information, Bayesian persuasion was popularized by Kamenica and Gentzkow [34]. The many applications include financial-sector stress testing (Goldstein and Leitner [27]), medical research (Kolotilin [36]), security (Rabinovich et al. [46], Xu et al. [50, 51]), online advertising (Arieli and Babichenko [4], Badanidiyuru et al. [9], Emek et al. [23]), and voting (Alonso and Câmara [2]). Thorough overviews include those by Bergemann and Morris [11], Dughmi [18], Forges [24], and Kamenica [33].

Our paper analyzes algorithmic Bayesian persuasion with limited signal spaces, most closely related to the work of Dughmi and Xu [19] and Dughmi et al. [21]. The former give a polynomial-time algorithm to calculate the optimal scheme for i.i.d. instances and show that the problem is #P-hard in the independently but not identically distributed case. The latter focus on bilateral trade with constrained communication but prove two general results: (i) only a $O\left(\frac{\#Signals}{\#States}\right)$ factor of utility in the unconstrained communication scenario is obtainable by the sender, and (ii) it is NP-hard to approximate the optimal sender utility within a constant factor with a limited number of signals. Our work complements these, as we give an optimal polynomial-time algorithm for symmetric instances beyond the i.i.d. setting of Dughmi and Xu [19], and a polynomial-time constant-factor approximation for a class of independent instances.

Another related paper, by Aybas and Turkel [7], proves the existence of an optimal scheme when signals are limited. They also show that the sender loses at most a $2/k$ factor of utility when the number of signals decreases from k to $k-1$. We strengthen this result for symmetric instances by showing that the cumulative loss when using k instead of n signals is at most $(n-k)/n$ and this is tight. Put differently, we show matching lower and upper bounds of k/n on the fraction of the sender utility that can be obtained when using k instead of n signals. Up to small constant factors, similar bounds hold for independent instances.

More generally, extensions of algorithmic persuasion to multiple receivers have been studied by Babichenko and Barman [8] and Arieli and Babichenko [4], who focus on private signals, as well as Dughmi and Xu [20], who contrast private and public signals. Bhaskar et al. [12] and Rubinstein [47] study scenarios in which the receivers are players in games, proving various hardness results. Xu [49] gives efficient approximation algorithms for some subclasses of these scenarios. Dughmi et al. [22] employ Lagrangian duality to characterize (near-)optimal persuasion schemes and study a further extension that includes payments. For some of their scenarios, they assume symmetry of the actions that is similar to our symmetric instances. Finally, to complement the multiple-receiver setting, multiple-sender settings have been studied by Au and Kawai [5], Gentzkow and Kamenica [25, 26], Li and Norman [41], and Gradwohl et al. [28].

A different approach was taken by Hahn et al. [31, 32], who designed approximation algorithms for online versions of the single-sender, single-receiver setting. In their models, the state of nature is revealed sequentially to $\mathcal{S}$, $\mathcal{S}$ sends a signal in each round to $\mathcal{R}$, and $\mathcal{R}$ then makes a binary decision. The setting of Hahn et al. [31] is reminiscent of our independent instances, and that of Hahn et al. [32] is close to our symmetric instances with the random-order assumption. Also somewhat related is the paper of Le Treust and Tomala [40], who study a repeated setting with limited communication through a noisy channel.

Strategic communication and the study of Bayesian persuasion have also recently gained traction in operations research; see Candogan [14] for a detailed overview. Different applications have been considered. Examples include the study of revenue optimization through online retailers' information policies, for example, by signaling the availability and demand of an item (Drakopoulos et al. [17], Lingenbrink and Iyer [42]) or the purchase histories of other customers (Küçükgül et al. [39]). A second application is the classical queuing setting, in which users decide whether to join an unobservable first-in, first-out queue to obtain a service for a fixed price. The service provider can use signals to optimize the expected revenue by trying to incentivize users to join the queue (Lingenbrink and Iyer [43]). This model was further studied in a social-services setting, where users with different levels of need arrive and the service provider tries to maximize social welfare and not its own revenue (Anunrojwong et al. [3]).

Third, in the realm of networks, Candogan and Drakopoulos [15] study the following setting. The provider of a social network tries to maximize engagement of the network's user with the content while at the same time minimizing users' engagement with inaccurate information on the network by sending a (private) signal about the content. A similar setting using social networks was studied in Candogan [13]. In contrast to Candogan and Drakopoulos [15], the focus in Candogan [13] is on mechanisms using public signals, that is, all receivers seeing the same message. A fourth, broader, setting studies platform management, including work on market platforms that offer their services and their aggregated information to third-party sellers (Gur et al. [30]) or rating platforms offering information on services to their users trying to engage the "exploration" mode of their users (Papanastasiou et al. [45]). Finally, there are studies on the information policies of public health organizations that inform their members about the severity of an upcoming health crisis (Alizamir et al. [1]), or a government that warns its citizens and thereby elicits various responses (de Véricourt et al. [16]).

2. Model

2.1. Signaling with Limited Messages

There are two agents, a sender, $\mathcal{S}$, and a receiver, $\mathcal{R}$. The receiver can take one of n *actions*. We denote the set of actions by $[n] = \{1, \ldots, n\}$. Each action $i \in [n]$ has a *type* θ_i from a known type set Θ_i. We assume throughout that all type sets are finite. The state of nature $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is drawn according to a commonly known distribution over Θ, where $\Theta \subseteq \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n$. We denote the probability of drawing the state $\boldsymbol{\theta}$ by $q_{\boldsymbol{\theta}}$.

Action i's type θ_i is associated with a value pair $(\varrho(\theta_i), \xi(\theta_i))$, where $\varrho(\theta_i)$ is the value for $\mathcal{R}$ and $\xi(\theta_i)$ is the value for $\mathcal{S}$ if action i is taken by $\mathcal{R}$. Both agents want to maximize their respective expected utility from the action taken. Whereas the distribution over states of nature is common knowledge, the realized state $\boldsymbol{\theta}$ is observed only by $\mathcal{S}$. After observing $\boldsymbol{\theta}$, $\mathcal{S}$ sends some abstract signal $\sigma \in \Sigma$ to $\mathcal{R}$.

We assume that $\mathcal{S}$ has commitment power, that is, $\mathcal{S}$ commits in advance to a *signaling scheme* φ. It maps the observed state of nature $\boldsymbol{\theta}$ to a signal σ. More formally, $\varphi(\theta, \sigma)$ denotes the probability that in state θ, the scheme sends signal σ. φ is revealed to $\mathcal{R}$ before $\boldsymbol{\theta}$ is realized. The game we study proceeds as follows: (1) both players know the prior distribution q; (2) $\mathcal{S}$ commits to a signaling scheme φ and reveals it to $\mathcal{R}$; (3) the state of nature $\boldsymbol{\theta}$ is realized and is revealed to $\mathcal{S}$; (4) $\mathcal{S}$ draws signal σ according to the distribution $\varphi(\theta, \cdot)$ and sends σ to $\mathcal{R}$; (5) $\mathcal{R}$ chooses an action $i \in [n]$, and utilities are realized.

In the standard case of Bayesian persuasion with $|\Sigma| = k \geq n$, the sender can use signals to directly recommend every possible action to the receiver. In this paper, we are interested in $k < n$ when $\mathcal{S}$ might not be able to directly recommend every single action to $\mathcal{R}$. Because the case of a single signal and $k = 1$ is trivial, we assume $k \geq 2$ throughout.

We denote the expected utility for $\mathcal{X} \in \{\mathcal{S}, \mathcal{R}\}$ by $u_{\mathcal{X}}(\varphi)$ when $\mathcal{S}$ uses scheme φ and $\mathcal{R}$ best responds to φ by picking, for every signal σ, an action with optimal expected utility conditioned on observing σ. Given σ, if $\mathcal{R}$ has several optimal actions, we assume $\mathcal{R}$ breaks ties in favor of the sender.[1] If within the set of actions with best utility for $\mathcal{R}$ there are several that have best utility for $\mathcal{S}$, we assume without loss of generality (w.l.o.g.) that $\mathcal{R}$ chooses one of them via any fixed tie-breaking rule.

We will be interested in *direct* and *persuasive* schemes. In a direct scheme, $\mathcal{S}$ uses each signal to recommend a single specific action. In a persuasive scheme, the receiver has no incentive to deviate from the recommended action. When considering persuasiveness, a useful quantity is the best expected utility of any fixed action for $\mathcal{R}$, which we denote by $\varrho_E = \max_{i \in [n]} \sum_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}} \cdot \varrho(\theta_i)$.

For $\mathcal{R}$, direct and persuasive schemes offer a very simple choice: comply with the recommendation—which is good in expectation—or deviate and make a guess solely based on the known prior with an expectation that is at most as high. Hence, for a direct and persuasive scheme, a rational receiver's obvious choice of strategy is compliance with the sender's scheme. The sender, on the other hand, is tasked with finding the optimal direct and persuasive scheme. Hence, if $\mathcal{S}$ is able to find an optimal scheme, neither $\mathcal{S}$ nor $\mathcal{R}$ will deviate to a different strategy.

2.2. Symmetric Instances

In a symmetric instance, any two states of nature that are permutations of one another occur with the same probability. Formally, in a symmetric instance, $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}'}$ whenever $\boldsymbol{\theta}'$ is any permutation of $\boldsymbol{\theta}$.[2] In particular, because of symmetry, $\varrho_E = \sum_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}} \cdot \varrho(\theta_i)$ for every $i \in [n]$.

Any symmetric distribution with finite type sets can be represented rather explicitly by a set of vectors, each having n (not necessarily distinct) types, and a probability distribution over the vectors. A state of nature $\boldsymbol{\theta}$ is generated by drawing one of the vectors according to the distribution and then permuting the chosen vector uniformly at random. We denote by d the number of vectors in the representation and call this a *d-random-order scenario*. For $d = 1$, we obtain the random-order scenario from the introduction.

However, there are also interesting symmetric distributions with a much more compact representation. For the i.i.d. scenario, the natural representation is only a type distribution for a single action from which we draw n times to generate the state of nature. In the vector-based d-random-order representation, d could be exponential in the number of types for a single action. Hence, we also study a more compactly represented *prophet-secretary* scenario: Here, we have n (not necessarily distinct) distributions over types. The state of nature $\boldsymbol{\theta}$ is generated by an independent random draw from each of the n distributions and a subsequent uniform random permutation of the n types. The name stems from the literature on online algorithms. The prophet-secretary scenario strictly generalizes both i.i.d. and random-order scenarios.

For simplicity, we will assume throughout that all types are indeed distinct. Note that this assumption will be without loss of generality, because we allow distinct types to be associated with the same pair of utility values for $\mathcal{S}$ and $\mathcal{R}$.

2.3. Independent Instances

In an independent instance, every action $i \in [n]$ has a type space Θ_i. For simplicity we assume that the sets Θ_i are distinct, where we note that distinct types can have the same utility pairs. For each action $i \in [n]$, we have a distribution over types. We denote the probability of type $\theta_i \in \Theta_i$ by q_{i,θ_i}. The state of nature $\boldsymbol{\theta}$ is generated by an independent draw from each of the n distributions.

2.4. Direct and Persuasive

We assume the sender has only $2 \le k \le n$ possible signals. Every instance with k signals has an optimal direct and persuasive scheme. For symmetric instances, we can assume these are the first k actions. The proof is a simple revelation-principle-style argument.

Lemma 1. *There exists an optimal scheme with k signals that is direct and persuasive and uses the signals to recommend k distinct actions. In symmetric instances, there is an optimal direct and persuasive scheme in which $\mathcal{S}$ recommends the actions from $[k]$.*

Proof. The first statement follows from a simple revelation-principle-style argument. Consider any signaling scheme φ. Given any signal σ, we can assume $\mathcal{R}$ chooses one action that maximizes the conditional expectation of her utility. Suppose for two signals σ, σ', $\mathcal{R}$ chooses the same action. Then $\mathcal{S}$ can simply drop σ' and issue σ every time it issued σ', thereby achieving the same behavior of $\mathcal{R}$. Thus, each of the k signals can be assumed to correspond to a distinct choice of action of $\mathcal{R}$, which maximizes the conditional expectation of the utility of $\mathcal{R}$. Hence, we can equivalently assume that $\mathcal{S}$ uses the signal to issue a direct recommendation for an action such that $\mathcal{R}$ wants to follow the recommendation.

For the second statement, symmetry in the instance allows us to restrict attention to the first k actions. Consider an optimal direct and persuasive scheme φ that recommends actions from a size k subset $K \subseteq [n]$. Permute the labels of all actions in φ (and w.l.o.g. the tie-breaking rule of $\mathcal{R}$) such that it recommends actions from $[k]$. Denote the permuted φ by φ'. Because the distribution over states of nature is symmetric, it is invariant to permutation of action labels. Hence, applying φ' yields the same conditional expectations for the utility of $\mathcal{R}$ for the actions in $[k]$ as φ yields for K. Thus, φ' is direct and persuasive with recommendations from $[k]$ and $u_{\mathcal{S}}(\varphi) = u_{\mathcal{S}}(\varphi')$. □

3. Symmetric Instances

3.1. Characterization of Optimal Schemes

In this section, we derive a characterization of an optimal scheme in symmetric instances. Because of Lemma 1, we consider a direct and persuasive scheme that recommends actions from the set $[k]$. Suppose we are given a realization $\boldsymbol{\theta}$ of the state of nature. We interpret the action types as points in the two-dimensional plane. Type θ_i corresponds to point $(\varrho(\theta_i), \xi(\theta_i))$. We use C to denote the realized set of action types of the first k actions.

Given any direct and persuasive scheme φ, consider the event that the state of nature gives rise to a set C of types for the first k actions. We denote the probability of this event by $q_C = \Pr[\cup_{i \in [k]} \{\theta_i\} = C]$. Conditioned on the set C of types of the first k actions, consider the point composed of the expected utilities for $\mathcal{R}$ and $\mathcal{S}$, that is, the point $(\mathbb{E}[u_{\mathcal{R}}(\varphi) \mid C], \mathbb{E}[u_{\mathcal{S}}(\varphi) \mid C])$. Graphically, this point lies inside the convex hull of the points of C. We term this the *recommendation point for* C of φ.

More generally, let us define a *point collection*. A point collection $\mathcal{P}$ contains for each set C of action types for the first k actions a point $p(C) = (p_{\mathcal{R}}(C), p_{\mathcal{S}}(C))$ inside the convex hull of C. We define the utilities of $\mathcal{S}$ and $\mathcal{R}$ for $\mathcal{P}$ by

$$u_{\mathcal{S}}(\mathcal{P}) = \sum_C q_C \cdot p_{\mathcal{S}}(C) \quad \text{and} \quad u_{\mathcal{R}}(\mathcal{P}) = \sum_C q_C \cdot p_{\mathcal{R}}(C).$$

Observe that the recommendation points of a direct and persuasive signaling scheme are a point collection, and the utility of the collection equals the utility of the scheme, for both $\mathcal{S}$ and $\mathcal{R}$. However, in general, a point collection might not correspond to a persuasive signaling scheme.

Our interest lies in point collections where, for every subset C, the point lies on the corresponding Pareto frontier of C. Graphically speaking, the Pareto frontier of C can be assumed to start from a type with the largest sender utility with a horizontal line (possibly of length zero) with slope zero and end at a type with the largest receiver utility with a vertical line (again, possibly of length zero) with slope $-\infty$. Hence, for every slope $s \in [0, -\infty]$, there is a point on the Pareto frontier such that a line with slope s lies tangent to the Pareto frontier at this point. We say that a type or a point *corresponds to a slope* s if a line with slope s lies tangent to the Pareto frontier in the point. In Figure 1, we depict a set C where the point $p(C)$ corresponds to some slope s on the Pareto frontier.

We concentrate on point collections that satisfy the following slope condition.

Definition 1. For $s \leq 0$, a point collection $\mathcal{P}$ is *s-Pareto* if (1) for every subset C, $p(C)$ is on the Pareto frontier of C and corresponds to slope s, and (2) $u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E$.

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

Theorem 1. *For every symmetric instance, there is an optimal scheme whose recommendation points are a sender-optimal s-Pareto point collection, over all* $s \leq 0$.

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

Lemma 2. *For every direct and persuasive scheme* φ, *there is an s-Pareto point collection* $\mathcal{P}$ *with* $u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi)$.

Proof. Consider an arbitrary persuasive scheme φ that uses signals corresponding to the first k actions. Let $\mathcal{P}(\varphi)$ be the point collection of recommendation points of φ. Because φ is persuasive, the collection $\mathcal{P}(\varphi)$ satisfies the second condition of being s-Pareto. Now we adjust $\mathcal{P}(\varphi)$ in two steps to show the lemma.

First, move every recommendation point up vertically to the Pareto frontier. This improves the sender utility of the point collection but keeps the receiver utility the same. Hence, the resulting point collection $\mathcal{P}$ has all points on the Pareto frontiers and continues to satisfy $u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E$ and $u_{\mathcal{S}}(\mathcal{P}') \geq u_{\mathcal{S}}(\varphi)$. Figure 2 outlines this improvement for a set C of action types of the first k actions.

Figure 1. A set C of action types of the first k actions, where the point $p(C)$ corresponds to slope s on the Pareto frontier of C (dashed line).

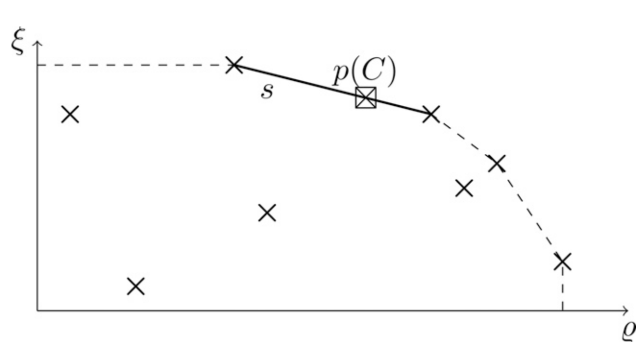
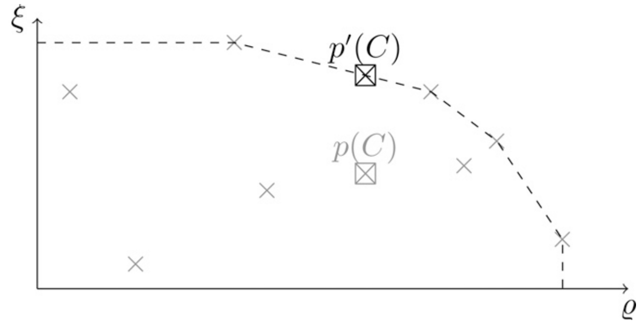


Figure 2. A set C of action types of the first k actions and a direct and persuasive scheme φ where the expected utility for $\mathcal{S}$ can be improved by moving the point $p(C) = (\mathbb{E}[u_{\mathcal{R}}(\varphi) \mid C], \mathbb{E}[u_{\mathcal{S}}(\varphi) \mid C])$ vertically upward to $p'(C)$ on the Pareto frontier of C.



Second, suppose there are different subsets $C_1 \neq C_2$ and there is no common slope that points $p(C_1)$ and $p(C_2)$ both correspond to. We use the short notation $p_1 = (\varrho_1, \xi_1) = (p_{\mathcal{R}}(C_1), p_{\mathcal{S}}(C_1))$ and $p_2 = (\varrho_2, \xi_2) = (p_{\mathcal{R}}(C_2), p_{\mathcal{S}}(C_2))$, respectively. In particular, suppose p_1 corresponds to slope s_1 and p_2 to slope $s_2 < s_1$. As the slopes are nonpositive, s_2 is "steeper" than s_1.

We construct a new point collection $\mathcal{P}_1$. For any subset $C \neq C_1, C_2$ of types of the first k actions, we keep $p(C)$. For sets C_1 and C_2, we adjust the points—we set $p_1' = (\varrho_1 + \delta q_{C_2}, \xi_1 + \delta q_{C_2} s_1)$ and $p_2' = (\varrho_2 - \delta q_{C_1}, \xi_2 - \delta q_{C_1} s_2)$ by some sufficiently small $\delta > 0$. Note that such a δ exists if p_1 and p_2 correspond to different slopes $s_1 > s_2$. If p_1 is on a "kink" of the Pareto frontier, it corresponds to a range of slopes $[s_1^{(r)}, s_1^{(l)}]$, where $s_1^{(r)}$ is the slope to the right of the kink and $s_1^{(l)}$ the one to the left. Analogously, there exists a range $[s_2^{(r)}, s_2^{(l)}]$ if p_2 is on a "kink." If these intervals do not overlap, we assume w.l.o.g. that $s_2^{(l)} < s_1^{(r)}$ and set $s_1 = s_1^{(r)}$ and $s_2 = s_2^{(l)}$. Intuitively, we move p_1 to the "right" for the set C_1 and to the "left" for C_2, thereby shifting the points along the segments on their respective Pareto frontiers. This implies that the sender utility of the point collection grows to

$$\begin{aligned} u_{\mathcal{S}}(\mathcal{P}_1) &= \sum_{C \neq C_1, C_2} q_C \cdot p_{\mathcal{S}}(C) + q_{C_1} \cdot (\xi_1 + \delta q_{C_2} s_1) + q_{C_2} \cdot (\xi_2 - \delta q_{C_1} s_2) \\ &= u_{\mathcal{S}}(\mathcal{P}) + q_{C_1} \delta q_{C_2} \cdot (s_1 - s_2) \\ &> u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi), \end{aligned}$$

because $0 \geq s_1 > s_2$. For the receiver utility,

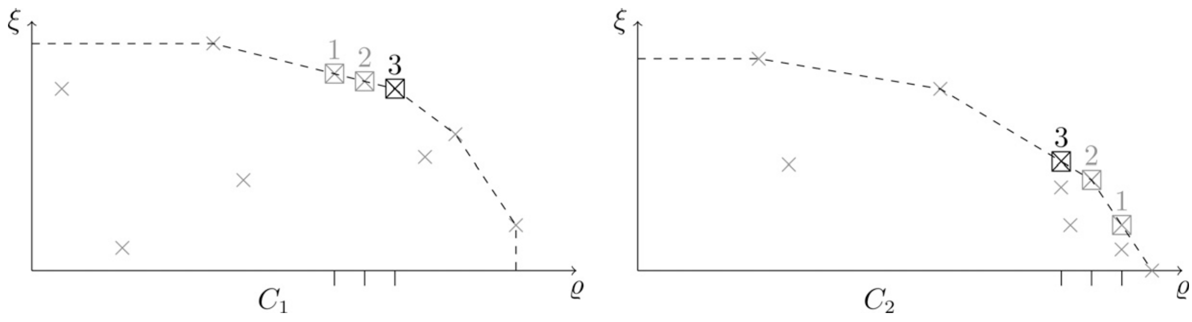
$$\begin{aligned} u_{\mathcal{R}}(\mathcal{P}_1) &= \sum_{C \neq C_1, C_2} q_C \cdot p_{\mathcal{R}}(C) + q_{C_1} \cdot (\varrho_1 + \delta q_{C_2}) + q_{C_2} \cdot (\varrho_2 - \delta q_{C_1}) \\ &= u_{\mathcal{R}}(\mathcal{P}) + q_{C_1} \delta q_{C_2} - q_{C_2} \delta q_{C_1} \\ &= u_{\mathcal{R}}(\mathcal{P}) \geq \varrho_E. \end{aligned}$$

Hence, $\mathcal{P}_1$ satisfies the second property of being s-Pareto, while improving the utility for the sender.

The factor δ is chosen such that p_1' and p_2' both stay on the line segments of slopes s_1 and s_2, respectively. Now repeated application of this modification yields collections $\mathcal{P}_2, \mathcal{P}_3, \ldots$ until finally points p_1 and p_2 correspond to at least one common slope: whenever an endpoint of a line segment is reached, if this endpoint does not correspond to a slope of the other point, the process can be continued. Moreover, we can apply this modification repeatedly as long as there are two size k sets C_1, C_2 of types with points that have no common slope. Eventually, we reach an s-Pareto point collection $\mathcal{P}$ with $u_{\mathcal{S}}(\mathcal{P}) \geq u_{\mathcal{S}}(\varphi)$. Figure 3 outlines this improvement in sender utility. □

Consider any s-Pareto point collection $\mathcal{P}$. We define a direct scheme φ^* as follows: Given a set C of types in the first k actions and the point $p(C)$, φ^* recommends one of the (at most) two actions that compose the corresponding line segment of $p(C)$ on the Pareto frontier. The actions are chosen independently of their actual number within the first k actions. By setting appropriate probabilities, the point $p(C)$ corresponds to the (conditioned on the given set C) expected utilities of φ^* for $\mathcal{S}$ and $\mathcal{R}$. This directly implies that $u_{\mathcal{S}}(\varphi^*) = u_{\mathcal{S}}(\mathcal{P})$ and $u_{\mathcal{R}}(\varphi^*) = u_{\mathcal{R}}(\mathcal{P})$.

Figure 3. The Pareto frontiers of two different sets, C_1 and C_2, of action types of the first k actions with $q_{C_1} = q_{C_2}$ and combined iterative improvement of the overall expected sender utility, such that the points labeled with "3" in both sides correspond to a common slope. The points with the same label correspond to a state in the improvement procedure. Whereas the absolute difference δ in ϱ for both C_1 and C_2 is the same in every step, the overall change in ξ is positive.



Because of symmetry of the instance and a choice of action independent of its number within the first k actions, the scheme φ^* is symmetric. A *symmetric scheme* φ (see also Dughmi and Xu [19]) is direct and recommends with each signal a distinct action in $[k]$. The conditional distribution over types (resulting from the prior and φ) is the same for each recommended action. The conditional distribution over types is the same for each nonrecommended action in $[k]$ and the same for each nonrecommended action in $[n] \setminus [k]$, no matter which (other) action is recommended. Thus, a symmetric scheme gives rise to three distributions over types: a distribution $\mathcal{D}_{yes}$ for any recommended action, a distribution $\mathcal{D}_{no}$ for any nonrecommended action in $[k]$, and a distribution $\mathcal{D}_{never}$ for any nonrecommended action in $[n] \setminus [k]$. For symmetric schemes, we show that persuasiveness is equivalent to the following simple constraint.

Lemma 3. *In symmetric instances, a symmetric scheme φ is persuasive if and only if $u_{\mathcal{R}}(\varphi) \geq \varrho_E$.*

Proof. Clearly, if a scheme φ guarantees strictly less utility than ϱ_E to $\mathcal{R}$, then $\mathcal{R}$ could profit by deviating to, say, action 1 throughout. Hence, $u_{\mathcal{R}}(\varphi) \geq \varrho_E$ is necessary for every persuasive scheme φ.

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

The symmetric scheme φ^* based on an s-Pareto point collection satisfies the constraint in Lemma 3 by definition. As such, we obtain the following result, which finishes the proof of Theorem 1.

Lemma 4. *For every s-Pareto point collection $\mathcal{P}$, there is a symmetric, direct, and persuasive signaling scheme φ^* with $u_{\mathcal{S}}(\varphi^*) = u_{\mathcal{S}}(\mathcal{P})$.*

3.2. Efficient Computation of Optimal Schemes

The Slope-Algorithm (Algorithm 1) systematically enumerates a set S containing all meaningful candidate slopes s for an s-Pareto point collection. For every pair of types a and b, the algorithm determines the probability (denoted by p_{ab}) that their line segment (denoted by $\overline{ab}$) is contained in the Pareto frontier of the set C of realizations of the first k actions. For every pair with $s > 0$, one type Pareto dominates the other, and the pair can be discarded. Similarly, if $p_{ab} = 0$, the pair can be discarded. The critical step in the first part of the algorithm is the computation of p_{ab} in line 4. For now, we assume that the algorithm has oracle access to these quantities via a *probability oracle*. We will discuss below how to implement the probability oracle in polynomial time.

Algorithm 1 (Slope-Algorithm)
Input: Symmetric instance with set $\Theta = \Theta_1 = \ldots = \Theta_n$ of action types and distribution q
1. $S \leftarrow \emptyset$, $L \leftarrow \emptyset$
2. **for** *every pair of types* $a, b \in \Theta$, $a \neq b$ **do**

3. Let s be the slope of $\overline{ab}$ and set $p_{ab} \leftarrow 0$
4. **if** $s \leq 0$ **then** determine probability p_{ab} that $\overline{ab}$ is on the Pareto frontier of types of actions in $[k]$
5. **if** $p_{ab} > 0$ **then** $S \leftarrow S \cup \{s\}$
6. Sort the slopes of S: $s_1 < s_2 < \ldots < s_\ell$
7. Pick $\ell + 1$ auxiliary slopes: $t_1 < s_1 < t_2 < s_2 < \ldots < s_\ell < t_{\ell+1}$
8. $S \leftarrow S \cup \{t_1, \ldots, t_{\ell+1}\}$
9. **for** *every slope* $s \in S$ **do**
10. **for** *every type* $c \in \Theta$ **do**
11. Determine probability $p_c^{(s)}$ that c is the unique point corresponding to s on the Pareto frontier of types of actions in $[k]$
12. Solve LP (1) to determine an s-Pareto point collection
13. **if** LP (1) has feasible optimal solution $\boldsymbol{\alpha}^{(s)}$ **then** $L \leftarrow \{(\boldsymbol{\alpha}^{(s)}, s)\}$
14. **return** *best point collection in L with corresponding slope*

At the end of the first "for" loop, the algorithm has collected in S all meaningful slopes of nonempty segments that can appear on the Pareto frontier of the types of the first k actions. In addition to these slopes, every Pareto frontier can be assumed to contain all slopes from $[0, -\infty)$. An optimal scheme might not necessarily correspond to a slope of any nonempty segment attained in the first "for" loop. If it does not, it must correspond to some slope t with $s_i < t < s_{i+1}$. Note that all slopes $t \in (s_i, s_{i+1})$ correspond to the same point on the Pareto frontier. Hence, t_i in line 7 can be chosen arbitrarily.

Now even if a slope s is attained by some segment $\overline{ab}$, it might be that for some other subset of types C, slope s corresponds to only a single point on the Pareto frontier of C. As such, the algorithm also determines in line 11 for every $s \in S$ the probability that a single type $c \in \Theta$ corresponds to s on the Pareto frontier of C. This is the critical step in the second part of the algorithm. Again, we assume that the algorithm has oracle access to these quantities via a probability oracle. We will discuss in the next section how to implement the probability oracle in polynomial time.

Finally, after having computed all probabilities, the algorithm solves the following LP:

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

For the LP, we assume that s is the common slope of the point collection. Clearly, for all subsets C where a single point c corresponds to slope s, the choice is trivial. For all subsets C in which some line segment $\overline{cd}$ with slope s is on the Pareto frontier, there is a choice to pick a point from that segment. This choice is represented by the variable $\alpha_{cd}^{(s)} \in [0,1]$. The LP optimizes point locations to maximize the expected utility for $\mathcal{S}$ (in the objective function) and to guarantee at least the average utility of ϱ_E for $\mathcal{R}$. For a given slope s, the LP might be infeasible. However, by enumerating all relevant common slopes, the algorithm sees at least one feasible solution. It returns the best feasible linear programming solution along with the slope s^*.

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

Proposition 1. *Given an efficient algorithm to compute the probability oracle, the Slope-Algorithm computes an optimal direct and persuasive scheme for symmetric instances in polynomial time.*

Proof. Correctness follows from the characterization in the last section and the observations above. We denote the maximal running time of the probability oracle by T_o and the maximal time needed to solve LP (1) by T_{LP}. Let $m = |\Theta|$ denote the finite number of types. Then finding the slopes can be done in time $O(m^2 \cdot T_o)$. Sorting the slopes needs time $O(m^2 \log m)$. For the second "for" loop, we iterate through $O(m^2)$ slopes. For each slope, we need at most m calls to the probability oracle and solve one LP of polynomial size. Overall, the running time is $O(m^3 \cdot T_o + m^2 \cdot T_{LP} + m^2 \log m)$. □

3.3. Efficient Probability Oracles

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

Theorem 2. *An optimal signaling scheme with k signals can be computed in polynomial time for the prophet-secretary and the d-random-order scenarios.*

We divide the proof of Theorem 2 into two subsections for the prophet-secretary and the d-random-order scenarios.

3.3.1. Prophet-Secretary Scenario.

Consider the prophet-secretary scenario, in which we have n probability distributions over type spaces $\Theta^1, \ldots, \Theta^n$, respectively. For simplicity, we reverse the generation process of the state of nature $\boldsymbol{\theta}$: First, permute the n distributions in uniform random order, then draw a single type from each distribution independently.

We denote by q^i_θ the probability that type $\theta \in \Theta^i$ is drawn from distribution i. For the n type spaces, we assume w.l.o.g. that they are mutually disjoint. In addition, we assume for simplicity that types are in general position, that is, there are no more than two distinct types on any given straight line. We discuss in the end how our observations can be adapted when this assumption does not hold.

Overall, the representation size of the input is at least linear in n, $\max_i |\Theta^i|$, and $\max_{i,\theta} \log 1/q^i_\theta$. For a polynomial-time probability oracle, we have to implement two classes of queries in time polynomial in the aforementioned quantities:

a. Given a pair of types a and b, return the probability p_{ab} that $\overline{ab}$ is in the Pareto frontier of the type set C of the first k actions.

b. Given a type c and a slope s, return the probability $p^{(s)}_c$ that c is the unique point that corresponds to slope s on the Pareto frontier of the type set C of the first k actions.

3.3.1.1. Class a.

If the two types are from the same distribution, then $p_{ab} = 0$. Otherwise, let i_a and i_b be such that $a \in \Theta^{i_a}$ and $b \in \Theta^{i_b}$. For each distribution $i \neq i_a, i_b$, we consider every type $c \in \Theta^i$. If c lies above the line through a and b and is included in C, then c lifts the Pareto frontier above $\overline{ab}$, and the segment vanishes from the Pareto frontier. Thus, if c lies above the line through a and b, then c must not be in C. Otherwise, c is an *allowed type*. We denote by Θ^i_{ab} the set of allowed types of distribution i, and by $q^i_{ab} = \sum_{c \in \Theta^i_{ab}} q^i_c$ the probability to draw an allowed type in distribution i. Clearly, these probabilities can be determined in time linear in the total number of types.

Now, in order to have $\overline{ab}$ on the Pareto frontier, it must be the case that (1) distributions i_a and i_b are permuted to the first k actions; (2) a and b are drawn from distributions i_a and i_b, respectively; and (3) for every other distribution i permuted to the first k actions, we draw an allowed type. The probability for (1) is $k/n \cdot (k-1)/(n-1)$, and the probability for (2) is $q^{i_a}_a \cdot q^{i_b}_b$. To compute the probability of (3), we consider every subset $A \subseteq \{1, \ldots, n\} \setminus \{i_a, i_b\}$ of $|A| = k-2$ distributions and compute the probability that from every distribution of A we draw an allowed type. Overall,

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

To compute the last term, we need to compute the sum of products of all $(k-2)$-size subsets of $n-2$ numbers. This can done in time $O(nk)$ using a dynamic program.

3.3.1.2. Class b.

Let i_c be such that $c \in \Theta^{i_c}$. For each distribution $i \neq i_c$, we again consider every type $d \in \Theta^i$. Point type c shall be the unique point corresponding to slope s on the Pareto frontier, so there must not be any type on or above the line going through c with slope s. Hence, all types that remain strictly below this line are allowed types. We denote by Θ^i_c the set of allowed types of distribution i, and by $q^i_c = \sum_{d \in \Theta^i_c} q^i_d$ the probability to draw an allowed type in distribution i. These probabilities can be determined in time linear in the total number of types.

For c to be the unique point that corresponds to s on the Pareto frontier, it must be the case that (1) distribution i_c is permuted to the first k actions, (2) c is drawn from distribution i_c, and (3) for every other distribution i permuted to the first k actions, we draw an allowed type. The probability for (1) is k/n, and the probability for (2) is $q^{i_c}_c$. To compute the probability of (3), we consider every subset $A \subseteq \{1, \ldots, n\} \setminus \{i_c\}$ of $|A| = k-1$ distributions and compute the probability that from every distribution of A we draw an allowed type. Overall,

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

Again, the last term can be computed by a dynamic program in time $O(nk)$.

3.3.1.3. On General Position. When types are not in general position, that is, there are three or more types on a straight line, the events of them forming a line segment on the Pareto frontier are not disjoint. Hence, the probabilities to set up the LP have to be computed in a slightly different manner.

We ensure that a segment is not counted multiple times by considering, for any given slope, only the longest possible line segment in the Pareto frontier. Hence, the following modification has to be made for queries of class a when determining the set of allowed types Θ_{ab}^i: All types of Θ^i that are on the segment $\overline{ab}$ are allowed because this does not prohibit $\overline{ab}$ from being the longest possible line segment. All types of Θ^i that are on the straight line going through a and b but not on $\overline{ab}$ must not be allowed. With this modification to calculate the probabilities for p_{ab}, general position of types is no longer required.

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

Proposition 2. *For the prophet-secretary scenario, we can implement a probability oracle for the Slope-Algorithm in polynomial time.*

3.3.2. *d*-Random-Order Scenario. For d-random-order instances, we have d type vectors $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^d$ and a distribution over these vectors. We denote by $q_{\boldsymbol{\theta}^i}$ the probability of $\boldsymbol{\theta}^i$. Without loss of generality, we assume that all dn types in the d vectors are distinct. To generate a state of nature, we draw vector $\boldsymbol{\theta}^i$ with probability $q_{\boldsymbol{\theta}^i}$ and then permute the vector uniformly at random. The representation size of the input is linear in dn and $\max_i \log 1/q_{\boldsymbol{\theta}^i}$. For a polynomial-time probability oracle, we again have to implement two classes of queries discussed in the previous section. The running time will be polynomial in the aforementioned quantities. For simplicity, we again assume types are points in general position.

3.3.2.1. Class a. If the two types a and b come from different vectors $\boldsymbol{\theta}^j$ and $\boldsymbol{\theta}^{j\prime}$, then $p_{ab} = 0$. Otherwise, suppose a and b are from $\boldsymbol{\theta}^j$. Consider each type c from $\boldsymbol{\theta}^j$ with $c \neq a, b$. If c lies above the line through a and b and is included in C, then c lifts the Pareto frontier above $\overline{ab}$, and the segment vanishes from the Pareto frontier. Thus, if c lies above the line through a and b, then c must not be in C. Otherwise, c is an allowed type. We denote by A_{ab}^j the set of allowed types from vector $\boldsymbol{\theta}^j$. Clearly, A_{ab}^j can be computed in time linear in n.

Now, in order to have $\overline{ab}$ on the Pareto frontier, it must be the case that (1) $\boldsymbol{\theta}^j$ is drawn from the distribution, (2) a and b are permuted to the first k actions, and (3) every other type from $\boldsymbol{\theta}^j$ permuted to the first k actions is an allowed type. The probabilities for these events are (1) $q_{\boldsymbol{\theta}^j}$, (2) $\frac{k}{n} \cdot \frac{k-1}{n-1}$, and (3) $\binom{|A_{ab}^j|}{k-2} \Big/ \binom{n-2}{k-2}$, where we assume that $\binom{|A_{ab}^j|}{k-2} = 0$ if $|A_{ab}^j| < k-2$. Overall,

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

Clearly, this expression can be computed in polynomial time for every pair of types a, b.

3.3.2.2. Class b. Let c be a type from vector $\boldsymbol{\theta}^j$. Consider each type d from $\boldsymbol{\theta}^j$ with $d \neq c$. Type c shall be the unique point corresponding to slope s on the Pareto frontier, so d must not be on or above the line going through c with slope s. If d remains strictly below this line, it is an allowed type. We denote by A_c^j the set of allowed types from $\boldsymbol{\theta}^j$. Clearly, A_c^j can be computed in time linear in n.

For c to be the unique point that corresponds to s on the Pareto frontier, it must be the case that (1) $\boldsymbol{\theta}^j$ is drawn from the distribution, (2) c is permuted to the first k actions, and (3) every other type from $\boldsymbol{\theta}^j$ permuted to the first k actions is an allowed type. The probabilities for these events are (1) $q_{\boldsymbol{\theta}^j}$, (2) $\frac{k}{n}$, and (3) $\binom{|A_c^j|}{k-1} \Big/ \binom{n-1}{k-1}$, where we assume that $\binom{|A^j_{ab}|}{k-1} = 0$ if $|A^j_{ab}| < k-1$. Overall,

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

Again, the expression can be computed in polynomial time for every type c.

The adjustments to remove the assumption of general position are the same as for the prophet-secretary scenario in the previous section. The main insight in this section is summarized in the following proposition.

Proposition 3. *For the d-random-order scenario, we can implement a probability oracle for the Slope-Algorithm in polynomial time.*

3.4. Truncated Instances

For a symmetric instance with k signals and n actions, consider a *truncation* operation: Remove actions $k+1, \ldots, n$ from consideration, and restrict every state of nature $\boldsymbol{\theta}$ to its first k entries. This yields the *truncated instance* with k signals and k actions. Suppose we apply the characterization from Theorem 1 and the Slope-Algorithm to compute an optimal scheme in the original instance and the truncated instance. Indeed, it is a straightforward consequence of symmetry that the resulting scheme is the same.

Theorem 3. *For symmetric instances with k signals and n actions, there is an optimal scheme that is an optimal scheme for the truncated instance with k signals and k actions, and vice versa.*

We show that one can apply algorithms to the truncated instance and obtain similar results for the scenario with $k < n$ actions. By truncating the instance, we return to the standard scenario of Bayesian persuasion with $n = k$ actions and signals.

In particular, for the i.i.d. scenario, truncation yields an instance where we draw from the same underlying distribution simply for k instead of n actions. Hence, an optimal scheme with n i.i.d. actions and k signals is an optimal scheme for k i.i.d. actions and k signals. Instead of using the Slope-Algorithm, it can also be obtained by solving a single LP of polynomial size (Dughmi and Xu [19]).

3.5. Bicriteria Approximation

In the following section, we show how a bicriteria approximation for symmetric instances can be obtained using a result from Dughmi and Xu [19] and the result from the previous Section 3.4.

Having black-box access to the prior over states of nature, we obtain a bicriteria approximation by using a Monte Carlo sampling approach. In the following, we assume that all utility values are in $\varrho(\theta_i), \xi(\theta_i) \in [-1, 1]$. We assume $\mathcal{S}$ has black-box oracle access to the prior, that is, she can draw states of nature as samples from the distribution.

Let φ^* be an optimal direct and persuasive scheme. Given any parameter $\varepsilon > 0$, a direct scheme φ is ε-persuasive if $\mathbb{E}[\varrho(\theta_i) \mid \sigma = i] \ge \mathbb{E}[\varrho(\theta_j) \mid \sigma = i] - \varepsilon$ for all actions $j \in [n]$. A direct scheme is ε-*optimal* if $u_{\mathcal{S}}(\varphi) \ge u_{\mathcal{S}}(\varphi^*) - \varepsilon$, where for $u_{\mathcal{S}}(\varphi)$ we assume that $\mathcal{R}$ follows the recommendation. An ε-persuasive and ε-optimal scheme gives both players a guarantee that their expected utility is at most an (additive) ε away from a utility benchmark. For $\mathcal{R}$, the benchmark is the utility of the best action given φ, and for $\mathcal{S}$, it is the utility obtained by an optimal persuasive scheme.

The main result of this section is that the bicriteria FPTAS from Dughmi and Xu [19] can be applied to the truncated instance.

Corollary 1. *In symmetric instances with $k \le n$ signals, utility values in $[-1, 1]$, and black-box oracle access to the distribution over states of nature, an ε-persuasive and ε-optimal scheme can be computed in time polynomial in n and $1/\varepsilon$, for every $\varepsilon > 0$.*

Proof. We apply the bicriteria FPTAS from Dughmi and Xu [19] to the truncated instance, which implies the result for the truncated instance. We now argue that the guarantees of ε-optimal and ε-persuasive also apply in the original instance. Because we observed above that there is a scheme that is optimal in both the truncated and original instances, ε-optimality is immediate. It remains to show ε-persuasiveness.

In addition to the given state of nature, the scheme draws a polynomial number of independent samples from the black-box oracle. It then computes the optimal direct and ε-persuasive scheme for the uniform distribution over the sample set. This is done by solving an LP of polynomial size (see Dughmi et al. [21, section 5.1]). In the solution of the LP, we assume that all ties are broken uniformly at random. Then, if we permute all states of nature in the sample in the same way, the resulting scheme also permutes the signal distributions in the same way. Because of symmetry in the instance, every permutation is equally likely. As a consequence, the resulting scheme is symmetric. We denote by ϱ_{yes}, ϱ_{no}, and ϱ_{never} the expected utility for $\mathcal{R}$ for the distributions of recommended action, nonrecommended action in $[k]$, and nonrecommended action in $[n] \setminus [k]$, respectively. Note that $\varrho_{never} = \varrho_E$. Because of symmetry, as in Lemma 3, we have $1/k\,\varrho_{yes} + (k-1)/k\,\varrho_{no} = \varrho_E$, and because of ε-persuasiveness in the truncated instance, we know $\varrho_{yes} \geq \varrho_{no} - \varepsilon$. Combining the two inequalities leads to $\varrho_{yes} \geq (k\varrho_E - \varrho_{yes})/(k-1) - \varepsilon$, which implies $\varrho_{yes} \geq \varrho_{never} - (k-1)/k \cdot \varepsilon$. □

4. Independent Instances

In this section, we move away from symmetric instances and concentrate on the case of independent actions. For such instances, computing the expected utility for $\mathcal{S}$ is #P-hard, even in the standard case with n actions and n signals (Dughmi and Xu [19]). We discuss how to obtain a persuasive scheme for k signals that guarantees a constant-factor approximation to the optimal sender utility for k signals.

We first identify an action with the highest a priori utility ϱ_E for $\mathcal{R}$. If there are multiple such actions, pick one that maximizes the expected utility for $\mathcal{S}$. If there are several of these, pick an arbitrary one from these. We renumber the actions such that this is action n. Our signaling schemes use k signals to recommend a set $S \cup \{n\}$ of k actions. The signal for action n plays the role of a dummy signal (cf. Dughmi et al. [19]).

Our algorithm applies in independent instances, in which there is an optimal scheme φ^* such that $\mathcal{R}$ obtains a conditional expectation of at least ϱ_E for every signal. We term this condition *ϱ_E-optimality*. For example, ϱ_E-optimality is fulfilled when there is an action that has deterministic utility of ϱ_E for $\mathcal{R}$ (but possibly randomized utility for $\mathcal{S}$). Then $\mathcal{R}$ can always secure a value of ϱ_E by choosing this action. As such, to be persuasive, φ^* must guarantee at least a conditional expected utility of ϱ_E for every signal.

Our signaling schemes consist of two steps: (a) choose a suitable set S of $k-1$ actions, and (b) given any set $S \cup \{n\}$ of k actions, compute a signaling scheme that recommends one of these actions. We give two variants that follow this approach. First, in Section 4.1, we consider the independent scheme φ_{IS} based on a greedy algorithm for step (a). The approximation guarantee is given in the subsequent theorem. It is $3/8 = 0.375$ for $k=2$. For $k \to \infty$, it approaches $(1-1/e)^2 \approx 0.3996$.

Theorem 4. *The independent scheme φ_{IS} is a direct and persuasive scheme for ϱ_E-optimal independent instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$,*

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

Subsequently, in Section 4.2, we describe an improved procedure to compute a good set S in step (a). This improves the approximation ratio considerably for larger values of k. The ratio is at least $0.375 - \varepsilon$ for $k = 2$. For $k \to \infty$, it is at least $1 - 1/e - \varepsilon$.

Theorem 5. *The improved independent scheme φ_{IIS} is a direct and persuasive scheme for ϱ_E-optimal independent instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$ and every constant $\varepsilon > 0$,*

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

We observe below that for large values of k, this is essentially a tight guarantee for our approach. A further improvement of the approximation ratio requires significantly different techniques.

4.1. Constant-Factor Approximation

In this section, we describe the independent scheme and prove Theorem 4. For each type set Θ_i, we, w.l.o.g., include a sufficient number of dummy types θ_i with $q_{i,\theta_i} = 0$ and assume that $|\Theta_i| = |\Theta_j| = m$, for all $i, j \in [n]$. We use $[m]$ to enumerate the possible types of each action i. Now for any subset $S \subseteq [n-1]$ of the first $n-1$ actions, consider a set function $f : 2^{[n-1]} \to \mathbb{R}$ defined by

$$f(S) = \max\left\{ \sum_{i \in S \cup \{n\}} g_i(z_i) \;\middle|\; \sum_{i \in S \cup \{n\}} z_i \leq 1 \text{ and } z_i \geq 0 \;\forall\, i \in S \cup \{n\} \right\}, \tag{2}$$

where

$$\begin{aligned} g_i(z) = \text{Max.} \quad & \sum_{j=1}^{m} x_{ij}\xi_{ij}, \\ \text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} \le z, \\ & \sum_{j=1}^{m} x_{ij}\varrho_{ij} \ge \varrho_E \cdot \sum_{j=1}^{m} x_{ij}, \\ & x_{ij} \in [0, q_{ij}] \quad \forall j \in [m]. \end{aligned} \tag{3}$$

For an intuition, we interpret z_i as an overall probability of a signal for action i. Then $g_i(z_i)$ maximizes the expected utility for the sender conditioned on a probability mass of z_i on action i. In LP (3), x_{ij} describes the portion of the probability mass on type j of action i. The first constraint of LP (3) limits the total mass of action i to at most z. The second constraint ensures that the conditional expected utility of $\boldsymbol{x}$ for $\mathcal{R}$ is at least ϱ_E. Finally, the last constraint states that the probability of a signal for type j is at most the probability that type j is realized.

Consider any direct and persuasive scheme $\varphi_{S\cup\{n\}}$ that uses $|S|+1$ signals to recommend the actions $S \cup \{n\}$. Suppose x_{ij} is the ex post probability to recommend action i with type j in $\varphi_{S\cup\{n\}}$. Clearly, the constraints in (2) and (3) do not fully capture the constraints on x_{ij}. However, all constraints are necessary. In particular, setting x_{ij} to the ex post probability of recommending action i with type j in the optimal scheme $\varphi^*_{S\cup\{n\}}$ gives a feasible solution for every LP (3), and $z_i = \sum_{j=1}^{m} x_{ij}$ is feasible for (2) (cf. Hahn et al. [31, lemma 1]). Hence, for any given subset $S \cup \{n\}$ of recommended actions, $f(S)$ is an upper bound on the optimal sender utility, that is, $f(S) \ge u_{\mathcal{S}}(\varphi^*_{S\cup\{n\}})$.

Now, consider the independent scheme φ_{IS}. It consists of two steps: (a) choose a suitable set S of $k-1$ actions, and (b) given any set $S \cup \{n\}$ of k actions, compute a signaling scheme that recommends one of these actions. Step (a) is done in ActionsGreedy (Algorithm 2), and step (b) in ComputeSignal (Algorithm 3).

Algorithm 2 (ActionsGreedy)

Input: Type sets $\Theta_1, \ldots, \Theta_n$ and distributions $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, s.t. $\sum_j q_{n,j}\varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j}\xi_{nj} = \max_{i\in[n]:\sum_j q_{i,j}\varrho_{ij}=\varrho_E} \sum_j q_{i,j}\xi_{ij}$, parameter $2 \le k \le n$

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

Algorithm 3 (ComputeSignal)

Input: Type sets $\Theta_1, \ldots, \Theta_n$ and distributions $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, s.t. $\sum_j q_{n,j}\varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j}\xi_{nj} = \max_{i\in[n]:\sum_j q_{i,j}\varrho_{ij}=\varrho_E} \sum_j q_{i,j}\xi_{ij}$, parameter $2 \le k \le n$, set $S \subseteq [n-1]$ with $|S| = k-1$

1. For $i \in S \cup \{n\}$, let z_i^* and $\boldsymbol{x}_i^*$ be the values of the optimal solution in $f(S)$
2. Order actions in $S \cup \{n\}$ such that $\frac{g_{i_1}(z^*_{i_1})}{z^*_{i_1}} \ge \ldots \ge \frac{g_{i_k}(z^*_{i_k})}{z^*_{i_k}}$, where we assume $0/0 = 0$
3. **for** $\ell = 1, \ldots, k$ **do**
4. Observe type j of action i_ℓ and flip independent coin with probability $x^*_{i_\ell,j}/q_{i_\ell,j}$ for heads
5. **if** coin comes up heads **then return** *signal for action* i_ℓ
6. **return** *signal for action* n

We start our analysis by bounding the approximation of φ_{IS} in terms of optimal sender utility. Toward this end, we observe that ActionsGreedy implements the greedy algorithm for submodular maximization.

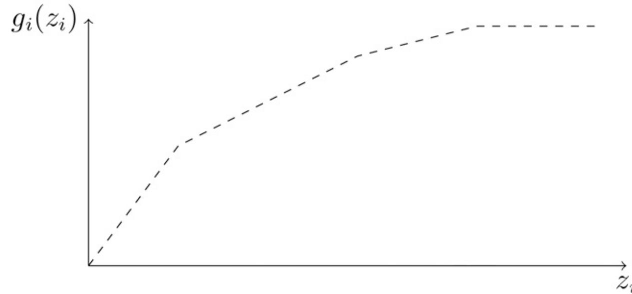
Lemma 5. *The function f is nonnegative, nondecreasing, and submodular.*

We illustrate g_i in Figure 4 above. For all i, g_i is piecewise linear and concave. This means that increasing z_i for some i yields diminishing returns, and we can maximize $f(S)$ by using a water-filling approach, that is, for all $i \in S$ increasing z_i for all g_i with the same slope at z_i such that we can maintain a common slope for all $i \in S$.

To show submodularity of f, we use an auxiliary function f' to bound the marginal increase in f to show that $f(T\cup\{j\}) - f(T) \le f(S\cup\{j\}) - f(S)$ for $j \notin T \supseteq S$.

Proof. The function f is clearly nonnegative and nondecreasing, becausece every g_j is nonnegative, piecewise linear, and concave. Hence, $f(S\cup\{j\})$ can only improve over $f(S)$. To see that f is submodular, note that f optimally distributes a unit of mass to a set of monotone, concave functions. Observe that the optimal assignment of z_i^* in $f(S)$ for all $i \in S$ can be reached through a water-filling approach. Keeping the same slope for all g_i (where we assume w.l.o.g. that a breakpoint between linear segments in g_i represents all intermediate slopes), $z_i^*, i \in S$ are

Figure 4. Schematic of a function g_i (see (3)) used for submodular function f in (2).



increased until $\sum_{i \in S} z_i^* = 1$. Consider the common slope of the functions g_i for $i \in S$ resulting from the optimal water-filling assignment of z_i^* in $f(S)$. When going from S to $S \cup \{j\}$, the slope can only decrease. As a consequence, when adding more elements to S, the z_i^* are nonincreasing.

Consider $S \subseteq T$ and $j \notin T$. Let z_j^S be the optimal choice in $f(S \cup \{j\})$ and z_j^T be the one in $f(T \cup \{j\})$. Note that $z_j^S \ge z_j^T$. Now assume that for $f'(S \cup \{j\})$, we allow assigning only at most z_j^T to g_j. Then $f'(S \cup \{j\}) \le f(S \cup \{j\})$, because in the former, a mass of $z_j^S - z_j^T$ yields a smaller growth in value because of assignment to $i \ne j$ with a smaller slope. When shifting from $f(S)$ to $f'(S \cup \{j\})$ and from $f(T)$ to $f(T \cup \{j\})$, in both cases the increase at j is $g_j(z_j')$, and a mass of z_j^T is removed from the remaining functions. This has a stronger effect in S, because the removal occurs at a higher slope. Overall, $f(T \cup \{j\}) - f(T) \le f'(S \cup \{j\}) - f(S) \le f(S \cup \{j\}) - f(S)$. □

By Lemma 1, we can assume that the optimal scheme φ^* directly recommends a set K of k actions.

Lemma 6. *For every $k \ge 2$, ActionsGreedy computes a subset S of $k-1$ actions such that*

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

Proof. ActionsGreedy is a standard greedy algorithm for submodular maximization. Note that

$$u_{\mathcal{S}}(\varphi^*) \le u_{\mathcal{S}}(\varphi^*_{K \cup \{n\}}) \le f(K) \le f(S_k^*),$$

where $S_k^* \in \arg\max\{f(S) \mid S \subseteq [n-1], |S| = k\}$. The action n is a priori receiver optimal, and in our scheme below, it will play the role of an outside option, a baseline or dummy signal (cf. Dughmi et al. [21], Hahn et al. [31]). However, it is not necessarily part of the optimal subset K of signals. As such, we overestimate the optimal value by $f(S_k^*)$, the best set of $k+1$ recommended actions, one of which must be action n.

A simple generalization of the standard analysis in Nemhauser et al. [44] (see, e.g., Krause and Golovin [37, theorem 1.5]) shows that for this case, the greedy solution S guarantees $f(S) \ge (1 - (1 - 1/k)^{k-1}) \cdot f(S_k^*)$, and the lemma follows. □

Now consider the second step of φ_{IS}, that is, the computation of a signal using ComputeSignal.

Lemma 7. *For every $k \ge 2$, let $S \cup \{n\}$ be any set of k actions. Given the set $S \cup \{n\}$ of actions, ComputeSignal computes a signaling scheme φ such that*

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

The algorithm decides for each action $i \in S \cup \{n\}$ independently whether to recommend this action, stopping the process after the first recommendation. Hence, the probability that an action is recommended consists of two parts, namely, no other action having been recommended beforehand and the coin flip for this action coming up as "recommend." Using the generalized mediant inequality, we can then bound the expected sender utility. The formal proof follows below.

Proof. Given the chosen set S of actions, we consider these actions one by one in nonincreasing order of $g_i(z_i^*)/z_i^*$. ComputeSignal flips an independent coin for each action whether to recommend it. We perform several bounding steps to provide a lower bound on $u_{\mathcal{S}}(\varphi)$. First, we assume that the final "backup signal" for action n in the last line, line 6, has value zero for $\mathcal{S}$. We use $p_\ell = \prod_{\ell'=1}^{\ell-1}(1-z^*_{\ell'})$ to denote the probability to arrive in iteration $\ell > 1$ in the "for" loop. Conditioned on arriving in iteration ℓ, the combined probability of action i_ℓ having state j and issuing a recommendation is $q_{i_\ell,j}\cdot(x^*_{i_\ell,j}/q_{i_\ell,j}) = x^*_{i_\ell,j}$. Thus, conditioned on arriving in iteration ℓ, the expected value for $\mathcal{S}$ from this iteration is $\sum_{j=1}^m x^*_{i_\ell}\xi_{i_\ell,j} = g_{i_\ell}(z^*_{i_\ell})$. Overall,

$$\frac{u_{\mathcal{S}}(\varphi)}{f(S)} \geq \frac{\sum_{\ell=1}^k g_{i_\ell}(z^*_{i_\ell})\cdot p_\ell}{\sum_{\ell=1}^k g_{i_\ell}(z^*_{i_\ell})} = \frac{\sum_{\ell=1}^k u_{i_\ell}\cdot z^*_{i_\ell}\cdot p_\ell}{\sum_{\ell=1}^k u_{i_\ell}\cdot z^*_{i_\ell}}, \tag{4}$$

where we use the notation $u_{i_\ell} = g_{i_\ell}(z^*_{i_\ell})/z^*_{i_\ell}$. Note that if $z=0$, then $g_{i_\ell}(z)=0$. More generally, if there is an action $i_\ell \in S$ with $g_{i_\ell}(z^*_{i_\ell}) = 0$, then we can drop it from consideration and consider the ratio with the $k-1$ remaining actions. Hence, we can assume that $u_{i_\ell} > 0$, for all $1 \leq \ell \leq k$. By scaling the terms, we obtain $u_{i_k} = 1$ without changing the ratio. Note that the last ratio in (4) is a weighted mediant, where the terms u_{i_ℓ}, $1 \leq \ell \leq k$, act as weights for the ratios

$$\frac{z^*_{i_1}}{z^*_{i_1}} > \frac{z^*_{i_2}p_1}{z^*_{i_2}} > \ldots > \frac{z^*_{i_k}p_k}{z^*_{i_k}}.$$

Repeated application of the generalized mediant inequality shows that when $u_{i_1} \geq \ldots \geq u_{i_k} = 1$, the ratio is minimized for $u_{i_1} = \ldots = u_{i_k} = 1$, that is,

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

For the third line, observe that the last function in the second line is symmetric and convex in every variable $z^*_{i_\ell}$. As such, it has a global minimum at $z^*_{i_1} = \ldots = z^*_{i_k} = 1/k$. □

Combining the previous lemmas allows us to bound the approximation ratio. We proceed to show persuasiveness of the scheme.

Lemma 8. *ComputeSignal returns a direct and persuasive signaling scheme for independent instances with k signals.*

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

Proof. Note that ComputeSignal solves LP (2) to optimality. Hence, because of the first constraint of LP (3), we have $\sum_{j=1}^m x_{ij} \leq z_i$ for every $i \in S \cup \{n\}$. We first argue that we can, w.l.o.g., assume that this constraint holds with equality.

Every LP (3) is a parametric linear program. Increasing scalar z increases the right-hand side of the first packing constraint. It is easy to see that $g_i(0) = 0$. Standard sensitivity analysis for parametric linear programs implies that $g_i(z)$ is nondecreasing, piecewise linear, and concave. Hence, an optimal assignment $\mathbf{z}^*$ in (2) results from a water-filling approach, where we raise the entries z^*_i until they sum up to one, while keeping a common slope for all functions g_i for $i \in S \cup \{n\}$. For every $i \in [n-1]$, there exists at most one breakpoint $\hat{z}_i \in [0,1]$ such that the slope of $g_i(z)$ is zero for all $\hat{z}_i \leq z \leq 1$. If no such breakpoint exists, we can set $\hat{z}_i = 1$. Without loss of generality, we assume $0 \leq z^*_i \leq \hat{z}_i$ and $z^*_n \geq 0$ such that $\sum_{i=1}^n z^*_i = 1$. Observe that for every $z_n \in [0,1]$, we can assume the first constraint in LP (3) holds with tightness without violating the second constraint with ϱ_E. As a consequence, we can assume w.l.o.g. for every $i \in [n]$ that in the optimal solution $\mathbf{z}^*$ of (2), the first constraint of every LP (3) is satisfied with equality $\sum_{j=1}^m x^*_{ij} = z^*_i$.

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

For each action $i \in S \cup \{n\}$, ComputeSignal observes the type realization and uses the optimal solution $\boldsymbol{x}^*$ for LP (3) to flip an independent coin that yields the recommendation for action i. First, condition on the event that the scheme returns the signal for action $i_\ell \in S$ in the last "for" loop. We again use $p_\ell = \prod_{\ell'=1}^{\ell-1}(1 - z^*_{i_{\ell'}})$ to denote the probability that the scheme arrives in iteration ℓ. Because of independent coin flips in the "for" loop, the probability that the signal is sent in iteration ℓ is $\sum_{j=1}^m q_{i_\ell,j} \cdot x^*_{i_\ell,j}/q_{i_\ell,j} = z^*_{i_\ell}$, where we assume the equality $z^*_i = \sum_{j=1}^m x^*_{ij}$ as observed above. A signal for action $i_\ell \neq n$ yields a conditional expected utility for $\mathcal{R}$ of

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

where the inequality follows from the second constraint in (3).

Now suppose ComputeSignal signals action n. First, suppose the signal results from the last line of the scheme. Then all coins in other iterations $\ell' \neq \ell$ with $i_{\ell'} \neq n$ have not come up heads, which has probability $p_{-\ell} = \prod_{\ell' \neq \ell}(1 - z^*_{i_{\ell'}})$. In addition, the signal in iteration ℓ with $i_\ell = n$ must not be sent. The receiver obtains an expected utility of

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

Second, assume the signal results from iteration ℓ of the "for" loop; then the expected utility is

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

A signal for action n yields a conditional expected utility for $\mathcal{R}$ of

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

where the inequality follows from the equality $z^*_i = \sum_{j=1}^m x^*_{ij}$ and the second constraint in (3).

Hence, for every recommended action, the expected value for $\mathcal{R}$ is at least ϱ_E. Thus, deviating to any action $i \notin S \cup \{n\}$ is not profitable for $\mathcal{R}$, because the type of action i is independent of the signal, and every action a priori has expected value at most ϱ_E for $\mathcal{R}$.

We condition on the case that ComputeSignal sends a signal for action $i_\ell \neq n$ in the "for" loop. The expected value of action $i_{\ell'}$ with $\ell' > \ell$ is at most ϱ_E, because the type of action $i_{\ell'}$ has not been observed. For $\ell' < \ell$, the scheme decided not to send a signal using an independent coin flip in iteration ℓ'. The overall value of action $i_{\ell'}$ for $\mathcal{R}$ is most ϱ_E, the value of a signal is at least ϱ_E, so a nonsignal for action $i_{\ell'}$ has value at most ϱ_E for $\mathcal{R}$. Similar arguments show that conditioned on a signal for action n, every other action has expected value at most ϱ_E. This proves that the resulting scheme is persuasive. □

In terms of running time, GreedyActions solves (2) an $O(nk)$ number of times. ComputeSignal solves (2) only once, and then computes at most $k - 1$ independent coin flips. Clearly, both algorithms can be implemented to run in time polynomial in the representation of the input. This concludes the proof of Theorem 4.

4.2. Improved Approximation and Tightness

In this section, we improve the approximation ratio of the scheme from the previous section. It is easy to see that Lemma 7 is tight—there are cases[3] in which the sender utility of any persuasive scheme for action set $S \cup \{n\}$ can indeed recover at most a fraction of $1 - (1 - 1/k)^k$ of $f(S)$.

Instead, we replace the standard greedy algorithm for submodular maximization by a more elaborate procedure to carefully choose a subset of actions. In this section, we describe an FPTAS to compute, for every given constant $\varepsilon > 0$, a set $S \subseteq [n-1]$ of $k-1$ actions such that $f(S) \geq (1-\varepsilon) \cdot f(S^*)$ for the set $S^* \subseteq [n-1]$ of $k-1$ actions that maximizes f.

Our approach in the algorithm described below is to use a discretized version $\hat{f}$ of function f. In $\hat{f}(S)$, we restrict the possible values for z_i, for every action $i \in [n]$, to $z_i \in \{0, \tau, 2\tau, 3\tau, \ldots, 1\}$, where $\tau = 1/\lceil k/\delta \rceil$. This restriction decreases the optimal value by at most a factor of δ, that is, $\hat{f}(S) \geq (1-\delta)f(S)$ for every subset $S \subseteq [n-1]$. We then construct a knapsack-style FPTAS to find, for any constant $\delta > 0$, a subset S such that $\hat{f}(S) \geq (1-\delta)\hat{f}(S^*) \geq (1-\delta)^2 f(S^*)$ in polynomial time, where $S^* \subseteq [n-1]$ is the set of $k-1$ actions maximizing f. Using $\delta = \varepsilon/2$ then yields $\hat{f}(S) \geq (1-\varepsilon)f(S^*)$. By submodularity, $f(S^*) \geq (k-1)/k \cdot f(S_k^*)$, and, hence, $f(S^*) \geq (k-1)/k \cdot f(K) \geq (k-1)/k \cdot u_{\mathcal{S}}(\varphi^*)$.

The following proposition summarizes the main insight from this section.

Proposition 4. *For every $k \geq 2$ and every constant $\varepsilon > 0$, there is a polynomial-time algorithm to compute a subset S of $k-1$ actions such that*

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

Combining the algorithm for selection of S with ComputeSignal, we obtain a signaling scheme that we term the improved independent scheme. Proposition 4 together with Lemmas 7 and 8 implies Theorem 5.

Let us now describe the algorithm and the guarantee in Proposition 4 in more detail. We first apply a discretization, for which we need to solve LP (3) a total of at most $O(nk/\varepsilon)$ times. The subsequent FPTAS procedure needs $O(n^2k^6/\varepsilon^3)$ time which, arguably, seems rather high. Our goal here was to simplify the exposition and the analysis of the FPTAS as much as possible. It is an interesting direction for future work to improve the running time in terms of the dependence on k and ε.

4.2.1. Discretization. For approximating f, we consider approximating the function $\hat{f}$. The definition of $\hat{f}$ is the same as for f in (2), where we add a discretization constraint that $z_i \in \{0, \tau, 2\tau, 3\tau, \ldots, \tau(1/\tau - 1), 1\}$ with $\tau = 1/\lceil k/\delta \rceil$.

Lemma 9. *Consider the subset $S^* \subseteq [n-1]$ that maximizes $f(S^*)$. It holds that $\hat{f}(S^*) \geq (1-\delta)f(S^*)$.*

Proof. Because (2) is a packing problem, we can assume w.l.o.g. that $|S^*| = k-1$. We denote by $\boldsymbol{z}^*$ the optimal solution for $f(S^*)$ in (2). For $z_i' = (1-\delta)z_i^*$, concavity and monotonicity of g_i implies $g_i(z_i') \geq (1-\delta)g_i(z_i^*)$ for every $i \in S^* \cup \{n\}$. Observe that $\sum_{i \in S^* \cup \{n\}} z_i' \leq (1-\delta)$ because $\boldsymbol{z}^*$ is a feasible solution. We round z_i' up to the next multiple of τ, that is, $\hat{z}_i = \tau \cdot \lceil z_i'/\tau \rceil$. Then

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

Now $\hat{\boldsymbol{z}}$ is a feasible solution for the optimization problem of $\hat{f}(S)$, so

$$\begin{aligned}\hat{f}(S) &\geq \sum_{i \in S^* \cup \{n\}} g_i(\hat{z}_i) \geq \sum_{i \in S^* \cup \{n\}} g_i(z_i') \\ &\geq (1-\delta) \sum_{i \in S^* \cup \{n\}} g_i(z_i^*) = (1-\delta)f(S^*).\end{aligned}$$

We rephrase the optimization problem of $\hat{f}(S)$ as having $1/\tau$ many *particles* that can be assigned to the actions $S \cup \{n\}$. The ℓ th particle assigned to action i has marginal profit $m_i^\ell = g_i(\ell\tau) - g_i((\ell-1)\tau)$. For every action i, the marginal profit of the ℓth assigned particle is $m_i^\ell \geq 0$ and $m_i^{\ell+1} \leq m_i^\ell$, for all $\ell \geq 1$. Clearly, the optimal solution for $\hat{f}(S)$ can be computed by a simple greedy algorithm: Assign the $1/\tau$ particles to actions $S \cup \{n\}$ in nonincreasing order of marginal profit. Consider the set $\hat{S}^*$ that optimizes $\hat{f}(S)$ over all subsets S of size at most $k-1$. Let m^* be the profit of the last particle assigned by the greedy algorithm to any action in $\hat{S}^* \cup \{n\}$.

Our main idea in the FPTAS is to guess m^*. Put differently, we run the algorithm discussed in the following for all marginal profits from all particles of all functions g_i, $i \in [n]$. Because only $1/\tau$ particles must be considered for any action, we have at most $n/\tau = O(nk/\delta)$ calls to the algorithm. For the rest of this section, we outline our approach for a given marginal profit value m.

Given a value m, consider an action i. We denote by $\ell_i(m)$ the largest number of a particle with marginal profit strictly larger than m. Suppose that $i \in \hat{S}^*$ and $m = m^*$. Then, in $\hat{f}(\hat{S}^*)$, we will assign at least $z_i \geq \tau\ell_i(m)$ to action i. With foresight, we use the notation $w_i^r(m) = \tau \cdot \ell_i(m)$ and $p_i^r(m) = g_i(\tau\ell_i(m))$. Suppose i has particles $\ell_i(m)+1, \ell_i(m)+2, \ldots, \ell_i(m)+t_i(m)$ with marginal profit m; then $\hat{f}$ assigns a mass of $z_i \in [\tau\ell_i(m), \tau(\ell_i(m)+t_i(m))]$. We use the notation $w_i^o(m) = \tau \cdot t_i(m)$ and $p_i^o(m) = m \cdot \tau \cdot t_i(m) = m \cdot w_i^o(m)$. Otherwise, if g_i has no particle with marginal profit m, then $z_i = \tau\ell_i(m)$, and we set $w_i^o(m) = p_i^o(m) = 0$. With this notation, we can express $\hat{f}(\hat{S}^*)$ by

$$\begin{aligned}\hat{f}(\hat{S}^*) &= \sum_{i \in \hat{S}^* \cup \{n\}} g_i(\tau\ell_i(m^*)) + m^* \cdot \sum_{i \in \hat{S}^* \cup \{n\}} (z_i - \tau\ell_i(m^*)) \\ &= \sum_{i \in \hat{S}^* \cup \{n\}} p_i^r(m^*) + m^* \cdot \left(1 - \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(m^*)\right).\end{aligned}$$

This means that $\hat{f}$ distributes a total of $1/\tau$ particles to $\hat{S}^*$ such that for all functions g_i, $i \in \hat{S}^* \cup \{n\}$, we exhaust all particles from these functions with marginal profit strictly larger than m^*. The remaining particles achieve a marginal profit of exactly m^*. This implies, in particular, that

$$0 \leq 1 - \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(m^*) \leq \sum_{i \in \hat{S}^* \cup \{n\}} w_i^o(m^*). \tag{5}$$

4.2.2. Knapsack Problem. Consider the following integer optimization problem for a given marginal profit value m. We strive to find a subset S of at most $k-1$ actions such that $1/\tau$ particles can be assigned with a marginal profit of at least m from actions $i \in S \cup \{n\}$ to maximize the resulting total profit:

$$\begin{aligned} h(m) = \text{Max.} &\sum_{i=1}^{n} y_i p_i^r(m) + \min\left(m - m\sum_{i=1}^{n} y_i w_i^r(m), \sum_{i=1}^{n} y_i p_i^o(m)\right), \\ \text{s.t.} &\sum_{i=1}^{n} y_i w_i^r(m) \leq 1, \\ &\sum_{i=1}^{n-1} y_i \leq k-1, \\ &y_n = 1, \\ &y_i \in \{0,1\}. \end{aligned} \tag{6}$$

For given m, we denote the optimal solution for $h(m)$ by $\boldsymbol{y}^*$ and the action set optimizing $h(m)$ by $S_m^* = \{i \mid y_i^* = 1, i \neq n\}$.

Lemma 10. *For every marginal profit m, the following hold:*

a. *If $h(m)$ is feasible, then $h(m) \leq \hat{f}(\hat{S}^*)$.*
b. *If $m = m^*$, then $h(m^*)$ is feasible and $h(m^*) = \hat{f}(\hat{S}^*)$.*
c. *If $h(m)$ is infeasible, then $m \neq m^*$.*

The definition of h above in (6) describes the greedy algorithm for distributing particles with marginal profit rate at least m. Together with (5), we can show part a using this observation. The remaining properties, b and c, can then quickly be verified.

Proof. In h, we sum the value from each action $i \in S_m^*$ for the required assignment of particles to arrive at marginal profit m, and then use the remaining particles to generate additional value at a rate of m. Consider any marginal profit m and a feasible solution $\boldsymbol{y}$ for $h(m)$ with action set $S = \{i \mid y_i = 1, i \neq n\}$. If S satisfies (5), then $h(m) = \hat{f}(S)$, because h correctly captures the greedy algorithm to assign particles to g_i in nonincreasing order of marginal profit. However, there might be values m and solutions $\boldsymbol{y}$, such that for the corresponding set $S \cup \{n\}$ of actions it is impossible to find a total of $1/\tau$ particles with marginal profit at least m. Clearly, if this happens, then

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

This implies, in particular, that either $m \neq m^*$ or $S \neq \hat{S}^*$, because otherwise we would violate (5). Moreover, in h the set S only yields a value of

$$\sum_{i \in S \cup \{n\}} p_i^r(m) + \min\left(m - m \sum_{i \in S \cup \{n\}} w_i^r(m), \sum_{i \in S \cup \{n\}} p_i^o(m)\right)$$
$$= \sum_{i \in S \cup \{n\}} p_i^r(m) + \sum_{i \in S \cup \{n\}} p_i^o(m),$$

that is, it only sums up the value generated by particles with marginal profit at least m. In contrast, in $\hat{f}(S)$, we would continue the greedy algorithm and assign particles beyond the ones with marginal profit at least m. This holds in particular for $S = S_m^*$, so $h(m) \leq \hat{f}(S_m^*)$. Because $\hat{f}(S_m^*) \leq \hat{f}(\hat{S}^*)$, this proves part a.

It is straightforward to verify that for m^* and the optimal set $\hat{S}^*$, the conditions in (5) guarantee that $h(m^*)$ is feasible. Moreover, (5) implies that in the objective function,

$$\min\left(m^* - m^* \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(m^*), \sum_{i \in \hat{S}^* \cup \{n\}} p_i^o(m^*)\right)$$
$$= m^*\left(1 - \sum_{i \in \hat{S}^* \cup \{n\}} w_i^r(m^*)\right).$$

This implies that $h(m^*) = f(\hat{S}^*)$, and part b follows.

If $h(m)$ is infeasible, then for every subset $S \subseteq [n-1]$ with $|S| \leq k-1$ actions, we have

$$\sum_{i \in S \cup \{n\}} w_i^r(m) > 1.$$

Then $m \neq m^*$ because (5) is violated. This proves part c. □

4.2.3. Dynamic Program. As a consequence of Lemma 10, in order to compute an approximation to $\hat{f}(\hat{S}^*)$ we focus on approximating $h(m)$ in (6) for every given value m. For convenience, we use a knapsack terminology. There is an *required item* for action i with size $w_i^r(m)$ and profit $p_i^r(m)$. In addition, there is an *optional item* with size $w_i^o(m)$ and profit $p_i^o(m)$. The constraints in (6) (with the exception of the trivial constraint $y_n = 1$) exactly represent the constraint set of the 1.5-dimensional knapsack problem (Kellerer et al. [35, section 9.7]).

The objective function can be interpreted as follows. Upon packing a required item of action i into the knapsack, we also allow to fill the remaining space in the knapsack with (any fraction of) the optional item of i. Note that all optional items correspond to particles with marginal profit m. Optional items can be removed to free space for required items of other actions. Because required items correspond to particles with marginal profit larger than m, they generate more value per unit of size they occupy in the knapsack. Hence, adding required items (as long as the constraint set allows it) and removing (parts of) optional ones is always desirable.

For every given m, we now describe an FPTAS to approximate the optimal solution of (6) by $(1-\delta)$ in polynomial time, for every constant $\delta > 0$. The approach resembles the standard dynamic programming approach for the knapsack problem. We assume w.l.o.g. that all required items fit into the knapsack, that is, $w_i^r(m) \leq 1$ for all $i \in [n-1]$, because otherwise we can drop the action from consideration.

Consider $p_{max}(m) = \max\{p_i^r(m), \min(m, p_i^o(m)) \mid i \in [n]\}$, and assume $\kappa = (\delta \cdot p_{max}(m))/2k$. We consider the adjusted profits $\overline{p}_i^r = \lfloor p_i^r(m)/\kappa \rfloor$ and $\overline{p}_i^o = \lfloor p_i^o(m)/\kappa \rfloor$. Our dynamic programming table is given by $A(i, j, \overline{p}^r, \overline{p}^o)$ with the interpretation that for this entry we consider a subset of solutions of the following form: (1) the packed required items are from actions $\{1, \ldots, i, n\}$, (2) we pack the required items of action n and exactly j of the remaining actions, (3) the packed required items have a total adjusted profit of $\overline{p}^r$, and (4) the adjusted profit of optional items corresponding to packed required items sums to $\overline{p}^o$. For each entry $A(i, j, \overline{p}^r, \overline{p}^o)$, we store the minimum total size of required items of any solution that fulfills the conditions of this entry. The number of possible table entries is $O(n \cdot k^5/\delta^2)$, which is a polynomial number in n and k. We initialize all entries with ∞. Then the base cases of the recursion are

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

We fill the table in increasing order of the parameters by setting

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

where we assume the entry is ∞ whenever the arguments become negative. Clearly, this recursion allows us to fill the table in time linear in the size of the table. As in the standard knapsack problem, the recursion simply distinguishes between packing the required item of action i into the knapsack or not.

The rationale behind this approach is as follows. Consider the set of solutions represented by $A(i,j,\overline{p}^r,\overline{p}^o)$. Clearly, when we have $\overline{p}^r$ adjusted profit from packed required items and a potential adjusted profit of $\overline{p}^o$ from optional items, the best solution is one that minimizes the size of packed required items to allow for a maximum portion of optional items to be included into the knapsack.

After completing the table, we consider all entries with $A(i,j,\overline{p}^r,\overline{p}^o) \le 1$, because these entries correspond to a feasible solution. From each of these entries, we pick the one that maximizes the adjusted profit $\kappa \cdot \overline{p}^r + \min(m - m \cdot A(i,j,\overline{p}^r,\overline{p}^o), \kappa \cdot \overline{p}^o)$.

4.2.4. Approximation Ratio. Consider the adjusted profit of the optimal solution S_m^*, which is

$$\begin{aligned} &\sum_{i\in S_m^*\cup\{n\}} \kappa \overline{p}_i^r + \min\left(m - m \cdot \sum_{i\in S_m^*\cup\{n\}} w_i^r, \sum_{i\in S_m^*\cup\{n\}} \kappa \overline{p}_i^o \right) \\ &\ge h(m) - 2k\kappa = h(m) - \delta p_{max}. \end{aligned}$$

If p_{max} is attained for a profit of a required item p_i^r, then consider packing only the required item i. This is a feasible solution because $w_i^r \le 1$. Otherwise, suppose p_{max} is attained for an entry $\min(m, p_i^o(m))$. We use $\min(m, p_i^o(m))$ in the definition of p_{max}, because the optional item is not assumed to fit into the knapsack completely, and $m \cdot 1$ is the profit of a knapsack filled completely with any set of (parts of) optional items. Now suppose we pack only the optional item of i (or parts of it until the knapsack is full). Then pack the required item of i, thereby possibly replacing parts of the optional item. This is a feasible solution because $w_i^r(m) \le 1$. The replacement increases the profit over $\min(m, p_i^o(m))$. Overall, these observations imply $h(m) \ge p_{max}$.

The dynamic program computes a solution S' with the best adjusted profit. The profit of S' is more than the adjusted profit, which is more than the adjusted profit of S_m^*, which is more than $h(m) - \delta p_{max}$. Because $h(m) \ge p_{\max}$, the profit of S' is at least $(1-\delta) \cdot h(m)$.

Because we run the dynamic program for all marginal profits of particles, the best solution S that is found overall has value $f(S) \ge \hat{f}(S) \ge (1-\delta)h(m^*) = (1-\delta)\hat{f}(\hat{S}^*) \ge (1-\delta)^2 f(S^*) = (1-\varepsilon)f(S^*)$ because of Lemmas 9 and 10.

4.3. Beyond ϱ_E-Optimality

Let us briefly observe that our approach does not easily translate to independent instances without ϱ_E-optimality. Consider the following example. There are $n = 2$ actions and $k = 2$ signals. Action 1 has deterministic type $\Theta_1 = \{\theta_{11}\}$ with $(\xi_{11}, \varrho_{11}) = (1,0)$. Action 2 has types $\Theta_2 = \{\theta_{21}, \theta_{22}\}$ with $(\xi_{21}, \varrho_{21}) = (0,1)$ and $(\xi_{22}, \varrho_{22}) = (0,0)$, and $q_{21} = q_{22} = 1/2$. Note $\varrho_E = 1/2$ for action 2.

The optimal scheme φ^* recommends action 1 in state $(\theta_{11}, \theta_{22})$ and action 2 in state $(\theta_{11}, \theta_{21})$. In the former case, $\mathcal{R}$ has conditional expectation of zero for each of the actions, so action 1 is a best response. In the latter case, the recommended action is optimal for $\mathcal{R}$. The expected utility for $\mathcal{S}$ in φ^* is $1/2$.

Instead, suppose we solve LP (2). Because the constraints in (3) require a conditional expectation of ϱ_E for every signal, the optimal solution is $x_{21}^* = x_{22}^* = 1/2$, and thus $g_1(z^*) = g_2(z^*) = 0$. Hence, the optimal value of the LP is zero. Clearly, the optimal scheme does not give rise to a feasible LP solution, and the optimal LP-value does not upper bound the expected utility of φ^* for $\mathcal{S}$.

More fundamentally, any positive value for $\mathcal{S}$ results from $\mathcal{R}$ taking action 1, which in turn must be inherently correlated with the state of action 2. This correlation is not sufficiently reflected in the LP or the algorithms above, which exploit independence conditions. Obtaining a constant-factor approximation for general independent instances in polynomial time is an interesting open problem.

5. Approximation by Restricted Signals

Let OPT_k denote the expected sender utility of the optimal scheme with k signals. We quantify the performance loss against a case when the sender has (at least) n signals available and achieves OPT_n.

5.1. Symmetric Instances

We define the imitation scheme φ_{Imi} for any symmetric instance with n actions and k signals. It first runs an optimal symmetric scheme φ^*_n for n signals. Let i be the action chosen by φ^*_n. If $i \in [k]$, we signal action i; otherwise, we signal any action chosen uniformly at random from $[k]$.

The running time of φ_{Imi} is determined by the running time to implement an optimal symmetric scheme for n signals. In particular, such a scheme can be computed with the Slope-Algorithm, so an efficient probability oracle is sufficient for polynomial running time of φ_{Imi}. We now show that φ_{Imi} provides a tight approximation ratio in terms of OPT_n.

Proposition 5. *The imitation scheme is symmetric, direct, and persuasive in symmetric instances. For every $k \geq 2$, it holds that $u_{\mathcal{S}}(\varphi_{Imi}) \geq k/n \cdot \mathrm{OPT}_n$. There exists a random-order instance such that $\mathrm{OPT}_k \leq k/n \cdot \mathrm{OPT}_n$.*

Proof. We first prove the result for the imitation scheme. The optimal scheme φ^*_n is symmetric. If φ_{Imi} deviates from the recommendation of φ^*_n, it recommends a uniform random action in $[k]$. Hence, φ_{Imi} is also symmetric.

Conditioned on action i being recommended by φ_{Imi}, the type distribution of action i is $\mathcal{D}_{yes}$ from φ^*_n with probability k/n or $\mathcal{D}_{no}$ from φ^*_n with probability $(n-k)/n$. If an action $i \in [k]$ is not recommended, the type distribution is $\mathcal{D}_{no}$ from φ^*_n, no matter which action $j \in [k]$ is recommended. Let ϱ_{yes} and ϱ_{no} be the expected utilities of $\mathcal{R}$ in $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ from φ^*_n, respectively. In φ_{Imi}, the expected utility for $\mathcal{R}$ for any given action $i \in [k]$ must satisfy

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

Because $\varrho_{no} \leq \varrho_E$, the expected utility for $\mathcal{R}$ when following a recommended action in φ_{Imi} can be bounded by $k/n \cdot \varrho_{yes} + (n-k)/n \cdot \varrho_{no} \geq \varrho_E$. By Lemma 3 we see that φ_{Imi} is persuasive.

The optimal scheme φ^*_n is symmetric and recommends each action with probability $1/n$. With probability k/n, φ_{Imi} recommends the same action as φ^*_n, so $u_{\mathcal{S}}(\varphi_{Imi}) \geq k/n \cdot \mathrm{OPT}_n$.

For the upper bound on OPT_k, we consider an instance from the random-order scenario. There are n types. Type θ_1 has utility pair (1, 1), and all $n-1$ remaining types have utility pair (0, 0). Obviously, $\mathrm{OPT}_n = 1$, the sender gives a signal for the action with type 1. With k signals, there is an optimal scheme that recommends only the first k actions. With probability k/n, type 1 is among those k actions. Otherwise, type 1 cannot be recommended. Hence, $\mathrm{OPT}_k \leq k/n$, which completes the proof. □

5.2. Independent Instances

The first lemma shows that there are independent (and symmetric) instances such that the best approximation ratio is in $O(k/n)$.

Lemma 11. *There exists an i.i.d. instance such that $\mathrm{OPT}_k \leq e/(e-1)\cdot k/n \cdot \mathrm{OPT}_n$.*

Proof. In the distribution for every action, there is a good type θ_1 with utility pair (1, 1) and $q_{\theta_1} = 1/n$, and a bad type θ_0 with utility pair (0, 0) and $q_{\theta_0} = 1 - 1/n$. Clearly, the optimal mechanism is to signal an action with θ_1 whenever it exists (within the first k actions). This yields a ratio of

$$\frac{\mathrm{OPT}_k}{\mathrm{OPT}_n} = \frac{1-\left(1-\frac{1}{n}\right)^k}{1-\left(1-\frac{1}{n}\right)^n} = \frac{k/n - \sum_{i=2}^{k}\binom{k}{i}\left(\frac{-1}{n}\right)^i}{1-\sum_{i=2}^{k}\binom{n}{i}\left(\frac{-1}{n}\right)^i}.$$

This ratio is at most $e/(e-1)\cdot k/n$, for all $k \in \{2,\ldots,n\}$, where $e/(e-1) \approx 1.58$. To see this, observe

$$\mathrm{OPT}_k \geq \frac{k}{k+1}\cdot \mathrm{OPT}_{k+1},$$

because the left-hand side is a lower bound on the sender utility when using φ_{Imi} for k signals based on the optimal i.i.d. scheme for $k+1$ signals. Hence,

$$\frac{\mathrm{OPT}_k}{k} \geq \frac{\mathrm{OPT}_{k+1}}{k+1}.$$

Therefore, for all $k = 2, \ldots, n$, we have $\mathrm{OPT}_k/k \leq \mathrm{OPT}_1/1 = \mathrm{OPT}_1 = 1/n$, that is,

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

The factor is monotone in n and grows to $1/(1-1/e) = e/(e-1)$. □

Note that all symmetric instances satisfy ϱ_E-optimality, because symmetric schemes always guarantee an expected utility of at least ϱ_E conditioned on a recommendation. Hence, the i.i.d. instance used in the proof of Lemma 11 above also shows an upper bound for instances satisfying ϱ_E-optimality.

To provide an asymptotically tight bound of $\Omega(k/n)$ for ϱ_E-optimal instances, we consider the independent-imitation scheme. Similar to the schemes in Sections 4.1 and 4.2 above, it consists of the two steps of (a) choosing a suitable subset of actions and (b) computing a good direct signaling scheme for the chosen subset of actions. In the independent-imitation scheme, we use ActionsReduce (Algorithm 4) for step (a) and ComputeSignal (Algorithm 3) for step (b) as above. Because the main computational step in both algorithms is to solve a single linear program, the scheme can be implemented in polynomial time.

Algorithm 4 (ActionsReduce)

Input: Type sets $\Theta_1, \ldots, \Theta_n$ and distributions $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, s.t. $\sum_j q_{n,j}\varrho_{nj} = \varrho_E$ and $\sum_j q_{n,j}\xi_{nj} = \max_{i\in[n]:\sum_j q_{i,j}\varrho_{ij} = \varrho_E} \sum_j q_{i,j}\xi_{ij}$, parameter $2 \leq k \leq n$

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

Theorem 6. *The independent-imitation scheme is direct and persuasive for independent ϱ_E-optimal instances with k signals. It can be implemented in time polynomial in the input size. For every $k \geq 2$,*

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

Proof. Following (2) we observed that $f(S) \geq u_{\mathcal{S}}(\varphi^*_{S\cup\{n\}})$, so in particular, $f([n-1]) \geq \mathrm{OPT}_n$. For every action $i \in [n-1]$ and every type $j \in \Theta_i$, let z_i^* and x_{ij}^* be the values of the optimal LP solution for $f([n-1])$. It is straightforward to verify that for every subset S, the values $(z_i^*)_{i\in S\cup\{n\}}$ and $(x_{ij}^*)_{i\in S\cup\{n\}, j\in\Theta_i}$ constitute a feasible solution for the LP when optimizing $f(S)$. Because ActionsReduce chooses S to contain the $k-1$ actions with largest $g_i(z_i^*)$,

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

The approximation ratio now follows using Lemma 7. By Lemma 8, the resulting signaling scheme is direct and persuasive. □

Endnotes

[1] This is a standard assumption in bilevel optimization problems. It is mainly used to avoid technicalities such as tiny perturbations to break ties.

[2] Another way to think about symmetric instances is that $\Theta_i = \Theta_j$ for all $i, j \in [n]$.

[3] Consider a set $S \cup \{n\}$ consisting of k i.i.d. actions. Every action $i \in S \cup \{n\}$ has two possible types $\Theta^i = \{\theta_1, \theta_0\}$, where $(\varrho(\theta_1), \xi(\theta_1)) = (1, 1)$, $q_{\theta_1} = 1/k$, and $(\varrho(\theta_0), \xi(\theta_0)) = (0,0)$. Observe that $f(S) = 1$. The best persuasive scheme recommends an action with type θ_1 whenever there is one, which happens only with probability $1 - (1-1/k)^k$.

[4] An extended abstract of this paper is included in the proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms (Gradwohl et al. [29]).

References

[1] Alizamir S, de Véricourt F, Wang S (2020) Warning against recurring risks: An information design approach. *Management Sci.* 66(10): 4612–4629.
[2] Alonso R, Câmara O (2016) Persuading voters. *Amer. Econom. Rev.* 106(11):3590–3605.
[3] Anunrojwong J, Iyer K, Manshadi V (2020) Information design for congested social services: Optimal need-based persuasion. Preprint, submitted May 14, https://arxiv.org/abs/2005.07253.
[4] Arieli I, Babichenko Y (2019) Private Bayesian persuasion. *J. Econom. Theory* 182:185–217.
[5] Au PH, Kawai K (2020) Competitive information disclosure by multiple senders. *Games Econom. Behav.* 119:56–78.
[6] Aumann R, Maschler M (1966) Game theoretic aspects of gradual disarmament. Report 80, U.S. Arms Control and Disarmament Agency, Washington, DC, 1–55.
[7] Aybas Y, Turkel E (2019) Persuasion with limited signal spaces. Preprint, submitted October 29, https://arxiv.org/abs/1910.13547.
[8] Babichenko Y, Barman S (2017) Algorithmic aspects of private Bayesian persuasion. *Proc. Eighth Sympos. Innovations Theoret. Comput. Sci.* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany), 34:1–34:16.
[9] Badanidiyuru A, Bhawalkar K, Xu H (2018) Targeting and signaling in ad auctions. Czumaj A, ed. *Proc. 29th ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 2545–2563.
[10] Bahar G, Smorodinsky R, Tennenholtz M (2016) Economic recommendation systems: One page abstract. Conitzer V, Bergemann D, Chen Y, eds. *Proc. 17th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 757.
[11] Bergemann D, Morris S (2019) Information design: A unified perspective. *J. Econom. Lit.* 57(1):44–95.
[12] Bhaskar U, Cheng Y, Ko YK, Swamy C (2016) Hardness results for signaling in Bayesian zero-sum and network routing games. Conitzer V, Bergemann D, Chen Y, eds. *Proc. 17th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 479–496.
[13] Candogan O (2019) Persuasion in networks: Public signals and *k*-cores. Tardos É, Elkind E, Vohra R, eds. *Proc. 20th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 133–134.
[14] Candogan O (2020) Information design in operations. *Pushing the Boundaries: Frontiers in Impactful OR/OM Research.* Druehl C, Elmaghraby W, eds. TutORials in Operations Research (INFORMS, Catonsville, MD), 176–201.
[15] Candogan O, Drakopoulos K (2020) Optimal signaling of content accuracy: Engagement vs. misinformation. *Oper. Res.* 68(2):497–515.
[16] de Véricourt F, Gurkan H, Wang S (2021) Informing the public about a pandemic. *Management Sci*. 67(10):6350–6357.
[17] Drakopoulos K, Jain S, Randhawa R (2021) Persuading customers to buy early: The value of personalized information provisioning. *Management Sci.* 67(2):828–853.
[18] Dughmi S (2017) Algorithmic information structure design: A survey. *ACM SIGecom Exchanges* 15(2):2–24.
[19] Dughmi S, Xu H (2016) Algorithmic Bayesian persuasion. Wichs D, Mansour Y, eds. *Proc. 48th Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 412–425.
[20] Dughmi S, Xu H (2017) Algorithmic persuasion with no externalities. Daskalakis C, Babaioff M, Moulin H, eds. *Proc. 18th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 351–368.
[21] Dughmi S, Kempe D, Qiang R (2016) Persuasion with limited communication. Conitzer V, Bergemann D, Chen Y, eds. *Proc. 17th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 663–680.
[22] Dughmi S, Niazadeh R, Psomas A, Weinberg SM (2019) Persuasion and incentives through the lens of duality. Caragiannis I, Mirrokni V, Nikolova E, eds. *Proc. 15th Internat. Conf. Web Internet Econom.* (Springer, Cham, Switzerland), 142–155.
[23] Emek Y, Feldman M, Gamzu I, Leme RP, Tennenholtz M (2014) Signaling schemes for revenue maximization. *ACM Trans. Econom. Comput.* 2(2):5.
[24] Forges F (2020) Games with incomplete information: From repetition to cheap talk and persuasion. *Ann. Econom. Stat.* 137:3–30.
[25] Gentzkow M, Kamenica E (2017) Bayesian persuasion with multiple senders and rich signal spaces. *Games Econom. Behav.* 104:411–429.
[26] Gentzkow M, Kamenica E (2017) Competition in persuasion. *Rev. Econom. Stud.* 84(1):300–322.
[27] Goldstein I, Leitner Y (2018) Stress tests and information disclosure. *J. Econom. Theory* 177:34–69.
[28] Gradwohl R, Hahn N, Hoefer M, Smorodinsky R (2022) Reaping the informational surplus in Bayesian persuasion. *Amer. Econom. J. Microeconomics*. Forthcoming. https://www.aeaweb.org/articles?id=10.1257/mic.20200399&&from=f.
[29] Gradwohl R, Hahn N, Hoefer M, Smorodinsky R (2021) Algorithms for persuasion with limited communication. Marx D, ed. *Proc. 32nd ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 637–652.
[30] Gur Y, Macnamara G, Saban D (2020) On the disclosure of promotion value in platforms with learning sellers. Preprint, submitted November 21, https://arxiv.org/abs/1911.09256.
[31] Hahn N, Hoefer M, Smorodinsky R (2020) Prophet inequalities for Bayesian persuasion. Bessiere C, ed. *Proc. 29th Internat. Joint Conf. Artificial Intelligence* (International Joint Conferences on Artificial Intelligence Organization), 175–181.
[32] Hahn N, Hoefer M, Smorodinsky R (2020) The secretary recommendation problem. Biró P, Hartline JD, Ostrovsky M, Procaccia AD, eds. *Proc. 21st ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 189.
[33] Kamenica E (2019) Bayesian persuasion and information design. *Annu. Rev. Econom.* 11(1):249–272.
[34] Kamenica E, Gentzkow M (2011) Bayesian persuasion. *Amer. Econom. Rev.* 101(6):2590–2615.
[35] Kellerer H, Pferschy U, Pisinger D (2013) *Knapsack Problems* (Springer, Berlin).
[36] Kolotilin A (2015) Experimental design to persuade. *Games Econom. Behav.* 90:215–226.
[37] Krause A, Golovin D (2014) Submodular function maximization. Bordeaux L, Hamadi Y, Kohli P, eds. *Tractability: Practical Approaches to Hard Problems* (Cambridge University Press, Cambridge, UK), 71–104.
[38] Kremer I, Mansour Y, Perry M (2014) Implementing the "wisdom of the crowd." *J. Political Econom.* 122(5):988–1012.
[39] Küçükgül C, Özer Ö, Wang S (2022) Engineering social learning: Information design of time-locked sales campaigns for online platforms. *Management Sci.*, ePub ahead of print November 19, https://pubsonline.informs.org/doi/abs/10.1287/mnsc.2021.4151.
[40] Le Treust M, Tomala T (2019) Persuasion with limited communication capacity. *J. Econom. Theory* 184:104940.
[41] Li F, Norman P (2018) On Bayesian persuasion with multiple senders. *Econom. Lett.* 170:66–70.
[42] Lingenbrink D, Iyer K (2018) Signaling in online retail: Efficacy of public signals. Preprint, submitted May 26, http://dx.doi.org/10.2139/ssrn.3179262.

[43] Lingenbrink D, Iyer K (2019) Optimal signaling mechanisms in unobservable queues. *Oper. Res.* 67(5):1397–1416.
[44] Nemhauser G, Wolsey L, Fisher M (1978) An analysis of approximations for maximizing submodular set functions—I. *Math. Programming* 14:265–294.
[45] Papanastasiou Y, Bimpikis K, Savva N (2018) Crowdsourcing exploration. *Management Sci.* 64(4):1727–1746.
[46] Rabinovich Z, Jiang AX, Jain M, Xu H (2015) Information disclosure as a means to security. Weiss G, Yolum P, Bordini RH, Elkind E, eds. *Proc. 14th Conf. Autonomous Agents Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 645–653.
[47] Rubinstein A (2017) Honest signaling in zero-sum games is hard, and lying is even harder. Chatzigiannakis I, Indyk P, Kuhn F, Muscholl A, eds. *Proc. 44th Internat. Colloquium Automata, Languages, Programming* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany), 77:1–77:13.
[48] Wardrop JG (1952) Road paper. Some theoretical aspects of road traffic research. *Proc. Inst. Civil Engineers* 1(3):325–362.
[49] Xu H (2020) On the tractability of public persuasion with no externalities. Chawla S, ed. *Proc. 30th ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 2708–2727.
[50] Xu H, Rabinovich Z, Dughmi S, Tambe M (2015) Exploring information asymmetry in two-stage security games. Bonet B, Koenig S, eds. *Proc. 29th Conf. Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), 1057–1063.
[51] Xu H, Freeman R, Conitzer V, Dughmi S, Tambe M (2016) Signaling in Bayesian Stackelberg games. Jonker CM, Marsella S, Thangarajah J, Tuyls K, eds. *Proc. 15th Conf. Autonomous Agents Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 150–158.