Artificial Intelligence 345 (2025) 104348

Contents lists available at ScienceDirect

Artificial Intelligence

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

Efficient and effective budget-feasible mechanisms for submodular valuations ☆

Kai Han [a,b,*], Haotian Zhang [c], Shuang Cui [c]

[a] Ministry of Education's Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai, PR China
[b] School of Computing and Artificial Intelligence, Shanghai University of Finance and Economics, Shanghai, PR China
[c] School of Computer Science and Technology, Soochow University, Suzhou City, Jiangsu Province, PR China

ARTICLE INFO

Keywords:
Mechanism design
Budget-feasible
Approximation
Complexity

ABSTRACT

We revisit the classical problem of designing Budget-Feasible Mechanisms (BFMs) for submodular valuation functions, which has been extensively studied since the seminal paper of Singer [FOCS'10] due to their wide applications in crowdsourcing and social marketing. We propose TripleEagle, a novel algorithmic framework for designing BFMs, based on which we present several simple yet effective BFMs that achieve better approximation ratios than the state-of-the-art work. Moreover, our BFMs are the first in the literature to achieve linear query complexity under the value oracle model while ensuring obvious strategyproofness, making them more practical than the previous BFMs. We conduct extensive experiments to evaluate the empirical performance of our BFMs, and the experimental results demonstrate the superiorities of our approach in terms of efficiency and effectiveness compared to the state-of-the-art BFMs.

1. Introduction

In the celebrated Influence Maximization (IM) problem (e.g., [30,37,29]), an advertiser needs to select several "seed users" in a social network such that the influence spread is maximized under certain influence propagation models, such as the Independent Cascade model and the Linear Threshold model [37]. Singer [50] further studies how to provide appropriate incentives for individuals to serve as seed users in influence maximization. For example, an online retailer selling concert tickets may wish to promote a concert through word-of-mouth in a social network. The retailer can offer discounts to seed users who are willing to buy a ticket and announce their purchase to their friends in the social network, while the total amount of discounts that the advertiser can offer is limited. In such a case, the retailer needs to design an incentive-compatible mechanism to allocate the discounts under the budget constraint [50].

Another similar problem is designing incentive mechanisms for crowdsourcing, which has been considered by many previous studies such as [5,51,42,58,59,25,34]. In this problem, a crowdsourcing task owner needs to hire workers to perform a task and pay them under a budget, with the goal of maximizing the crowdsourcing revenue. Each worker has a cost associated with participating in the crowdsourcing campaign and may strategically report this cost to maximize their utility, which is the difference between the payment received and their actual cost. The task owner must determine how to pay the workers in a way that ensures no worker can benefit from dishonestly reporting their costs.

☆ This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172384.
* Corresponding author.
E-mail addresses: hankai@mail.shufe.edu.cn (K. Han), 20234027013@stu.suda.edu.cn (H. Zhang), lakers@mail.ustc.edu.cn (S. Cui).

https://doi.org/10.1016/j.artint.2025.104348
Received 30 October 2023; Received in revised form 25 April 2025; Accepted 25 April 2025

Available online 7 May 2025

In addition to the applications described above, many other proposals have also addressed similar mechanism design problems related to AI and multi-agent systems, including:

(1) Incentive mechanisms for federated learning [57]: Federated Learning (FL) [36] is a popular decentralized learning approach in which machine learning models are trained directly on the devices or servers where the data resides. Since the data owners (i.e., clients) typically consume computing and communication resources for local training, researchers are interested in how to incentivize these self-interested clients to contribute their resources for FL model training [55]. In particular, Zhang et al. [57] have studied the problem of designing FL incentive mechanisms based on clients' reputation and reverse auctions, such that the quality of model training can be improved without violating a budget constraint on the payments to the clients.

(2) Data trading for recommender systems [20]: Most machine learning systems, such as recommendation systems and personalized medical treatments, require a large amount of personal data for training and testing. However, sharing personal data may compromise the privacy of data providers. Consequently, the problem of data pricing for privacy compensation in machine learning has garnered significant interest (see the survey in [19]). In particular, [20] propose an approach for trading data in recommendation systems, where a budget-constrained learner must compensate data providers for their privacy loss under the Differential Privacy model.

(3) Incentive mechanisms for collective decision making [41]: Liu et al. [41] indicate that many applications in collective decision making (e.g., committee selection, conducting surveys, public opinion prediction, voting, school choice) require selecting multiple agents to represent the overall population. Since the agents have private costs for serving as representative members and for information collection, the planner needs to compensate the selected agents under a budget constraint while avoiding strategic behavior by the agents.

All the application problems mentioned above belong to the class of *budget-feasible mechanism (BFM) design*, which was initiated by the seminal paper by Singer [49]. Singer [49] proposes a general and fundamental BFM model described as follows: A buyer needs to select a set S of strategic sellers (e.g., workers in crowdsourcing or seed users in influence maximization) with private costs from a ground set $\mathcal{N}$, where $|\mathcal{N}| = n$. The buyer also wants to pay the sellers under a budget B in such a way that the revenue of selecting S (denoted by the function value $f(S)$) is maximized, and the sellers are incentivized to report their private costs truthfully. Singer [49] considers a revenue function $f(\cdot)$ that is *monotone submodular*, which generalizes additive functions and has many applications. For example, the influence spread function in the aforementioned influence maximization application is known to be submodular under several famous influence propagation models [37].

Following Singer [49]'s work, many budget-feasible mechanisms (BFMs) (e.g., [15,34,10,5,21,2,11]) have been proposed. Almost all of the theoretical studies on BFMs have focused on improving the approximation ratios of the proposed mechanisms, as elaborated in Section 1.1.

1.1. Main results of the related work

There exist BFMs for different valuation functions including (monotone or non-monotone) submodular functions, additive, XOS and subadditive functions. We review them separately in the following.

BFMs for monotone submodular valuation functions: Singer [49] proposes a randomized BFM with an approximation ratio of 117.7. Chen et al. [15] improve the work of [49] by proposing a randomized BFM with an approximation ratio of 7.91, and they also propose a lower bound of 2 (resp. $\sqrt{2}+1$) for the approximation ratio of any randomized (resp. deterministic) BFM for submodular valuation functions. Jalaly and Tardos [33,34] further prove that the randomized mechanism in [49] actually can achieve an approximation ratio of 5 by setting different values of input parameters for randomization. It is noted that all the above mentioned BFMs are randomized and incur a super-linear number (at least $\Omega(n^2)$) of value oracle queries. The studies of [49,33,34] also propose BFMs with exponential complexity on value oracle queries, among which the randomized BFM proposed by [33,34] achieves the best ratio of 4. Recently, Balkanski et al. [10] propose the first deterministic BFM with polynomial running time, which achieves a ratio of 4.75 by using $\mathcal{O}(n^2 \log n)$ value oracle queries, and they also show a lower bound of 4.5 for the ratio of their proposed BFM.

Observing the large number of participants in some BFM applications such as crowdsourcing, some studies [5,33,34] have also considered a "large market" model where the additional assumptions of $\frac{c_{max}}{B} \approx 0$ or $\frac{v_{max}}{\mathrm{OPT}} \approx 0$ are made, with c_{max} (resp. v_{max}) denoting the maximum cost (resp. valuation) of any single participant. Under these assumptions, Anari et al. [5] and Jalaly and Tardos [33,34] show that there exist 3-approximation and 2.58-approximation BFMs using $\mathcal{O}(n^2)$ and $\mathcal{O}(n^6)$ value oracle queries, respectively. However, their time complexity still can be too high for large markets, and the additional assumptions mentioned above may not always hold. For example, in the influence maximization problem, the advertiser's budget could be relatively small even though the social network has billion-scale, so $\frac{c_{max}}{B} \approx 0$ may not be true. Therefore, in this work we will consider the original BFM model of [49].

BFMs for non-monotone submodular valuation functions: When the valuation function is non-monotone and submodular, Amanatidis et al. [3,4] propose a randomized BFM with an approximation ratio of 505 leveraging the continuous greedy algorithm in [38], while Balkanski et al. [10] propose a deterministic BFM with an approximation ratio of 64 under $\mathcal{O}(n^2 \log n)$ time complexity. Very recently, Huang et al. [32] propose a randomized BFM with an improved approximation ratio of $(3+\sqrt{5})^2 \approx 27.4$ under $\mathcal{O}(n \log n)$ query complexity. However, this approximation ratio is much worse than that achieved by the aforementioned studies (e.g., [15,33,34,10]) for monotone submodular valuations.

BFMs for other valuation functions and models: There also exist BFMs for additive valuation functions [27,49,5,15], among which Gravin et al. [27] achieve the best-known approximation ratios of 2 and 3 for randomized and deterministic BFMs, respectively. For subadditive and XOS valuation functions, the work of [21,11,10] proposes BFMs using *demand oracle queries*, as it is known that no BFMs with polynomial number of value oracle queries can achieve any ratio better than $n^{0.5-\epsilon}$ for these valuation functions [49].

In particular, for subadditive valuation functions, the work in [21] gives a randomized universally truthful mechanism with an approximation ratio of $\mathcal{O}(\log^2 n)$, and a deterministic truthful approximation mechanism with an approximation ratio of $\mathcal{O}(\log^3 n)$. The work of [11] improves the performance bounds of [21] for both subadditive valuation functions and XOS valuation functions. The work of [12] extends the work of [11] by showing the existence of an $\mathcal{O}(1)$-approximation mechanism for subadditive valuations. The work in [47] studies "beyond worst-case" BFMs under the large market model mentioned before. The work in [41] considers a variant of BFM model where the goal is to maximize the minimum proportion of the overall value of the selected sellers from different groups.

1.2. Limitations of prior art and our contributions

Despite the great progress on BFM design in more than a dozen years as described above, the existing BFMs for submodular valuations still suffer from several major deficiencies. First, the state-of-the-art BFMs for monotone submodular valuations have at least $\Omega(n^2)$ query complexity, which can be very time-consuming for large markets with a lot of participants. Second, from a practical point of view, their algorithmic frameworks for achieving truthfulness still have some serious drawbacks, as explained below.

The BFMs in [49,15,33,34,5,11] have used *sealed-bid auctions* roughly described as follows: the sellers first report their costs and then are sorted according to the non-increasing order of their *densities* (i.e., the ratios of their *marginal gains* to costs), and finally the mechanism selects some winners with the largest densities and ensures truthfulness using Myerson's lemma [44]. However, sealed-bid auctions have been criticized for lacking practicability [10,35,7,40], because it is often hard for the players to verify truthfulness and to trust that the auctioneer will faithfully implement the auction protocol, resulting in strategic behaviors [1,40]. For example, although reporting the true item value is a dominant strategy for all players in a sealed-bid auction (e.g., a second-price auction), this fact is neither obvious nor easy to explain. So the players may still behave strategically because they fail to comprehend that they should follow the dominant strategies [40].

To address the issues of sealed-bid auctions, there is growing interest in designing "simple mechanisms" or "obviously strategyproof mechanisms"[1] (e.g., [46,8,18,40,43]). The seminal work by Milgrom and Segal [43] introduces a new class of auctions known as clock auctions, which run over multiple rounds. In each round of a clock auction, each player is offered a price that is no higher than the price offered to this player in the previous round. A player can choose to either accept or reject the offered price. If a player rejects the price, they must exit the auction permanently; otherwise, they remain active. When the auction concludes, only the active players can be selected as winners (but it is not mandatory to select all active players), and each winner is paid the last price they were offered. Milgrom and Segal [43] indicate that clock auctions satisfy obvious strategyproofness and several other properties (e.g., weak group strategyproofness, transparency, and unconditional winner privacy) not possessed by sealed-bid auctions. This implies that, as long as one follows the format of clock auctions, the truthfulness and individual rationality of players can be automatically guaranteed, so the problem of designing a truthful auction is reduced to a purely algorithmic problem of designing a pricing rule with descending prices.[2] Inspired by [43], Balkanski et al. [10] propose a BFM using clock auction with the following design: they make $\mathcal{O}(\log_2 n)$ guesses on OPT using a "doubling trick" similar to that in online learning theory [48], and create $\mathcal{O}(\log_2 n)$ candidate solutions without violating B by offering each active seller a new price for each guessed OPT, and the final winners are selected from the best candidate solutions. The clock auction BFM in [32] adopts a similar framework as [10], but uses random sampling to handle non-monotone submodular valuation functions.

Although clock auctions are considered to be more practical than sealed-bid auctions [10], we argue that they sometimes still lack practicability, as enquiring a player with too many descending prices may be time-consuming and cause the effect that the player loses patience and quits the auction early. We hereby propose the concept of ***pricing complexity*** of clock auctions: The (worst-case) pricing complexity of a clock auction is the maximum number of prices it offers to any player participating in the auction.

It can be seen from above that the BFMs in [10] have $\mathcal{O}(\log_2 n)$ pricing complexity, which could still be large for impatient sellers. For example, a seller could be inquired for 10 times under the pricing complexity of $\mathcal{O}(\log_2 n)$ with $n = 1024$. In fact, Balkanski et al. [10] raise an open question of "whether there exist even simpler families of budget feasible mechanisms with which one can obtain constant approximations mechanisms", and prove that *posted-price mechanisms* are insufficient for deriving a constant ratio. Note that posted-price mechanisms (e.g., [14]) are perhaps the simplest form of truthful mechanisms; these mechanisms show each player one "take-it-or-leave-it" price, and all players who accept the offered prices must be selected as winners. Therefore, post-price mechanisms have precisely 1 pricing complexity. The main difference between posted-price mechanisms and clock auctions is about how they deal with any seller u who accepts the offered price: it is mandatory to select such a seller u as a winner in posted-pricing mechanisms, while clock auctions may not follow this rule (this implies that clock auctions have the additional power of use "side observations" on the other sellers' behavior to decide whether u should be a winner). However, from a practical perspective, clock auctions with low pricing complexities are almost as simple as posted-pricing mechanisms, because both of them can be efficiently implemented and can be easily understood by the players for ensuring truthfulness.

In this paper, we address all the deficiencies of the existing BFMs mentioned above by presenting TripleEagle, a novel clock auction framework that brings us several simple, fast, effective, and practical BFMs. Specifically, we show that:

[1] Roughly speaking, in an obviously strategyproof auction, the bidders can trivially understand that they cannot benefit by manipulating the auction [10,40]. Obvious strategyproofness implies strategyproofness [40], i.e., playing the game truthfully is the weakly dominant strategy of each player.

[2] Due to this reason, we will not provide proofs on truthfulness and individual rationality. We will focus on designing pricing algorithms under the clock auction framework and analyzing their worst-case approximation guarantees.

- For monotone submodular functions, there exist a randomized BFM and a deterministic BFM with the approximation ratios of $\frac{\varphi}{\varphi-1} \approx 4.08$ and $2+\sqrt{6} \approx 4.45$, respectively, where $\varphi \approx 1.325$ is the real root of $\varphi^3 - \varphi - 1 = 0$; both of these BFMs use at most $\mathcal{O}(n)$ value oracle queries. This improves the best-known ratio of 4.75 in [10] (using $\mathcal{O}(n^2 \log n)$ value oracle queries) for this long-standing problem.
- For non-monotone submodular functions, there exists a randomized BFM with an approximation ratio of $\frac{4\psi}{\psi-1} \approx 11.67$, where $\psi \approx 1.521$ is the real root of $\psi^3 - \psi - 2 = 0$; using at most $\mathcal{O}(n)$ value oracle queries. This improves the best-known ratio of $(3+\sqrt{5})^2 \approx 27.4$ of the randomized BFM proposed in [32] using $(n \log n)$ value oracle queries.
- Under the assumption of [15,10] that each seller's cost is no more than B, our TripleEagle BFMs offer each seller only one price; without this assumption, at most one additional price query is needed in total. Therefore, our BFMs only have $\mathcal{O}(1)$ pricing complexity, which provides a confirmative answer to the open question of Balkanski et al. [10] mentioned above, because TripleEagle is virtually as simple as a posted-pricing mechanism.
- We conduct extensive experiments using the applications of influence maximization and crowdsourcing. The experimental results show that, compared to the state-of-the-art BFMs proposed in [33,34,10,32], our BFMs are faster in orders of magnitude, while achieving significantly better objective function values of the winning sellers.

Recall that the clock auction in [10] calculates prices offered to the sellers by blindly guessing OPT. In contrast, our TripleEagle framework adopts a novel idea of leveraging the simple yet fundamental *law of supply and demand* in economics [24] and the concept of reserve price in auction theory [45] to design clock auctions. Specifically, TripleEagle processes the sellers in an arbitrary order and calculates the price offered to a new "supplier" (i.e., an unprocessed seller) based on the "current supply" (i.e., the valuation of already processed sellers who accepted the offered prices). Therefore, the offered prices in TripleEagle tend to decrease when there are more sellers. Moreover, the pricing rule in TripleEagle guarantees that no seller is offered a price higher than a reserve price. When the clock auction terminates, TripleEagle returns a set of sellers with the maximum ratios of marginal gains to prices under the budget B. Due to this novel design, TripleEagle outperforms the clock auction in [10] in terms of several metrics including approximation ratio, query complexity and pricing complexity.

As far as we know, our work is the first attempt on reducing both the time complexity and pricing complexity of BFMs. We analogize a low pricing complexity to the "Triple Eagle" hole score in golf,[3] and believe that our ideas have the potential to be applied to other problems that can be addressed by clock auctions (e.g., FCC spectrum auctions).

A preliminary version of this paper was published in NeurIPS-2023 [31]. The main differences between this version and the conference paper are as follows:

- In the conference version, we have provided a randomized BFM for monotone submodular valuations with the approximation ratio of $\frac{\sqrt{13}+5}{2} \approx 4.3$. In this version, we propose a revised mechanism with a new pricing rule, and we prove that it has an improved approximation ratio of 4.08 through a more involved analysis. Moreover, we show that the analysis that leads to the ratio of 4.08 is tight given our choice of parameters, and that no choice of parameter can give a ratio better than 3.784.
- In the conference version, we have provided a randomized BFM for non-monotone submodular valuations with the approximation ratio of 12. In this version, we improve this ratio to 11.67 by introducing a revised BFM mechanism, and we also revise the performance analysis accordingly.
- Due to the significant changes of our algorithms compared to the conference version, we have updated our experimental results accordingly.

The rest of our paper is organized as follows. We provide the formal problem definitions in Section 2. For monotone submodular valuations, a randomized BFM and a deterministic BFM are introduced in Section 3 and Section 4, respectively. We then extend our BFMs to handle non-monotone submodular valuations in Section 5. We provide the performance evaluation in Section 6 before concluding our paper in Section 7. To maintain the fluency of our paper, we defer most of the proofs of our lemmas/theorems to the Appendix.

2. Preliminaries

Following [49], we assume that each seller u in a ground set $\mathcal{N}$ has a private cost $c(u) \geq 0$ and there is a non-negative submodular valuation function $f(\cdot) : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ satisfying $f(X) \geq 0$ for all $X \subseteq \mathcal{N}$ and

$$\text{for all } X, Y \subseteq \mathcal{N} : f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y). \tag{1}$$

We say that $f(\cdot)$ is monotone if $f(X) \leq f(Y)$ for all $X \subseteq Y \subseteq \mathcal{N}$, otherwise it is non-monotone. For simplicity, we call $f(X \mid Y) \triangleq f(X \cup Y) - f(Y)$ as the marginal gain of X with respect to Y for all $X, Y \subseteq \mathcal{N}$.

We will consider budget-feasible mechanisms in the form of clock auctions that run over multiple rounds. Initially, all sellers in the set $\mathcal{N}$ are active. In each round, each active seller u is offered a price that is no higher than the price offered to u in the previous

[3] E.g., a "hole-in-one" on a par-five, see [56] or https://en.wikipedia.org/wiki/Par_(score).

Algorithm 1: TripleEagleRan(α, β).

1 $A \leftarrow$ LSDPricing($\emptyset, \mathcal{N} \setminus \{v^*\}, \alpha, \beta$);
2 **if** $f(A) \geq f(v^*)$ **then**
3 $\quad A \leftarrow$ LSDPricing($A, \{v^*\}, \alpha, \beta$);
4 $\quad S \leftarrow$ the largest suffix of A satisfying $\sum_{u \in S} p(u) \leq B$;
5 **else**
6 $\quad$ Sample a random number Z from the uniform distribution $\mathcal{U}[0,1]$;
7 $\quad$ **if** $Z \leq \frac{\alpha}{1+\beta+\alpha}$ **then**
8 $\quad\quad p(v^*) \leftarrow B$; $S \leftarrow \{v^*\}$
9 $\quad$ **else**
10 $\quad\quad$ Set $p(v^*) = B - \sum_{u \in A} p(u)$ and show v^* the price $p(v^*)$;
11 $\quad\quad$ **if** v^* *accepts* $p(v^*)$ **then** $T \leftarrow A \cup \{v^*\}$;
12 $\quad\quad$ **else** $T \leftarrow A$;
13 $\quad\quad S \leftarrow T$;
14 **return** S

Algorithm 2: LSDPricing(A, C, α, β).

1 **foreach** $u \in C$ **do**
2 $\quad p(u) \leftarrow \min\{\frac{B \cdot f(u|A)}{\beta f(A) + \alpha f(v^*)}, B\}$; Show u the price $p(u)$;
3 $\quad$ **if** u *accepts* $p(u)$ **then** $A \leftarrow A \cup \{u\}$;
4 **return** A

round. The seller u can either reject the offered price and exit the clock auction permanently or accept the offered price and remain active. We use $p(u)$ to denote the last price offered to u at any time and $p(X)$ (for any $X \subseteq \mathcal{N}$) as shorthand for $\sum_{u \in X} p(u)$. The prices offered to the sellers in any round are computed using only public information, such as the history of prices offered. The clock auction can be terminated at the end of any round, and at that time only the still active sellers can be selected as winners. Suppose the set of winners is S. Each winner $u \in S$ is paid the last offered price $p(u)$ and receives a utility of $p(u) - c(u)$. Moreover, it is required that $p(S) \leq B$ to ensure budget feasibility. The approximation ratio of S is defined as $\mathrm{OPT}/f(S)$, where $\mathrm{OPT} = f(O)$ and O is an optimal solution to the problem $\max\{f(X) : X \subseteq \mathcal{N} \wedge c(X) \leq B\}$ (we use $c(X)$ as shorthand for $\sum_{u \in X} c(u)$).

We will also consider randomized clock auctions: such an auction is a deterministic clock auction for every realization of all its internal random choices, and it clearly satisfies all the properties of deterministic clock auctions ex post (rather than in expectation). For the simplicity of description, we adopt the assumption made in [10,15] that $c(v) \leq B$ for all $v \in \mathcal{N}$, and we will show how to remove this assumption later. In our mechanisms, if the elements in a set X are sequentially inserted into X, listed as $\{u_1, \cdots, u_q\}$ according to the order that they are inserted, then we call the subset $Y = \{u_s, \cdots, u_t\} \subseteq X$(for all $s, t : 1 \leq s \leq t \leq q$) as a *regular subset* of A, and also call Y a *suffix* (resp. *prefix*) of X if $t = q$ (resp. $s = 1$). Throughout the paper, we use v^* to denote a seller in $\mathcal{N}$ who has the maximum valuation, i.e., $v^* \in \arg\max_{u \in \mathcal{N}} f(u)$.

To give an example, let us recall the crowdsourcing application mentioned in Section 1. Suppose that each worker in the crowdsourcing campaign owns an image and the crowdsourcing task is to collect a set S of images that can cover the features in a feature set $\mathcal{T}$. For this crowdsourcing task, Singla et al. [52] have used the objective function $f(\cdot)$ proposed by [22], defined as $f(S) = \sum_{t \in \mathcal{T}} \omega_t \cdot [1 - \Pi_{s \in S}(1 - cover_s(t))]$, where ω_t is a weight for feature t and $cover_s(t) \in [0,1]$ is the probability that image s covers feature t. One can also use other objective functions to evaluate the image set S such as the V-ROUGE function proposed by [54]. It is known that all these evaluation functions are monotone submodular [52,22,54]. The workers may want some rewards for providing their images, and the crowdsourcing task owner could be budget-limited on paying the rewards. It can be seen that this crowdsourcing problem is an instance of the BFM problem defined above.

3. Randomized BFM for monotone submodular valuations

In this section, we consider a monotone submodular valuation function $f(\cdot)$ and provide a randomized BFM using the TripleEagle framework (we will explain this framework shortly), as shown by Algorithm 1 (i.e., the TripleEagleRan algorithm). Algorithm 1 takes as input two parameters α, β that will be explained shortly. In Line 1, Algorithm 1 calls the LSDPricing procedure (i.e., Algorithm 2) to process all sellers in $\mathcal{N}$ except v^*; each of these sellers is offered one price by Algorithm 2 and then Algorithm 2 returns a set A containing the sellers who accept their prices. After that, if $f(A) \geq f(v^*)$, then Algorithm 1 calls Algorithm 2 again to process v^* and finally returns a suffix of A respecting the budget constraint (Lines 3-4), because these sellers have the largest ratios of marginal gain to payment according to the pricing rule. Otherwise, Algorithm 1 makes a random decision of returning either v^* or T (Lines 6-13), where T is a superset of A and is got by trying to include v^* by offering v^* a "best effort" price of $B - p(A)$. In the case that $f(A) < f(v^*)$ after Line 1 of Algorithm 1 is executed, we must have $\sum_{u \in A} p(u) < B$ when $\alpha \geq 1$, because otherwise we have $f(A) \geq p(A) \cdot \frac{\beta f(\emptyset) + \alpha f(v^*)}{B} \geq \alpha f(v^*) \geq f(v^*)$ due to Lemma 1, contradicting $f(A) < f(v^*)$.

It can be seen that Algorithm 2 offers each seller u a price $\frac{B f(u|A)}{\beta f(A) + \alpha f(v^*)}$. This pricing rule is inspired by the fundamental *Law of Supply and Demand (LSD)* [24] and the important concept of *reserve price* in economics, as intuitively explained in the following.

First, note that A contains all the processed sellers who accept the offered prices, so $\beta f(A)$ can be regarded as the current (weighted) “supply” in the market, thus the price offered to a new “supplier” (i.e., an unprocessed user u) should decrease with the increment of $f(A)$. Besides, the marginal gain $f(u \mid A)$ represents how much value the user u can contribute given the current supply $f(A)$, so the price offered to u should increase with $f(u \mid A)$. Second, the factor $\alpha f(v^*)$ guarantees that no user can be paid a price larger than B/α, which can be regarded as a “reserve price” of the buyer. Without this reserve price, the mechanism could perform worse due to the overpayment to some sellers. Therefore, in all our mechanisms, we implicitly assume $\alpha \geq 1$ to ensure that the reserve price is no more than B, unless otherwise stated.

Recall that we have mentioned about our TripleEagle framework. This framework is characterized by the aforementioned novel idea of leveraging LSD and reserve price to design BFMs. All the mechanisms proposed in this paper adopt this framework.

3.1. Performance analysis of TripleEagleRan

It is evident that Algorithm 1 provides each seller only one price and incurs $2n$ value oracle queries (n queries for finding v^*). So we only analyze its approximation ratio. Let us consider the set A at the moment that Algorithm 1 finishes. For convenience, we use A_u to denote the set of elements already in A at the moment right before u (for all $u \in \mathcal{N}$) is processed. We first introduce Lemmas 1-2, which can be proved based on submodularity and the observation that each seller who accepts the offered price must have a sufficiently large density (and vice versa) due to the pricing rule of Algorithm 2.

Lemma 1. *For any regular subset $X = \{u_s, \cdots, u_t\}$ of A returned by Algorithm 2, we have $f(X) \geq \sum_{v \in X} f(v \mid A_v) \geq p(X) \cdot \frac{\beta f(A_{u_s}) + \alpha f(v^*)}{B}$.*

Lemma 2. *After Line 3 of Algorithm 1 is executed, we have $f(O) \leq (1+\beta) f(A) + \alpha f(v^*)$.*

With Lemmas 1-2, we can bound the approximation ratio of Algorithm 1 if S is generated by Line 4, as shown by Lemma 3 and Lemma 4. Intuitively, if $p(A) \leq B$, then we have $f(A) \geq f(v^*)$ due to Line 3 and hence we can use Lemma 2 to get $f(O) \leq (1+\beta+\alpha) f(S)$ directly; if $p(A) > B$, then $p(S)$ is also sufficiently large, which implies $f(S)$ is large, because S is a suffix of A and hence contains the sellers in A with the largest ratios of marginal gains to prices due to the pricing rule of Algorithm 2.

Lemma 3. *Consider the set A after Line 3 of Algorithm 1 is executed. Suppose that $p(A) > B$ and $f(A \setminus H) = \sigma f(v^*)$, where $H \subseteq A$ is the smallest suffix of A satisfying $p(H) > B$ and $\sigma \geq 0$, then we have*

$$f(S) \geq \begin{cases} (\sigma\beta + \alpha - 1)(1 + \frac{\beta}{\sigma\beta+\alpha}) f(v^*), & \text{if } \beta \in (0,1] \text{ or } \sigma\beta + \alpha \leq \frac{\beta}{\beta-1}. \\ (\sigma\beta+\alpha) f(v^*), & \text{otherwise.} \end{cases} \tag{2}$$

Lemma 4. *For any $\beta \in (0,1]$ and $\alpha > 2$, the set S generated by Line 4 of Algorithm 1 satisfies: (1) When $\alpha \leq \frac{\beta^2+\beta}{\alpha-\beta-1}$, we have*

$$f(O) \leq \max\left(1+\beta+\alpha, 2+\beta+\frac{1}{\beta}, 2+\beta+\frac{2\alpha+\beta}{(\alpha-1)(\alpha+\beta)}\right) \cdot f(S)$$

and (2) When $\alpha \geq \frac{\beta^2+\beta}{\alpha-\beta-1}$, we have $f(O) \leq \max\left(1+\beta+\alpha, 2+\beta+\frac{1}{\beta}\right) \cdot f(S)$.

Proof. Due to Line 2 of Algorithm 1, we must have $f(v^*) \leq f(A)$ when Line 4 of Algorithm 1 is executed. Therefore, if $p(A) \leq B$, then we have $S = A$ and hence

$$f(O) \leq (1+\beta) f(A) + \alpha f(v^*) \leq (1+\beta+\alpha) f(S) \tag{3}$$

due to Lemma 2. In the following, we consider the case of $p(A) > B$. Let H be the smallest suffix of A satisfying $p(H) > B$, so $H \setminus S$ contains exactly one element (denoted by w) which is added into A before the elements in S. Using Lemma 2 and submodularity, we get

$$\begin{aligned} f(O) &\leq (1+\beta) f(A) + \alpha f(v^*) \\ &\leq (1+\beta) f(S \cup \{w\}) + (1+\beta) f(A \setminus H) + \alpha f(v^*) \\ &\leq (1+\beta) f(S) + (1+\beta) f(A \setminus H) + (1+\beta+\alpha) f(v^*) \end{aligned} \tag{4}$$

Suppose that $f(A \setminus H) = \sigma f(v^*)$ for certain $\sigma \geq 0$, and let $x = \sigma\beta + \alpha$. Combining Eqn. (4) with Lemma 3, we get

$$\begin{aligned} f(O) &\leq (1+\beta) f(S) + (\sigma\beta + \sigma + 1 + \beta + \alpha) f(v^*) \\ &= (1+\beta) f(S) + \left(\frac{\beta+1}{\beta} x + 1 + \beta - \frac{\alpha}{\beta}\right) f(v^*) \end{aligned}$$

$$\leq (1+\beta)f(S) + \frac{\frac{\beta+1}{\beta}x + 1 + \beta - \frac{\alpha}{\beta}}{x - \frac{\beta}{x} + \beta - 1} f(S)$$

$$= \left(2 + \beta + \frac{1}{\beta} + \frac{1 + \frac{1-\alpha}{\beta} + \frac{\beta+1}{x}}{x - \frac{\beta}{x} + \beta - 1} \right) f(S) \tag{5}$$

Note that $x = \sigma\beta + \alpha \geq \alpha > 2$. We consider the following two cases:

1. $\alpha \leq \frac{\beta^2+\beta}{\alpha-\beta-1}$ and $x \in [\alpha, \frac{\beta^2+\beta}{\alpha-\beta-1}]$: In this case, we have $1 + \frac{1-\alpha}{\beta} + \frac{\beta+1}{x} \geq 0$, and the right-hand side of Eqn (5) is maximized when $x = \alpha$. This gives us

$$f(O) \leq \left(2 + \beta + \frac{2\alpha + \beta}{(\alpha-1)(\alpha+\beta)} \right) f(S) \tag{6}$$

2. $\alpha \geq \frac{\beta^2+\beta}{\alpha-\beta-1}$ or $x > \frac{\beta^2+\beta}{\alpha-\beta-1}$: In this case, we have $1 + \frac{1-\alpha}{\beta} + \frac{\beta+1}{x} \leq 0$ and hence we get

$$f(O) \leq \left(2 + \beta + \frac{1}{\beta} \right) f(S) \tag{7}$$

Combining Eqn. (3), Eqn. (6) and Eqn. (7), the lemma follows. □

Next, we bound the approximation ratio of Algorithm 1 when S is generated by Lines 6-13 of Algorithm 1 (using randomization), as shown by Lemma 5:

Lemma 5. *For any $\alpha > 2$, if Algorithm 1 returns S that is generated by Lines 6-13, then we have*

$$f(O) \leq (1 + \beta + \alpha) \cdot \mathbb{E}[f(S)]$$

Proof. In this case, we must have $f(A) < f(v^*)$ after Line 1 of Algorithm 1 is executed. This implies $p(A) < B$ because otherwise we have $f(A) \geq \alpha f(v^*)$ due to Lemma 1, contradicting $f(A) < f(v^*)$ when $\alpha > 1$. We prove that the set T generated by Lines 10-13 of Algorithm 1 must satisfy $f(O) \leq (1+\beta)f(T) + \alpha f(v^*)$ according to the following discussion:

1. $v^* \notin O$: In this case, we can use similar reasoning as Lemma 2 to get

$$f(O) \leq (1+\beta)f(A) + \alpha f(v^*) \leq (1+\beta)f(T) + \alpha f(v^*)$$

where the second inequality is due to $A \subseteq T$ according to Lines 11-12.

2. $v^* \in O$ and v^* accepts the price in Line 11: In this case, we have $T = A \cup \{v^*\}$ and hence can also use similar reasoning as Lemma 2 to get

$$f(O) - f(T) \leq \sum\nolimits_{u \in O \setminus T} f(u \mid A) \leq \beta f(A) + \alpha f(v^*) \leq \beta f(T) + \alpha f(v^*),$$

which yields $f(O) \leq (1+\beta)f(T) + \alpha f(v^*)$.

3. $v^* \in O$ and v^* rejects the price in Line 11: In this case, we must have $T = A$, $c(v^*) > B - p(A)$ and hence

$$c(O \setminus \{v^*\}) \leq p(A) \leq \frac{B \cdot f(A)}{\beta f(\emptyset) + \alpha f(v^*)} \leq \frac{Bf(A)}{\alpha f(v^*)} \leq \frac{B}{\alpha} \tag{8}$$

where the second inequality is due to Lemma 1. Note that all the sellers in $O \setminus \{v^*\}$ have been processed by Line 1 and each seller $u \in O \setminus \{v^*\} \setminus A$ must reject the offered price and hence satisfy $c(u) > p(u)$. So we can use this and Eqn. (8) to get

$$\begin{aligned} f(O \setminus \{v^*\}) - f(A) &\leq \sum\nolimits_{u \in O \setminus \{v^*\} \setminus A} f(u \mid A_u) \\ &= \sum\nolimits_{u \in O \setminus \{v^*\} \setminus A} p(u) \cdot \frac{\beta f(A_u) + \alpha f(v^*)}{B} \\ &\leq \sum\nolimits_{u \in O \setminus \{v^*\} \setminus A} c(u) \cdot \frac{\beta f(A) + \alpha f(v^*)}{B} \\ &\leq \frac{1}{\alpha}(\beta f(A) + \alpha f(v^*)) \leq \frac{\beta}{\alpha} f(A) + f(v^*) \end{aligned} \tag{9}$$

Therefore, when $\alpha > 2$, we have

$$\begin{aligned} f(O) &\leq f(O \setminus \{v^*\}) + f(v^*) \leq (1 + \frac{\beta}{\alpha}) f(A) + 2f(v^*) \\ &\leq (1+\beta)f(T) + \alpha f(v^*) \end{aligned} \tag{10}$$

According to Lines 6-13, Algorithm 1 returns T or $\{v^*\}$ with probability of $\frac{1+\beta}{1+\beta+\alpha}$ or $\frac{\alpha}{1+\beta+\alpha}$, respectively. So we get

$$\begin{aligned}\mathbb{E}[f(S)] &= \Pr[S=T]\cdot f(T) + \Pr[S=\{v^*\}]\cdot f(v^*)\\ &= \frac{(1+\beta)f(T)}{1+\beta+\alpha} + \frac{\alpha f(v^*)}{1+\beta+\alpha} \geq \frac{f(O)}{1+\beta+\alpha}\end{aligned} \tag{11}$$

which completes the proof. □

Combining Lemmas 4-5, we immediately get:

Theorem 1. *Let φ be the real number satisfying $\varphi^3 - \varphi - 1 = 0$. Specifically,*

$$\varphi = \frac{\sqrt[3]{1+\sqrt{\frac{23}{27}}} + \sqrt[3]{1-\sqrt{\frac{23}{27}}}}{\sqrt[3]{2}} \approx 1.325. \tag{12}$$

With $\alpha = 1+\varphi \approx 2.325$ and $\beta = \frac{1}{\varphi} \approx 0.755$, the TripleEagleRan *mechanism can return a set S satisfying $f(O) \leq \frac{\varphi}{\varphi-1}\cdot\mathbb{E}[f(S)] \approx 4.08\cdot\mathbb{E}[f(S)]$ by offering one price to each seller in $\mathcal{N}$ and incurring at most $\mathcal{O}(n)$ value oracle queries.*

Proof. Recall that Algorithm 1 provides each seller only one price and incurs $2n$ value oracle queries (n queries for finding v^*). It can be verified that, with $\alpha = 1+\varphi$ and $\beta = \frac{1}{\varphi}$, we have $\alpha = \frac{\beta^2+\beta}{\alpha-\beta-1}$ and

$$1+\beta+\alpha = 2+\beta+\frac{1}{\beta} = 2+\beta+\frac{2\alpha+\beta}{(\alpha-1)(\alpha+\beta)} = \frac{\varphi}{\varphi-1} \tag{13}$$

Combining this with Lemma 4 and Lemma 5 finishes the proof. □

Remark. As mentioned in Sec. 2, we have adopted the assumption made in [10,15] that $c(v) \leq B$ (for all $v \in \mathcal{N}$) for the simplicity of description. This assumption can be easily removed according to the following discussion. Note that each buyer is guaranteed to be offered a price no more than B in TripleEagleRan. Therefore, before running TripleEagleRan, we only need to identify a valid $v^* = \arg\max_{v:\, v\in\mathcal{N}\wedge c(v)\leq B} f(v)$. This can be done by first sorting all the sellers according to the non-increasing order of their values, and then offer B to them one by one until seeing the first seller accepting B and then that seller is clearly a valid v^*. After that, we can delete all the sellers rejecting B and then run TripleEagleRan. Clearly, using such a method, the complexity of our mechanism on the number of value oracle queries is still $\mathcal{O}(n)$, while only one additional price query (for v^*) is incurred. This method can also be used for all our mechanisms described in the sequel.

3.2. Lower bounds

As can be seen from Sec. 1.1, the 4.08-ratio of TripleEagleRan proved in Theorem 1 is currently the best-known ratio for polynomial-time BFMs with monotone submodular valuations. Since this ratio depends on the input parameters α and β, it is interesting to study the lower bounds of TripleEagleRan under different values of α and β, even under the case of $\alpha < 1$. In this section, we first prove that the 4.08-ratio is tight given the values of α and β in Theorem 1, shown by Theorem 2:

Theorem 2. *With $\alpha = 1+\varphi$ and $\beta = \frac{1}{\varphi}$ (φ is defined in Theorem 1), there exists a BFM problem instance for which the* TripleEagleRan *algorithm cannot achieve an approximation ratio better than $\frac{\varphi}{\varphi-1} - \epsilon \approx 4.08 - \epsilon$ for any $\epsilon \in (0,1)$.*

Moreover, we show that the 4.08-ratio proven in Theorem 1 is nearly tight for all non-negative values of α and β, because there exists a lower bound of 3.784 for TripleEagleRan, as proven by Theorem 3.

Theorem 3. *For any non-negative values of α and β and any $\epsilon > 0$, there exists a problem instance for which the* TripleEagleRan *algorithm cannot achieve an approximation ratio better than $3.784 - \epsilon$.*

4. Deterministic BFM for monotone submodular valuations

If we directly use the LSDPricing algorithm to process all the sellers in $\mathcal{N}$, it is possible that the valuation of the sellers who accept the offered prices is no more than $f(v^*)$, while v^* rejects its price and hence cannot be selected as a winner, resulting in a poor approximation ratio (especially when $f(v^*)$ is large). To address this issue, we have used randomization in TripleEagleRan. In this section, we introduce TripleEagleDet (i.e., Algorithm 3), a deterministic BFM that circumvents the aforementioned issue using an idea of "two-phase pricing", as elaborated in the following.

In the first pricing phase (Lines 2-6), Algorithm 3 constructs a set K by using the pricing rule in Line 3 to processes the sellers in $\mathcal{N}\setminus\{v^*\}$; this pricing rule is a relaxation of the pricing rule in the LSDPricing algorithm, because the valuation of the already

Algorithm 3: TripleEagleDet(α).

1 $K \leftarrow \emptyset; C \leftarrow \mathcal{N}$;
2 **foreach** $u \in \mathcal{N} \setminus \{v^*\}$ **do**
3 $\quad p(u) \leftarrow \frac{Bf(u|K)}{\alpha f(v^*)}$; Show u the price $p(u)$;
4 $\quad$ **if** *u accepts $p(u)$* **then** $K \leftarrow K \cup \{u\}$;
5 $\quad C \leftarrow C \setminus \{u\}$;
6 $\quad$ **if** $f(K) \geq f(v^*)$ **then break**;
7 **if** $f(K) < f(v^*)$ **then**
8 $\quad p(v^*) \leftarrow B$; $S \leftarrow \{v^*\}$;
9 **else**
10 $\quad A \leftarrow$ LSDPricing$(K, C, \alpha, 1)$;
11 $\quad S \leftarrow$ the largest suffix of A satisfying $\sum_{u \in S} p(u) \leq B$;
12 **return** S

processed sellers is not used to lower the price for the next seller to be processed. According to submodularity, it can be easily seen that Line 3 guarantees $p(u) \leq B$ when $\alpha \geq 1$.

The first pricing phase stops as soon as $f(K) \geq f(v^*)$ (Line 6). If $f(K)$ is still less than $f(v^*)$ when all the sellers in $\mathcal{N} \setminus \{v^*\}$ have been processed in the first pricing phase, then Algorithm 3 sets the winner set $S = \{v^*\}$ (Line 8). In the case that the first pricing phase generates a set K satisfying $f(K) \geq f(v^*)$, Algorithm 3 enters the second pricing phase in Line 10, where we use LSDPricing (with $\beta = 1$) to process all the sellers not processed in the first phase (including v^*). Intuitively, since $f(K) \geq f(v^*)$ and K is a prefix of A in the second pricing phase (Line 10), we can regard K as a substitute of v^* and hence no longer need to worry about the issue explained in the beginning of this section.

We analyze the approximation ratio of Algorithm 3 by considering two cases. In the first case, we have $f(K) < f(v^*)$ or $p(A) \leq B$, which implies that $S = \{v^*\}$ or $S = A$ due to Line 8 and Line 11. In this case, we can directly use the two-phase pricing rules to bound the total valuation of the sellers in O who rejected the offered prices and hence get Lemma 6:

Lemma 6. *When Algorithm 3 finishes, if $f(K) < f(v^*)$ or $p(A) \leq B$, then we have $f(O) \leq (2 + \alpha) f(S)$.*

In the second case (i.e., $f(K) \geq f(v^*)$ and $p(A) > B$), we must have that S is generated by Line 11 and $S \subset A$. In this case, bounding the performance of Algorithm 3 is more complicated and is given by Lemma 7:

Lemma 7. *Suppose that $\alpha \in (2, \frac{\sqrt{17}+1}{2})$. When Algorithm 3 finishes, if $f(K) \geq f(v^*)$ and $p(A) > B$, then we have*

$$f(O) \leq \max\left(3 + \frac{3\alpha + 4}{\alpha^2 + 2\alpha}, 2 + \frac{\alpha(2+\alpha)}{\alpha^2 + \alpha - 4}, 2 + \frac{\alpha(4+\alpha)}{\alpha^2 + \alpha - 2}\right) \cdot f(S) \tag{14}$$

We roughly explain the idea for proving Lemma 7 as follows. Consider the set A generated by Line 10 (recall that K is prefix of A). If the smallest suffix H of A satisfying $p(H) > B$ does not intersect K, then we can use similar reasoning as Lemma 4 to prove $f(O) \leq \left(3 + \frac{3\alpha+4}{\alpha^2+2\alpha}\right) f(S)$. Under the other case (i.e., $H \cap K \neq \emptyset$), H must have a prefix generated using the relaxed pricing rule of Line 3, which possibly degrades the valuation of H and hence S (note that S is a suffix of H). Fortunately, since $f(K)$ is at most $2f(v^*)$ due to Line 6, we can prove that the approximation ratio of $f(S)$ is only slightly weaker than that in the case of $H \cap K = \emptyset$, i.e., $f(O) \leq \max\left(2 + \frac{\alpha(2+\alpha)}{\alpha^2+\alpha-4}, 2 + \frac{\alpha(4+\alpha)}{\alpha^2+\alpha-2}\right) \cdot f(S)$. Combining Lemmas 6-7, we can immediately get the performance bounds of TripleEagleDet, as shown by Theorem 4, which reveals that TripleEagleDet has a better approximation ratio than the deterministic BFM with a provable ratio of 4.75 in [10]. Note that Balkanski et al. [10] show that their BFM cannot achieve a ratio better than 4.5 even if their analysis for the 4.75-ratio is not tight.

Theorem 4. *With $\alpha = \sqrt{6} \approx 2.45$, the* TripleEagleDet *mechanism can return a set S satisfying $f(O) \leq (2 + \sqrt{6}) f(S) \approx 4.45 \cdot f(S)$ by offering one price to each seller in $\mathcal{N}$ and incurring $\mathcal{O}(n)$ value oracle queries.*

5. BFM for non-monotone submodular valuations

In this section, we extend our TripleEagleRan algorithm to handle a non-monotone submodular valuation function $f(\cdot)$, as shown by the TripleEagleNm algorithm (i.e., Algorithm 4).

The TripleEagleNm algorithm is similar in spirit to the TripleEagleRan algorithm except that it maintains two sets of candidate winners: A_1 and A_2. In Lines 1-5 of Algorithm 4, each seller $u \in \mathcal{N} \setminus \{v^*\}$ is greedily routed to $A_j (j \in \{1, 2\})$ such that $f(u \mid A_j)$ is maximized, and then u is offered a price generated using the same rule as Line 2 of Algorithm 2. After all the sellers in $\mathcal{N} \setminus \{v^*\}$ are processed, Lines 11-18 of Algorithm 4 adopt the operations similar to those in Lines 6-13 of Algorithm 1, except that v^* is offered a "best-effort" price of $B - p(A_q)$ after it is greedily routed to A_q by Line 15. We show the performance bounds of Algorithm 4 in Theorem 5. The overall idea for proving Theorem 5 is similar to that for proving Theorem 1, with the main difference that

Algorithm 4: TripleEagleNm(α, β).

1 $A_1 \leftarrow \emptyset; A_2 \leftarrow \emptyset$;
2 **foreach** $u \in \mathcal{N} \setminus \{v^*\}$ **do**
3     $j \leftarrow \arg\max_{i\in\{1,2\}} f(u \mid A_i)$;
4     $p(u) \leftarrow \frac{B\cdot f(u|A_j)}{\beta f(A_j)+\alpha f(v^*)}$; Show u the price $p(u)$;
5   **if** *u accepts* $p(u)$ **then** $A_j \leftarrow A_j \cup \{u\}$;
6 **if** $\max\{f(A_1), f(A_2)\} \geq f(v^*)$ **then**
7   Let $u = v^*$ and process u using Lines 3-5;
8     $A^* \leftarrow \arg\max_{X\in\{A_1,A_2\}} f(X)$;
9     $S \leftarrow$ the largest suffix of A^* satisfying $\sum_{u\in S} p(u) \leq B$;
10 **else**
11   Sample a random number Z from the uniform distribution $\mathcal{U}[0,1]$;
12   **if** $Z \leq \frac{\alpha}{2+\alpha+\beta}$ **then**
13         $p(v^*) \leftarrow B;\ S \leftarrow \{v^*\}$;
14   **else**
15         $q \leftarrow \arg\max_{i\in\{1,2\}} f(v^* \mid A_i)$;
16   Set $p(v^*) = B - p(A_q)$ and show v^* the price $p(v^*)$;
17   **if** $f(v^* \mid A_q) \geq 0$ *and* v^* *accepts* $p(v^*)$ **then** $A_q \leftarrow A_q \cup \{v^*\}$;
18         $T \leftarrow \arg\max_{X\in\{A_1,A_2\}} f(X); S \leftarrow T$;
19 **return** S

we need to handle the two disjoint sets A_1 and A_2, because we use $f(A_1 \cup O) + f(A_2 \cup O)$ as an upper bound of $f(O)$ in the performance analysis when $f(\cdot)$ is non-monotone. We can use this upper bound of $f(O)$ because we have $f(A_1 \cup O) + f(A_2 \cup O) \geq f(A_1 \cup A_2 \cup O) + f((A_1 \cup O) \cap (A_2 \cup O)) \geq f(O)$, where the first inequality is due to the submodularity of $f(\cdot)$ and the second inequality is due to the non-negativity of $f(\cdot)$ and $A_1 \cap A_2 = \emptyset$.

Theorem 5. *Let ψ be the real number satisfying $\psi^3 - \psi - 2 = 0$. Specifically,*

$$\psi = \sqrt[3]{1 + \sqrt{26/27}} + \sqrt[3]{1 - \sqrt{26/27}} \approx 1.521. \tag{15}$$

With $\alpha = 1 + \psi \approx 2.521$ and $\beta = \frac{2}{\psi} \approx 1.315$, the TripleEagleRan *algorithm can return a set S satisfying $f(O) \leq \frac{4\psi}{\psi-1} \cdot \mathbb{E}[f(S)] \approx 11.67 \cdot \mathbb{E}[f(S)]$ by offering one price to each seller in $\mathcal{N}$ and incurring at most $\mathcal{O}(n)$ value oracle queries.*

6. Performance evaluation

We use two representative applications in social computing (i.e. influence maximization and crowdsourcing) to evaluate the performance of our mechanisms, as described below:

6.1. Influence maximization

We use the influence maximization application mentioned in Sec. 1 to compare the following BFMs: (1) Our TripleEagleRan (TER) and TripleEagleDet (TED) algorithms, (2) The Iterative-Pruning (IP) algorithm in [10] with a ratio of 4.75, and (3) The Random-TM (RTM) algorithm in [34] with a ratio of 5. All the algorithms are implemented using C++ and are run on a Linux server with Intel Xeon Gold 6126 @ 2.60GHz CPU and 128GB memory. Each implemented randomized algorithm is independently executed 50 times, and the average result is reported. We use four real social network datasets: (1) Flixster [6] with 28,843 nodes and 272,786 edges; (2) Epinions [39] with 75,879 nodes and 508,837 edges; (3) Slashdot [39] with 82,168 nodes 948,464 edges; and (4) Email [39] with 265,214 nodes and 420,045 edges. We adopt the well-known *Independence Cascade* (IC) model [37] for the influence spread function $f(\cdot)$, and follow the "Weighted Cascade model" proposed by [37] to set the activation probability $p_{u,v}$ of each edge (u, v) to $1/N_{in}(v)$, where $N_{in}(v)$ denotes the set of in-neighbors of v. This weighted cascade model is widely adopted in previous work such as [53,17,26,37]. The cost $c(u)$ of each node is generated uniformly at random from the interval $[0, 1]$. Since evaluating $f(\cdot)$ is known to be #P-hard [16], we adopt the approach in [13] to generate one million random *reverse-reachable sets* for approximately evaluating $f(\cdot)$. It is known that the influence spread function through such an approximation is still monotone and submodular [13].

In Fig. 1 and Fig. 2, we compare the implemented BFMs on the utility, the number of oracle queries, and the running time. The experimental results reveal that, for all three datasets, our BFMs outperform the baselines on utility (i.e., the valuation of returned solution). Specifically, TER and TED have similar utilities that outperform RTM (resp. IP) with the performance gain ranging from 66%-127% (resp. 7%-19%). Moreover, TER and TED incur much fewer value oracle queries and much less running time than RTM (resp. IP) in more than one (resp. three to four) orders of magnitude. It can also be seen that, although TER and RTM are both randomized mechanisms, RTM incurs large variations while TER is almost deterministic in Fig. 1. This can be explained by the fact that RTM always returns a random solution by its design, while the randomization of TER is only triggered in some special cases (see Sec. 3).

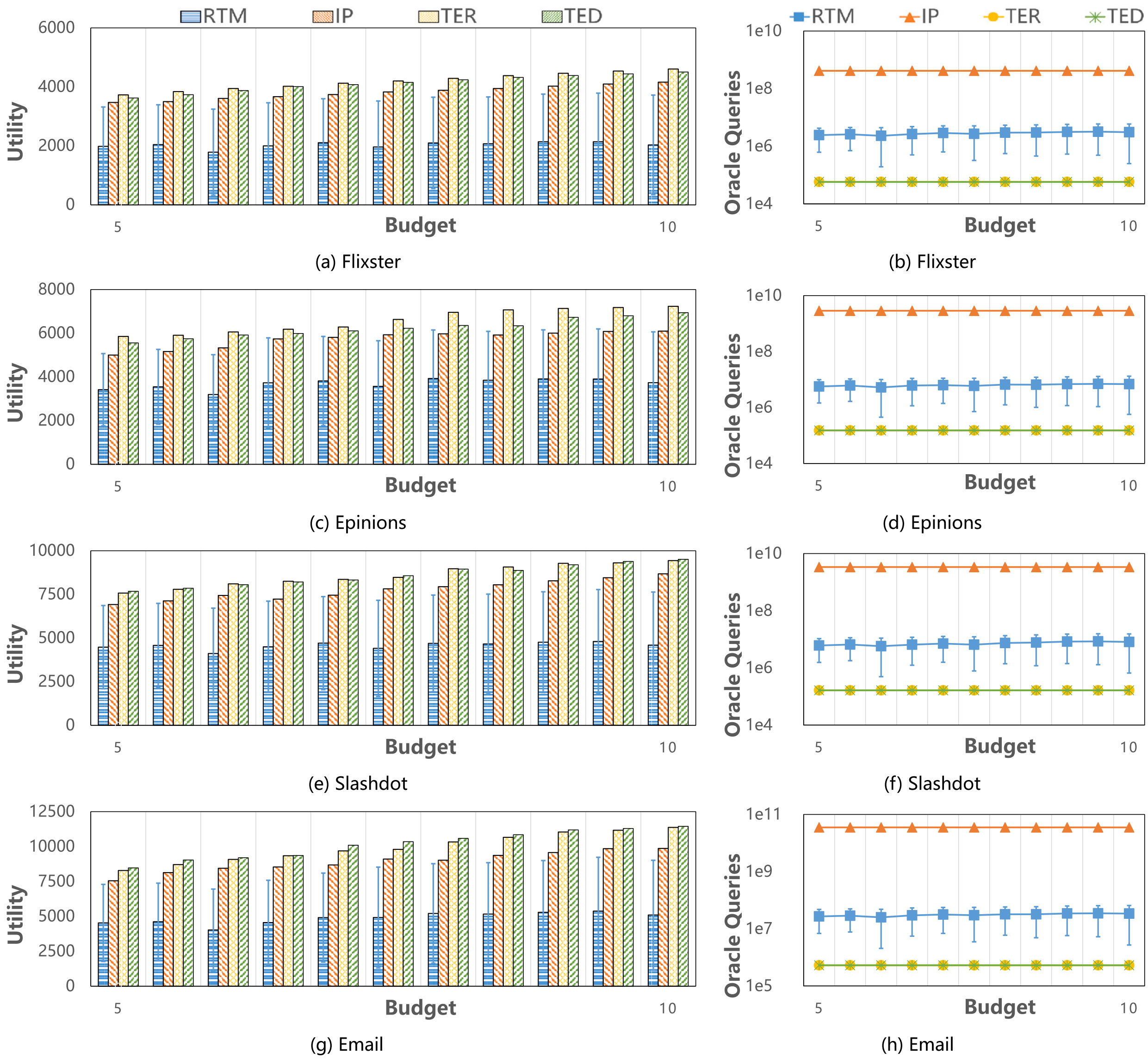

Fig. 1. Comparing the BFMs (w.r.t. utility and number of oracle queries) for the influence maximization application.

6.2. Crowdsourcing

In this section, we conduct experiments using a crowdsourcing application similar to that in [32,28,9,23], where each worker (seller) in $\mathcal{N}$ owns an image, and a buyer with a budget B needs to crowdsource a set S of representative images from the workers. Following [32,28,9,23], we use the CIFAR-10 dataset [32,28,9,23] containing ten thousands 32×32 color images, where each image is associated with a label denoting its category such as "Airplane", and the valuation (utility) of S is calculated as

$$f(S) = \sum\nolimits_{u \in \mathcal{N}} \max_{v \in S} s_{u,v} - \frac{1}{|\mathcal{N}|} \sum\nolimits_{u \in S} \sum\nolimits_{v \in S} s_{u,v}, \tag{16}$$

where $s_{u,v}$ denotes the similarity between any two images u and v and is measured by the cosine similarity of the 3,072-dimensional pixel vectors of image u and image v. As indicated by [9,23], the intuition is to choose a representative subset of images such that at least one image in this subset is similar to each image in the full collection (cast by the first term of Eqn. (16)), but the subset itself is diverse (cast by the second term of Eqn. (16)). It is known that such a valuation function $f(\cdot)$ is non-monotone and submodular [9,23]. We assume that each worker u has a private cost $c(u)$ for providing its image, and set $c(u)$ to be in proportional to the standard deviation of its pixel intensities. Following the same setting as that in [32], the costs of all images are normalized with the average value of 0.1, and the images with labels in {Airplane, Automobile, Bird} are considered. The other experimental settings are the same with those in Fig. 1.

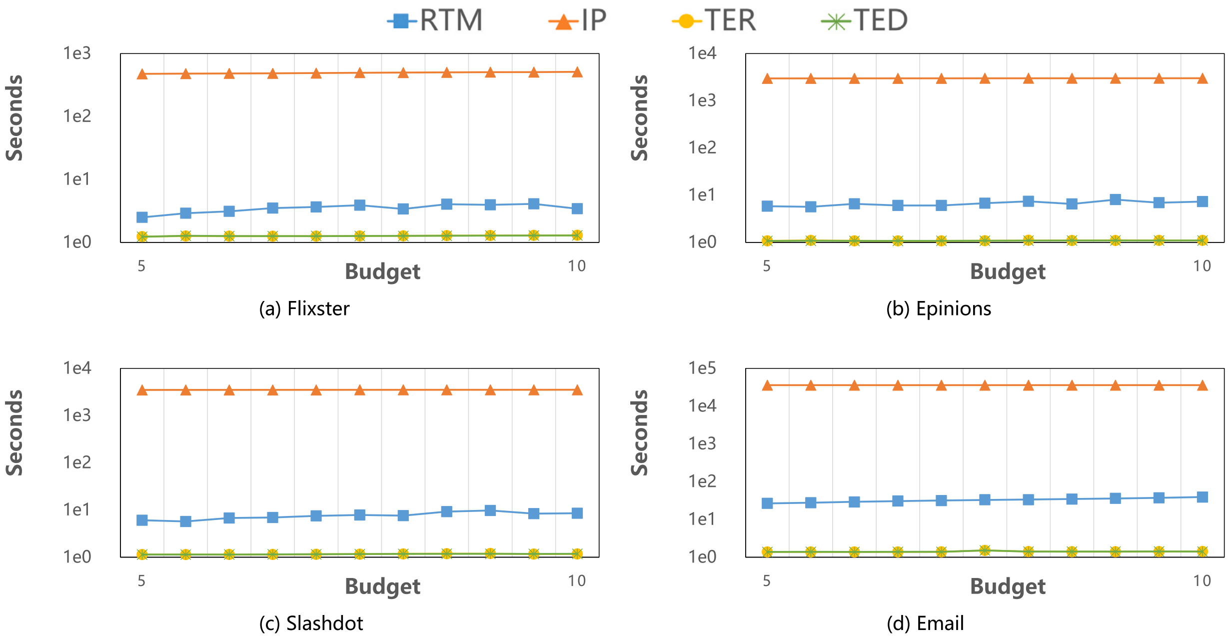

Fig. 2. Comparing the BFMs (w.r.t. running time) for the influence maximization application.

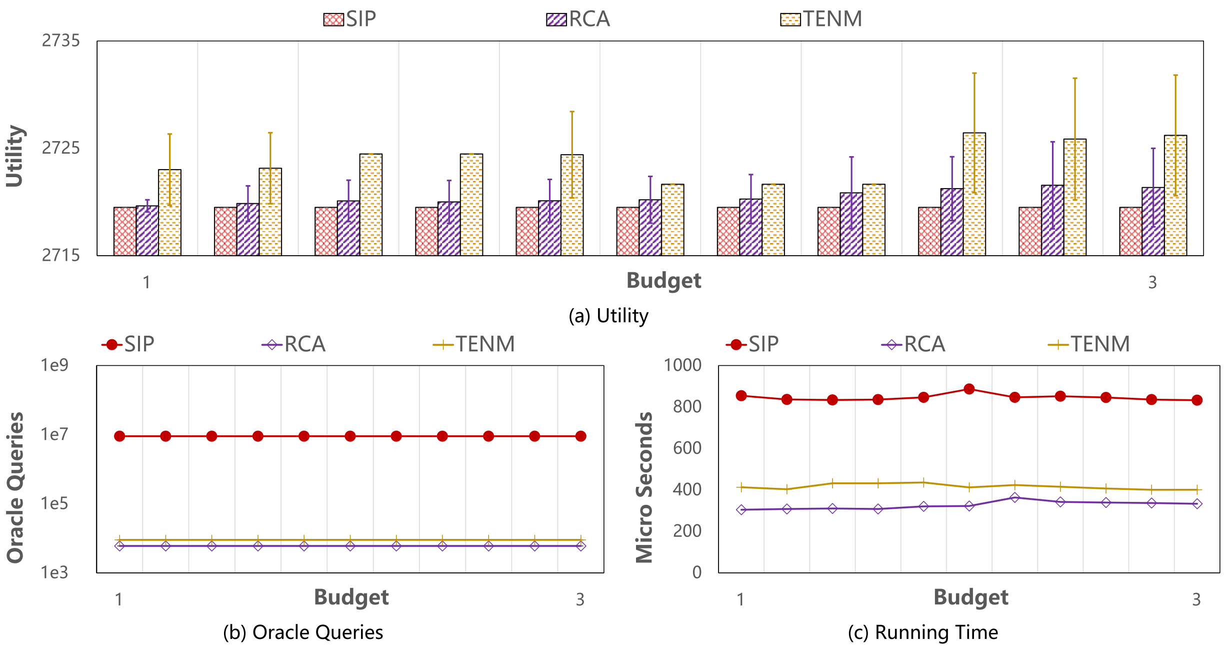

Fig. 3. Comparing the BFMs for the crowdsourcing application.

In Fig. 3, we compare our TripleEagleNm (TENM) algorithm with two state-of-the-art BFMs for non-monotone submodular valuations: (1) the Simultaneous-Iterative-Pruning (SIP) algorithm in [10] with an approximation ratio of 64; and (2) the RandomClockAuction (RCA) algorithm in [32] with an approximation ratio of $(3+\sqrt{5})^2$. The experimental results in Fig. 3 show that, our TENM algorithm consistently outperforms the other two baselines on utility, while incurring about three orders of magnitude fewer value oracle queries than the SIP algorithm. From Fig. 3, it can be seen that TENM performs deterministically in some cases, which can be explained by the fact that the random decision of TENM is only triggered when Lines 11-18 of Algorithm 4 are executed. It is also noted that RCA performs well on the number of value oracle queries and running time; this can be explained by the fact that RCA has a nearly-linear query complexity of $\mathcal{O}(n \log n)$ and hence is also efficient for this problem instance. However, RCA has much poorer performance on utility compared to TENM. These experimental results demonstrate the superiorities of our approach once again for non-monotone submodular valuations.

7. Conclusion

We have proposed TripleEagle, a novel algorithmic framework for designing budget feasible mechanisms with submodular valuations, based on which we have presented several simple, fast, effective and practical BFMs that achieve obvious strategyproofness, low complexity, and better approximation ratios than the-state-of-the-art studies. The efficiency and effectiveness of our approach has been strongly corroborated by our experiments on influence maximization and crowdsourcing.

CRediT authorship contribution statement

Kai Han: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Haotian Zhang:** Software. **Shuang Cui:** Software.

Declaration of competing interest

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

Appendix A. Proof of Lemma 1

Proof. It is evident that $f(X) \geq \sum_{v\in X} f(v \mid A_v) = f(A_{u_s} \cup X) - f(A_{u_s})$ due to submodularity. Besides, for each $v \in X$, we have $f(A_v) \geq f(A_{u_s})$ and $c(v) \leq p(v)$ (because v accepts the offered price), so we get

$$c(v) \leq p(v) = \frac{B f(v \mid A_v)}{\beta f(A_v) + \alpha f(v^*)} \leq \frac{B f(v \mid A_v)}{\beta f(A_{u_s}) + \alpha f(v^*)}$$

and hence

$$\sum_{v\in X} f(v \mid A_v) \geq \sum_{u\in X} p(v) \cdot \frac{\beta f(A_{u_s}) + \alpha f(v^*)}{B} = p(X) \cdot \frac{\beta f(A_{u_s}) + \alpha f(v^*)}{B}$$

which completes the proof. □

Appendix B. Proof of Lemma 2

Proof. After Line 3 is executed, all the elements in $\mathcal{N}$ must have been processed by the LSDPricing algorithm. As each element in $O \setminus A$ rejects the offered price, we have

$$\text{for all } u \in O \setminus A : c(u) > p(u) = \frac{B f(u \mid A_u)}{\beta f(A_u) + \alpha f(v^*)} \geq \frac{B f(u \mid A_u)}{\beta f(A) + \alpha f(v^*)} \tag{B.1}$$

Using this and submodularity, we get

$$\begin{aligned} f(O) - f(A) &\leq \sum_{u\in O\setminus A} f(u \mid A) \leq \sum_{u\in O\setminus A} f(u \mid A_u) \\ &\leq \sum_{u\in O\setminus A} c(u) \cdot \frac{\beta f(A) + \alpha f(v^*)}{B} \leq \beta f(A) + \alpha f(v^*) \end{aligned} \tag{B.2}$$

The lemma then follows by re-arranging the above inequality. □

Appendix C. Proof of Lemma 3

Proof. Note that $H \setminus S$ contains exactly one element (denoted by w) which is added into A before the elements in S. Suppose that $p(w) = \varrho B$ (and hence $p(S) > (1-\varrho)B$) for certain $\varrho \in [0, 1]$. According to Lemma 1, we have

$$f(v^*) \geq f(w \mid A \setminus H) \geq \varrho B \cdot \frac{\beta f(A \setminus H) + \alpha f(v^*)}{B} \geq \varrho(\sigma\beta + \alpha) f(v^*) \tag{C.1}$$

and hence

$$\begin{aligned} f(S) &\geq (1-\varrho)B \cdot \frac{\beta f(A \setminus H) + \beta f(w \mid A \setminus H) + \alpha f(v^*)}{B} \\ &\geq (1-\varrho)(1+\varrho\beta)(\sigma\beta + \alpha) f(v^*) \end{aligned}$$

$$= (1 + \varrho\beta - \varrho - \varrho^2\beta)(\sigma\beta + \alpha)f(v^*) \tag{C.2}$$

Let $x = \sigma\beta + \alpha$. We have $\varrho \le \frac{1}{x}$ due to Eqn. (C.1). Note that the right-hand side of Eqn. (C.2) is maximized when $\varrho = \frac{\beta-1}{2\beta}$. We consider the following cases:

- $\beta \in (0, 1]$: In this case, the right-hand side of Eqn. (C.2) is minimized when $\varrho = \frac{1}{x}$;
- $\beta > 1$ and $\frac{1}{x} \ge \frac{\beta-1}{\beta}$: In this case, the right-hand side of Eqn. (C.2) is minimized when $\varrho = \frac{1}{x}$;
- $\beta > 1$ and $\frac{1}{x} < \frac{\beta-1}{\beta}$: In this case, the right-hand side of Eqn. (C.2) is minimized when $\varrho = 0$;

So the lemma follows by combining the above cases. □

Appendix D. Proof of Theorem 2

Proof. Let $d = 4(\alpha+\beta) - 12 = 4(\varphi + \frac{1}{\varphi}) - 8 \approx 0.32$. Let $\delta = d - \epsilon$ and $\epsilon' = \min\{\frac{\epsilon}{5}, \delta\}$ (without loss of generality, we assume that $\epsilon < d$). Consider an instance of the maximum weighted coverage problem defined as follows. There are 17 weighted elements $\{a_1, \cdots, a_{17}\}$ with their weights defined as:

$$w(a_1) = 1 - \delta;\ w(a_5) = 1 + \delta;\ w(a_{17}) = \delta;$$

and $w(a_i) = 1$ for all other $i \notin \{1, 5, 17\}$. Let $\mathcal{N} = \{S_1, S_2, \cdots, S_5, S_6\}$ where

$$S_1 = \{a_1, a_2, a_3, a_4\}; S_2 = \{a_1, a_5, a_6, a_7\}; S_3 = \{a_2, a_8, a_9, a_{10}\};$$
$$S_4 = \{a_3, a_{11}, a_{12}, a_{13}\}; S_5 = \{a_4, a_{14}, a_{15}, a_{16}\}; S_6 = \{a_{17}\};$$

The costs of the elements in $\mathcal{N}$ are defined as:

$$c(S_1) = \frac{4-\delta}{4\alpha};\ c(S_3) = c(S_4) = c(S_5) = \frac{3+\epsilon'}{4(\alpha+\beta)};$$
$$c(S_2) = \frac{3+\delta+\epsilon'}{4(\alpha+\beta)};\ c(S_6) = \frac{\epsilon'}{4(\alpha+\beta)}$$

The objective submodular function is defined as

$$\text{for all } X \subseteq \mathcal{N} : f(X) = \sum_{u \in \cup_{S\in X} S} w(u) \tag{D.1}$$

So we have $f(S_1) = 4 - \delta; f(S_6) = \delta$ and for all $i \in \{2, 3, 4, 5\} : f(S_i) = 4$. Let the budget $B = 1$. Suppose that the TripleEagleRan algorithm visits $S_1, S_6, S_3, S_4, S_5, S_2$ sequentially. So it would return $A = \{S_1, S_6\}$ by calculating the prices as:

$$p(S_1) = \frac{4-\delta}{4\alpha} = c(S_1);\ p(S_6) = \frac{\delta}{\beta(4-\delta)+4\alpha} > \frac{\delta}{4(\alpha+\beta)} \ge c(S_6);$$
$$\text{for all } i \in \{3, 4, 5\} : p(S_i) = \frac{3}{4(\alpha+\beta)} < c(S_i) = \frac{3+\epsilon'}{4(\alpha+\beta)};$$
$$p(S_2) = \frac{3+\delta}{4(\alpha+\beta)} < c(S_2) = \frac{3+\delta+\epsilon'}{4(\alpha+\beta)};$$

However, the optimal solution is $O = \{S_2, S_3, S_4, S_5, S_6\}$ with a function value of $16 + \delta$, because

$$\begin{aligned} & c(S_2) + c(S_3) + c(S_4) + c(S_5) + c(S_6) \\ = & \frac{12+\delta+5\epsilon'}{4(\alpha+\beta)} \le \frac{12+d}{4(\alpha+\beta)} = 1 \end{aligned}$$

So the approximation ratio of TripleEagleRan on this problem instance is

$$\frac{f(O)}{f(\{S_1, S_6\})} = \frac{16+\delta}{4} = 1 + \alpha + \beta - \frac{\epsilon}{4} \ge 1 + \alpha + \beta - \epsilon = \frac{\varphi}{\varphi - 1} - \epsilon,$$

where the last equality is due to $\varphi^3 - \varphi - 1 = 0$. So the lemma follows. □

Appendix E. Proof of Theorem 3

Proof. To prove Theorem 3, we introduce a series of lemmas (i.e., Lemma 8-Lemma 15) to show that the approximation ratio of TripleEagleRan has specific lower bounds for different values of α and β, as summarized by Table E.1. Theorem 3 then follows by simply combining all these lemmas. Note that three lemmas (i.e., Lemmas 12-14) are introduced to prove Lemma 15. For the simplicity of description, we assume that division by zero is infinity in the proofs. □

Table E.1
The lower bounds of TripleEagleRan for different values of α, β.

Values of α, β	Lower bound	Lemma
$\alpha > 0, \beta = 0$	unbounded	Lemma 8
$\alpha \geq 0, \beta > 0, \alpha + \beta \leq 2$	4	Lemma 9
$\alpha \geq 0, \beta > 0, \alpha + \beta > 3$	4	Lemma 10
$\alpha \geq 0, \beta > 0, \alpha + \beta \in (2, 3], \frac{1}{\alpha} + \frac{1}{\alpha+\beta} > 1$	4	Lemma 11
$\alpha > 0, \beta > 0, \alpha + \beta \in (2, 3], \frac{1}{\alpha} + \frac{1}{\alpha+\beta} \leq 1$	$3.784 - \epsilon$	Lemma 15

Lemma 8. *For any positive α and $\beta = 0$, there is an instance for which the approximation ratio of* TripleEagleRan *is at least k for any positive integer k.*

Proof. Let $m = \max\{1, \lfloor \alpha \rfloor\}$. Suppose that $\mathcal{N} = \{v_i : i \in [km]\}$, $B = 1$ and $f(\cdot)$ is an additive function. Suppose that $f(v_i) = 1, c(v_i) = \min\{\frac{1}{\alpha}, \frac{1}{km}\}$ for any $i \in [km]$. Suppose that the TripleEagleRan algorithm visits v_i sequentially from $i = 1$ to km. So TripleEagleRan will offer a price $\min\{1, \frac{1}{\alpha}\}$ to each element in $\mathcal{N}$ and return the set $\{v_{km-m+1}, \cdots, v_{km}\}$ with a value of m. However, the optimal solution is $\mathcal{N}$ with a value of km. So the approximation ratio of TripleEagleRan on this instance is k, which completes the proof. $\square$

Lemma 9. *For any $\alpha \geq 0$ and $\beta > 0$ satisfying $\alpha + \beta \leq 2$, there exists an instance of the BFM problem for which the approximation ratio of the* TripleEagleRan *algorithm is at least* 4.

Proof. Let $k = \min\{i : i \in \mathbb{Z}^+ \wedge \alpha + i\beta > 2\}$. So we have $k \geq 2$. Consider an instance of the max-coverage problem where the ground set is $\mathcal{N} = \{S_1, S_2, \cdots, S_{k+4}\}$ defined as

$$\text{for all } i \in [k] : S_i = \{a_{4(i-1)+1}, a_{4(i-1)+2}, a_{4(i-1)+3}, a_{4(i-1)+4}\};$$

$$S_{k+1} = \{a_1, a_2, b_1, b_2\}, S_{k+2} = \{a_3, a_4, b_3, b_4\};$$

$$S_{k+3} = \{a_5, a_6, b_5, b_6\}, S_{k+4} = \{a_7, a_8, b_7, b_8\};$$

Suppose that $B = 1$ and for all $i \in [k] : c(S_i) = \min\{1, \frac{1}{\alpha+(i-1)\beta}\}$ and $c(S_{k+1}) = c(S_{k+2}) = c(S_{k+3}) = c(S_{k+4}) = \frac{1}{4}$, and the objective function is defined as for all $X \subseteq \mathcal{N} : f(X) = |\cup_{S_i : S_i \in X} S_i|$. Suppose that the TripleEagleRan algorithm visits $S_1, S_2, \cdots, S_{k+4}$ sequentially and hence offers the prices:

$$\text{for all } i \in [k] : p(S_i) = c(S_i) = \min\left\{1, \frac{1}{\alpha + (i-1)\beta}\right\} \geq \frac{1}{2};$$

$$\text{for all } i \in \{k+1, k+2, k+3, k+4\} : p(S_i) = \frac{1}{2(\beta k + \alpha)} < \frac{1}{4} = c(S_i)$$

So the TripleEagleRan algorithm would generate $A = \{S_1, \cdots, S_k\}$. Since $p(S_k) + p(S_{k-1}) > 1$, the algorithm would return $\{S_k\}$ with a value of 4. However, there exists a feasible solution $\{S_{k+1}, S_{k+2}, S_{k+3}, S_{k+4}\}$ with a value of 16. So the approximation ratio is at least 4. $\square$

Lemma 10. *For any $\alpha \geq 0$ and $\beta > 0$ satisfying $\alpha + \beta > 3$, there exists an instance of the BFM problem for which the approximation ratio of the* TripleEagleRan *algorithm is at least* 4.

Proof. Consider an instance of the max-coverage problem where there is a ground set $\mathcal{N} = \{S_1, S_2, \cdots, S_5\}$ defined as

$$S_1 = \{a_1, a_2, a_3, a_4\}; S_2 = \{a_1, b_1, b_2, b_3\}; S_3 = \{a_2, b_4, b_5, b_6\};$$

$$S_4 = \{a_3, b_7, b_8, b_9\}; S_5 = \{a_4, b_{10}, b_{11}, b_{12}\};$$

Suppose that $B = 1$, $c(S_1) = \min\{1, \frac{1}{2\alpha}\}$ and $c(S_2) = c(S_3) = c(S_4) = c(S_5) = \frac{1}{4}$, and the objective function is defined as $f(X) = |\cup_{S_i : S_i \in X} S_i|$ for all $X \subseteq \mathcal{N}$. Suppose that the TripleEagleRan algorithm visits $\{S_1, \cdots, S_5\}$ sequentially. So we have $\min\{1, \frac{1}{2\alpha}\} = c(S_1) \leq p(S_1) = \min\{1, \frac{1}{\alpha}\}$ and

$$\text{for all } i \in \{2, 3, 4, 5\} : c(S_i) = \frac{1}{4} > p(S_i) = \frac{B \cdot f(S_i \mid S_1)}{\beta f(S_1) + \alpha f(v^*)} = \frac{3}{4(\alpha + \beta)}$$

So the algorithm will return $\{S_1\}$ with a function value of 4. However, there exists a feasible solution $\{S_2, S_3, S_4, S_5\}$ with a function value of 16. So the approximation ratio is at least 4. $\square$

Lemma 11. *For any $\alpha \geq 0$ and $\beta > 0$ satisfying $\alpha + \beta \in (2, 3]$ and $\frac{1}{\alpha} + \frac{1}{\alpha+\beta} > 1$, there exists an instance of the BFM problem for which the approximation ratio of the* TripleEagleRan *algorithm is at least* 4.

Proof. Consider an instance of the max-coverage problem where there is a ground set $\mathcal{N} = \{S_1, S_2, \cdots, S_6\}$ and

$$S_1 = \{a_1, a_2, a_3, a_4\}; S_2 = \{a_5, a_6, a_7, a_8\}; S_3 = \{a_1, a_2, a_9, a_{10}\};$$
$$S_4 = \{a_3, a_4, a_{11}, a_{12}\}; S_5 = \{a_5, a_6, a_{13}, a_{14}\}; S_6 = \{a_7, a_8, a_{15}, a_{16}\};$$

Suppose that $B = 1$, $c(S_1) = \min\{1/\alpha, 1\}, c(S_2) = \frac{1}{\alpha+\beta}, c(S_3) = c(S_4) = c(S_5) = c(S_6) = \frac{1}{4}$, and the objective value function is $f(X) = |\cup_{S_i : S_i \in X} S_i|$. Suppose that the TripleEagleRan algorithm visits $S_1, \cdots, S_6$ sequentially, so we have:

$$p(S_1) = c(S_1) = \min\left\{1, \frac{1}{\alpha}\right\}; p(S_2) = c(S_2) = \frac{1}{\alpha+\beta}$$
$$\text{for all } i \in \{3,4,5,6\} : p(S_i) = \frac{1}{4\beta + 2\alpha} < \frac{1}{4} = c(S_i)$$

So the TripleEagleRan algorithm returns $\{S_2\}$ with a function value of 4. However, the optimal solution is $\{S_3, S_4, S_5, S_6\}$ with a function value of 16, which implies that the approximation ratio is at least 4. □

Lemma 12. *For any positive α and $\beta \in (0, 1)$ and any $\epsilon \in (0, 1)$, there exists an instance of the BFM problem for which the approximation ratio of the* TripleEagleRan *algorithm (with the input parameters α and β) is at least $(1 + \beta)\frac{e^\beta}{e^\beta - 1} - \epsilon$.*

Proof. Let $t \geq 1$ be a sufficiently large integer such that $\beta t + \alpha > 1$ and

$$\frac{(1+\beta)t + \alpha}{(e^\beta - 1)(t + \frac{\alpha}{\beta}) + 1} \geq \frac{(1+\beta)}{(e^\beta - 1)} - \frac{\epsilon}{2}, \tag{E.1}$$

and let $l \geq t + 1$ be the largest integer such that $\sum_{i=t+1}^{l} \frac{1}{\beta(i-1)+\alpha} \leq 1$ and $l \geq \frac{\alpha}{1-\beta}$. We consider an instance of the BFM problem with an additive function $f(\cdot)$, $B = 1$ and the ground set $\mathcal{N} = \{u_1, \cdots, u_{2l}, u_{2l+1}\}$, and the values and costs of the elements are defined as

$$\text{for all } i \in [l] : f(u_i) = 1, c(u_i) = \delta';\ f(u_{2l+1}) = 1; c(u_{2l+1}) = \frac{1+\delta}{\beta l + \alpha}$$
$$\text{for all } j : 2l \geq j \geq l + 1 : f(u_j) = \delta; c(u_j) = \frac{\delta}{\beta l + \alpha} + \delta'$$

where $\delta' = \frac{\epsilon}{4l(\beta l + \alpha)}$ and $\delta = (\beta + \frac{\alpha}{l})(1 - 2l\delta')$. It can be verified that the cost of each element is no more than 1 under such settings. Suppose that the TripleEagleRan algorithm visits $\{u_1, \cdots, u_{2l}, u_{2l+1}\}$ sequentially, so we have

$$\text{for all } i \in [l] : p(u_i) = \min\left\{\frac{1}{\beta(i-1)+\alpha}, 1\right\} \geq c(u_i);$$
$$\text{for all } j \in \{l+1, \cdots, 2l\} : p(u_i) = \min\left\{\frac{\delta}{\beta l + \alpha}, 1\right\} < c(u_j);$$
$$p(u_{2l+1}) = \frac{1}{\beta l + \alpha} < c(u_{2l+1})$$

Therefore, TripleEagleRan generates $A = \{u_1, \cdots, u_l\}$ and returns $S = \{u_{t+1}, \cdots, u_l\}$. According to the definition of l and t, we have

$$1 \geq \sum_{i=t+1}^{l} \frac{1}{\beta(i-1)+\alpha} = \frac{1}{\beta}\sum_{i=0}^{l-t-1} \frac{1}{(t+\frac{\alpha}{\beta}) + i} \geq \frac{1}{\beta}\ln(1 + \frac{l-t-1}{t+\frac{\alpha}{\beta}})$$

and hence $f(S) = l - t \leq (e^\beta - 1)(t + \frac{\alpha}{\beta}) + 1$. Let $Q = \{u_1, \cdots, u_{2l}\}$, we get

$$c(Q) = \frac{l\delta}{\beta l + \alpha} + 2l\delta' \leq 1$$
$$f(Q) = l + l\delta = l + (1 - 2l\delta')(\beta l + \alpha) \geq (1+\beta)l + \alpha - \frac{\epsilon}{2}$$

Combining this with Eqn. (E.1), we know that the approximation ratio of TripleEagleRan is at least

$$\frac{f(Q)}{f(S)} \geq \frac{(1+\beta)l + \alpha - \frac{\epsilon}{2}}{l - t} \geq 1 + \beta + \frac{(1+\beta)t + \alpha - \frac{\epsilon}{2}}{(e^\beta - 1)(t + \frac{\alpha}{\beta}) + 1} \geq 1 + \beta + \frac{1+\beta}{e^\beta - 1} - \epsilon$$

which completes the proof. □

Lemma 13. *For any positive α and β satisfying $\alpha+\beta\geq 2$ any $\epsilon\in(0,1)$, there exists an instance of the BFM problem for which the approximation ratio of the* TripleEagleRan *algorithm is at least $1+\beta+\alpha-\epsilon$.*

Proof. Let $\delta'=\frac{\epsilon}{2(\beta+\alpha)+\epsilon}$, $k=\lceil 100(1+\beta+\alpha)\rceil$ and $\delta=\frac{\beta+\alpha-\epsilon}{k}$. We consider an instance of the BFM problem with an additive function $f(\cdot)$, $B=1$ and the ground set $\mathcal{N}=\{u_1,u_2,\cdots,u_{k+1},u_{k+2}\}$, and the values and costs of the elements are defined as

$$f(u_1)=1, c(u_1)=\delta';\;\; f(u_{k+2})=1, c(u_{k+2})=1$$
$$\text{for all } i\in\{2,\cdots,k+1\}: f(u_i)=\delta, c(u_i)=\frac{\delta}{\beta+\alpha}+\frac{\delta'}{k};$$

Let $Q=\{u_1,u_2,\cdots,u_{k+1}\}$, so we have

$$c(Q)=\frac{k\delta}{\beta+\alpha}+2\delta'\leq\frac{\beta+\alpha-\epsilon}{\beta+\alpha}+\frac{\epsilon}{\beta+\alpha}\leq 1=B$$
$$f(Q)=1+k\delta=1+\beta+\alpha-\epsilon$$

Suppose that the TripleEagleRan algorithm visits $\{u_1,u_2,\cdots,u_{k+1},u_{k+2}\}$ sequentially, so we have

$$p(u_1)=\min\{1,1/\alpha\}\geq\delta'=c(u_1);$$
$$\text{for all } i\in\{2,\cdots,k+1\}: p(u_i)=\frac{\delta}{\beta+\alpha}<c(u_i)=\frac{\delta}{\beta+\alpha}+\frac{\delta'}{k}$$
$$p(u_{k+2})=\frac{1}{\beta+\alpha}\leq 1/2<c(u_{k+2})=1$$

and hence TripleEagleRan returns $S=\{u_1\}$. So the approximation ratio is at least $\frac{f(Q)}{f(u_1)}=1+\beta+\alpha-\epsilon$. $\square$

Lemma 14. *For any positive α,β satisfying $\alpha+\beta\in(2,3]$, $\frac{1}{\alpha}+\frac{1}{\alpha+\beta}\leq 1$ and any $\epsilon\in(0,1)$, there exists an instance of the BFM problem for which the approximation ratio of* TripleEagleRan *is at least $1+\beta+\frac{1+\alpha+\beta}{1+(\alpha+2\beta)(1-\frac{1}{\alpha}-\frac{1}{\alpha+\beta})}-\epsilon$.*

Proof. Note that $\frac{1}{\alpha}+\frac{1}{\alpha+\beta}+\frac{1}{\alpha+2\beta}>1$ for any positive α and β satisfying $2<\alpha+\beta\leq 3$. Let $z=(\alpha+2\beta)\left(1-\frac{1}{\alpha}-\frac{1}{\alpha+\beta}\right)$, so we have $z\in[0,1)$. Let

$$m=\lceil 100+\beta(2+z)+\alpha\rceil;\;\; \delta=\frac{(1-m\delta')(\beta(2+z+\delta')+\alpha)}{m-3}, \tag{E.2}$$

where $\delta'>0$ is a sufficiently small number in $(0,\min\{\frac{1}{12m},1-z\})$ such that

$$\frac{1+\beta+\alpha-12m\delta'}{1+(\alpha+2\beta)\left(1-\frac{1}{\alpha}-\frac{1}{\alpha+\beta}\right)+\delta'}\geq\frac{1+\beta+\alpha}{1+(\alpha+2\beta)\left(1-\frac{1}{\alpha}-\frac{1}{\alpha+\beta}\right)}-\epsilon \tag{E.3}$$

We consider an instance of the BFM problem with an additive function $f(\cdot)$, $B=1$ and the ground set $\mathcal{N}=\{u_1,\cdots,u_{m+1}\}$, and the values and costs of the elements are defined as

$$f(u_1)=1, c(u_1)=\delta'; f(u_2)=1, c(u_2)=\delta'; f(u_3)=z+\delta', c(u_3)=\delta'/6;$$
$$\text{for all } i: m\geq i\geq 4: f(u_i)=\delta, c(u_i)=\frac{\delta}{\beta(2+z+\delta')+\alpha}+\delta';$$
$$f(u_{m+1})=1, c(u_{m+1})=\frac{1}{\beta(2+z+\delta')+\alpha}+\delta';$$

Suppose that the TripleEagleRan algorithm visits $\{u_1,\cdots,u_{m+1}\}$ sequentially. So we have

$$p(u_1)=\min\{1/\alpha,1\}\geq c(u_1)=\delta'; p(u_2)=\frac{1}{\beta+\alpha}\geq c(u_2)=\delta';$$
$$p(u_3)=\frac{z+\delta'}{2\beta+\alpha}=1-\frac{1}{\alpha}-\frac{1}{\alpha+\beta}+\frac{\delta'}{2\beta+\alpha}\geq c(u_3)=\delta'/6$$
$$\text{for all } i: m\geq i\geq 4: p(u_i)=\frac{\delta}{\beta(2+z+\delta')+\alpha}<c(u_i)$$
$$p(u_{m+1})=\frac{1}{\beta(2+z+\delta')+\alpha}<c(u_{m+1})$$

So it generates $A=\{u_1,u_2,u_3\}$ and returns $S=\{u_2,u_3\}$. Let $Q=\{u_1,\cdots,u_m\}$. So Q is a feasible solution due to

$$f(Q)=f(A)+(m-3)\delta;\;\; c(Q)\leq m\delta'+\frac{(m-3)\delta}{\beta(2+z+\delta')+\alpha}\leq 1$$

Note that $\beta f(A) + \alpha = \beta(2 + z + \delta') + \alpha \leq 4\beta + \alpha \leq 12$. So the approximation ratio of TripleEagleRan on this problem instance is at least

$$\begin{aligned}\frac{f(Q)}{f(S)} &= \frac{f(A) + (1 - m\delta')(\beta f(A) + \alpha)}{f(S)} \geq \frac{(1+\beta)f(A) + \alpha - 12m\delta'}{f(S)} \\ &= 1 + \beta + \frac{1 + \beta + \alpha - 12m\delta'}{1 + (\alpha + 2\beta)(1 - \frac{1}{\alpha} - \frac{1}{\alpha+\beta}) + \delta'}\end{aligned} \tag{E.4}$$

So the lemma follows by combining Eqn. (E.3) and Eqn. (E.4). □

Lemma 15. *For any positive α and β satisfying $\alpha + \beta \in (2, 3]$, $\frac{1}{\alpha} + \frac{1}{\alpha+\beta} \leq 1$ and any $\epsilon \in (0, 1)$, there exists an instance of the BFM problem for which the approximation ratio of* TripleEagleRan *is at least* $3.784 - \epsilon$.

Proof. Let $g_1 = 1 + \alpha + \beta$; $g_2 = (1 + \beta)\frac{e^\beta}{e^\beta - 1}$ and

$$g_3 = 1 + \beta + \frac{1 + \alpha + \beta}{1 + (\alpha + 2\beta)(1 - \frac{1}{\alpha} - \frac{1}{\alpha+\beta})} \tag{E.5}$$

It can be verified that g_2 decreases when β increases for any $\beta \in (0, 1)$, so we have $g_2 \geq 3.784$ when $\beta \leq 0.5086$. If $\alpha + \beta \geq 2.784$, then we have $g_1 \geq 3.784$. Moreover, it can be verified that, when $\beta \geq 0.5086$ and $\alpha + \beta \in (2, 2.784]$ and $\frac{1}{\alpha} + \frac{1}{\alpha+\beta} \leq 1$, we have: (1) g_3 decreases with the decreasement of β for any fixed value of $\alpha + \beta$; and (2) g_3 decreases with the increment of α for any fixed value of β. Therefore, when $\beta \geq 0.5086$ and $\alpha + \beta \leq 2.784$, g_3 is minimized to 3.784 when $\beta = 0.5086$ and $\alpha + \beta = 2.784$. The lemma then follows by combining the above results with Lemmas 12-14. □

Appendix F. Proof of Lemma 6

Proof. Let O_1, O_2 be the set of sellers in O who reject the offered prices in Line 4 of Algorithm 3 and in the LSDPricing procedure (Line 10 of Algorithm 3), respectively. Let K_u denote the set of elements already in K when u is offered a price in Line 3. Under the case of $f(K) < f(v^*)$, we must have $S = \{v^*\}$ (due to Line 8), $O_1 = O \setminus \{v^*\} \setminus K$ and

$$\begin{aligned}f(O \setminus \{v^*\}) - f(K) &\leq \sum_{u \in O_1} f(u \mid K_u) = \sum_{u \in O_1} p(u) \cdot \frac{\alpha f(v^*)}{B} \\ &\leq \sum_{u \in O_1} c(u)\frac{\alpha f(v^*)}{B} \leq \alpha f(v^*)\end{aligned} \tag{F.1}$$

and hence

$$f(O) \leq f(K) + \alpha f(v^*) + f(v^*) \leq (2 + \alpha)f(v^*) = (2 + \alpha)f(S) \tag{F.2}$$

On the other hand, if $f(K) \geq f(v^*)$ and $p(A) \leq B$, then we have $O \setminus A = O_1 \cup O_2$. Note that K is prefix of A under this case. So we can use submodularity to get

$$\begin{aligned}f(O) - f(A) &\leq \sum_{u \in O_1} f(u \mid K_u) + \sum_{u \in O_2} f(u \mid A_u) \\ &\leq \sum_{u \in O_1} c(u)\frac{\alpha f(v^*)}{B} + \sum_{u \in O_2} c(u)\frac{f(A_u) + \alpha f(v^*)}{B} \\ &\leq c(O_1) \cdot \frac{\alpha f(v^*)}{B} + c(O_2) \cdot \frac{f(A) + \alpha f(v^*)}{B} \\ &\leq c(O) \cdot \frac{f(A) + \alpha f(v^*)}{B} \\ &\leq f(A) + \alpha f(v^*),\end{aligned} \tag{F.3}$$

which yields $f(O) \leq 2f(A) + \alpha f(v^*) \leq (2 + \alpha)f(A)$ by using $f(A) \geq f(K) \geq f(v^*)$. Note that $S = A$ under this case (due to Line 11), so the lemma follows. □

Appendix G. Proof of Lemma 7

Proof. By similar reasoning with Lemma 6, we have

$$f(O) \leq 2f(A) + \alpha f(v^*) \tag{G.1}$$

Let H be the smallest suffix of A satisfying $p(H) > B$, so $H \setminus S$ contains exactly one element (denoted by w) which is added into A right before the elements in S. For convenience, we define $\delta(X) = \sum_{u\in X} f(u \mid A_u)$ for any regular subset X of A. Suppose that $f(A \setminus H) = \mu f(v^*)$, $f(K) = \lambda f(v^*)$, $\delta(K \cap H) = \tau f(v^*)$ and $p(K \cap H) = \rho B$. Clearly, we have $p(H \setminus K) \geq (1-\rho)B$, $\lambda \geq 1$ and $\tau \leq \lambda \leq 2$, where $\lambda \leq 2$ is due to the reason that we immediately stop adding items to set K once $f(K) \geq f(v^*)$ (see Line 6 of Algorithm 3). We consider the following cases:

1. $K \cap H = \emptyset$: In this case, all the sellers in H are selected using the pricing rule of Algorithm 2. So we can use similar reasoning as Eqn. (5) in Lemma 4 (by setting $\beta = 1$) to get

$$f(O) \leq \left(4 + \frac{2-\alpha+2/x}{x-1/x}\right) f(S) \tag{G.2}$$

where $x = \mu + \alpha$. Note that $f(K) \geq f(v^*)$ and $K \subseteq A \setminus H$, so $\mu \geq 1$ and $x \geq \alpha + 1$. The right-hand side of Eqn (G.2) is no more than $4f(S)$ when $x \geq \frac{2}{\alpha-2}$ or $\alpha + 1 \geq \frac{2}{\alpha-2}$. When $\alpha + 1 < \frac{2}{\alpha-2}$ (i.e., $\alpha < \frac{\sqrt{17}+1}{2}$) and $x \in [\alpha+1, \frac{2}{\alpha-2})$, the right-hand side of Eqn (G.2) is maximized when $x = \alpha + 1$. Therefore, for any $\alpha \in (2, \frac{\sqrt{17}+1}{2})$ we have

$$f(O) \leq \left(4 + \frac{\alpha - \alpha^2 + 4}{\alpha^2 + 2\alpha}\right) f(S) = \left(3 + \frac{3\alpha+4}{\alpha^2+2\alpha}\right) f(S) \tag{G.3}$$

2. $|K \cap H| > 1$: In this case, we must have $|K \cap S| \geq 1$ because $|H \setminus S| = 1$ by definition. According to the construction of K, this implies $f(K \setminus S) < f(v^*)$ and hence

$$f(A) \leq f(K \setminus S) + f(S) \leq f(v^*) + f(S) \tag{G.4}$$

Note that each seller $u \in K$ satisfies $c(u) \leq p(u) = \frac{Bf(u|A_u)}{\alpha f(v^*)}$, so we have

$$\tau f(v^*) = \delta(K \cap H) \geq p(K \cap H)\frac{\alpha f(v^*)}{B} \geq \rho\alpha f(v^*) \tag{G.5}$$

This implies $\rho \leq \tau/\alpha$ and hence $p(H \setminus K) \geq (1 - \tau/\alpha)B$. By similar reasoning as Lemma 1, we have

$$\delta(H \setminus K) \geq (1-\rho)\left(f(K) + \alpha \cdot f(v^*)\right) \geq (1-\tau/\alpha)(\lambda+\alpha)\cdot f(v^*)$$

and hence

$$\begin{aligned}\delta(H) &= \delta(K \cap H) + \delta(H \setminus K)\\ &\geq [\tau + (1-\tau/\alpha)(\lambda+\alpha)]\cdot f(v^*) = (\alpha + \lambda - \tau\lambda/\alpha) f(v^*)\\ &\geq (\alpha + \lambda - \lambda^2/\alpha) f(v^*),\end{aligned} \tag{G.6}$$

where the last inequality is due to $\tau \leq \lambda$. Recall that $1 \leq \lambda \leq 2$ and $\alpha \in (2, \frac{\sqrt{17}+1}{2})$. So the right-hand side of Eqn. (G.6) is minimized when $\lambda = 2$. Therefore, we get

$$f(S) \geq f(H) - f(v^*) \geq (\alpha + 1 - 4/\alpha) f(v^*) \tag{G.7}$$

Combining Eqn. (G.1), Eqn. (G.4) and Eqn. (G.7), we get

$$f(O) \leq 2f(S) + (2+\alpha)f(v^*) \leq \left(2 + \frac{2+\alpha}{\alpha+1-4/\alpha}\right) f(S) \tag{G.8}$$

3. $|K \cap H| = 1$: In this case, we must have $A = K \cup S$, $K \cap S = \emptyset$ and $H \setminus K = S$ according to the definition of H. Moreover, due to Eqn. (G.5) we have $f(v^*) \geq \delta(K \cap H) \geq \rho\alpha f(v^*)$ and hence $\rho \leq 1/\alpha$. Note that all the elements in S are selected using the pricing rule of Algorithm 2. Therefore, using similar reasoning as Lemma 1, we get

$$f(S) \geq (1-\rho)\left(f(K) + \alpha \cdot f(v^*)\right) \geq (1-1/\alpha)(\lambda+\alpha) f(v^*) \tag{G.9}$$

Recall that $\lambda \leq 2$ and $A = K \cup S$. Combining this with Eqn. (G.1) and Eqn. (G.9), we get

$$\begin{aligned}f(O) &\leq 2f(A) + \alpha f(v^*) \leq 2f(K) + 2f(S) + \alpha f(v^*)\\ &= 2f(S) + (2\lambda+\alpha) f(v^*)\\ &\leq \left(2 + \frac{2\lambda+\alpha}{(1-1/\alpha)(\lambda+\alpha)}\right) f(S)\\ &= \left(2 + \frac{2}{1-1/\alpha} - \frac{\alpha}{(1-1/\alpha)(\lambda+\alpha)}\right) f(S)\end{aligned}$$

$$\leq \left(2 + \frac{4+\alpha}{\alpha + 1 - 2/\alpha}\right) f(S) \tag{G.10}$$

The lemma then follows according to the above discussion. □

Appendix H. Proof of Theorem 5

It can be easily seen that Algorithm 4 offers one price to each seller and incurs at most $3n$ value oracle queries (n queries for finding v^* and $2n$ queries for building A_1 and A_2). The approximation ratio of Algorithm 4 can be proved by directly combining Lemma 16 and Lemma 17:

Lemma 16. *If Line 9 of Algorithm 4 is executed, then we have $f(O) \leq \frac{4\psi}{\psi-1} f(S)$ by setting $\alpha = 1+\psi$ and $\beta = \frac{2}{\psi}$.*

Proof. In this case, all the elements in O must have been processed by Lines 3-5. Let O_1 (resp. O_2) denote the sellers in O who reject the offered prices that are calculated using A_1 (resp. A_2) in Line 4. For convenience, we define $\delta_i(u) = f(u \mid A_{i,u})$ for any $u \in \mathcal{N}$, where $A_{i,u}$ denotes the set of elements already in A_i at the moment that u is processed by Algorithm 4. It can be seen that, for each $u \in A_i (i \in \{1,2\})$, we have $\delta_i(u) \geq 0$, because otherwise we have $p(u) = \frac{B\delta_i(u)}{\beta f(A_{i,u}) + \alpha f(v^*)} < 0$ and hence $p(u)$ should be rejected by u due to $c(u) \geq 0$, which implies that u is not in A_i; a contradiction. So we have $f(X) \leq f(A_i)$ for any prefix X of A_i.

In the following, we consider the sets A_1, A_2, A^* after Line 9 of Algorithm 4 is executed. For each $i \in \{1,2\}$ and each $u \in O_i$, we can use similar reasoning as Lemma 1 to get

$$\frac{f(u \mid A_i)}{c(u)} \leq \frac{\delta_i(u)}{c(u)} < \frac{\beta f(A_{i,u}) + \alpha f(v^*)}{B} \leq \frac{\beta f(A_i) + \alpha f(v^*)}{B} \leq \frac{\beta f(A^*) + \alpha f(v^*)}{B}$$

Moreover, according to submodularity and the greedy rule of Line 3, for all $i, j \in \{1,2\}$ and $i \neq j$ we have

$$\sum_{u \in A_i \cap O} f(u \mid A_j) \leq \sum_{u \in A_i \cap O} \delta_j(u) \leq \sum_{u \in A_i \cap O} \delta_i(u) \leq \sum_{u \in A_i} \delta_i(u) \leq f(A_i) \leq f(A^*),$$

where the third inequality is due to $\delta_i(u) \geq 0$ for all $u \in A_i$. Using the above inequalities and submodularity, we get

$$\begin{aligned}
&f(O \cup A_1) - f(A_1) \leq \sum_{u \in O \cap A_2} f(u \mid A_1) + \sum_{u \in O_1 \cup O_2} f(u \mid A_1) \\
&\leq f(A^*) + \sum_{u \in O_1} \delta_1(u) + \sum_{u \in O_2} \delta_1(u) \leq f(A^*) + \sum_{u \in O_1} \delta_1(u) + \sum_{u \in O_2} \delta_2(u) \\
&\leq f(A^*) + [c(O_1) + c(O_2)] \cdot \frac{\beta f(A^*) + \alpha f(v^*)}{B} \\
&\leq (1+\beta) f(A^*) + \alpha f(v^*)
\end{aligned} \tag{H.1}$$

where the third inequality is due to the greedy rule in Line 3. Similarly, we get

$$f(O \cup A_2) - f(A_2) \leq (1+\beta) f(A^*) + \alpha f(v^*) \tag{H.2}$$

Due to the non-negativity and submodularity of $f(\cdot)$ and the fact that $A_1 \cap A_2 = \emptyset$, we get

$$\begin{aligned}
\sum_{i \in \{1,2\}} f(O \cup A_i) &\geq f(O \cup A_1 \cup A_2) + f((O \cup A_1) \cap (O \cup A_2)) \\
&\geq f((O \cup A_1) \cap (O \cup A_2)) = f(O)
\end{aligned} \tag{H.3}$$

Combining Eqns. (H.1)-(H.3) gives us

$$f(O) \leq \sum_{i \in \{1,2\}} f(O \cup A_i) \leq 2(2+\beta) f(A^*) + 2\alpha f(v^*), \tag{H.4}$$

where the first inequality is due to submodularity and $A_1 \cap A_2 = \emptyset$. Therefore, if $p(A^*) \leq B$, then we have $S = A^*$, $f(S) \geq f(v^*)$ and hence

$$f(O) \leq 2(2+\beta) f(S) + 2\alpha f(v^*) \leq 2(2+\alpha+\beta) f(S) \tag{H.5}$$

If $p(A^*) > B$, then let H be the smallest suffix of A^* satisfying $p(H) > B$, and suppose that $f(A^* \setminus H) = \sigma f(v^*)$ and $x = \sigma\beta + \alpha$. Using Eqn. (H.4), we get

$$\begin{aligned}
f(O) &\leq 2(2+\beta) f(H) + 2(2+\beta) f(A^* \setminus H) + 2\alpha f(v^*) \\
&\leq 2[(2+\beta) f(S) + ((2+\beta)(\sigma+1) + \alpha) f(v^*)]
\end{aligned}$$

$$= 2\left[(2+\beta)f(S) + \left(\frac{2+\beta}{\beta}x - \frac{2\alpha}{\beta} + 2 + \beta\right) f(v^*)\right] \tag{H.6}$$

When $\alpha = 1+\psi$ and $\beta = \frac{2}{\psi}$, we have $\beta > 1$ and $\frac{\beta(\beta+2)}{2} \le \alpha \le \frac{\beta}{\beta-1}$, which implies that $\beta + 2 - \frac{2\alpha}{\beta} \le 0$. Note that Lemma 3 still holds for TripleEagleNm due to that the pricing rule in Line 4 of TripleEagleNm is the same with that in Algorithm 2. According to Lemma 3, we discuss about the following cases:

1. $x \le \frac{\beta}{\beta-1}$: In this case, we have $f(S) \ge (x-1)(1+\frac{\beta}{x})f(v^*)$. Combining this with Eqn. (H.6) gives us

$$f(O) \le 2\left[(3+\beta+\frac{2}{\beta}) + \frac{\frac{2+\beta}{x} + 1 + \frac{2-2\alpha}{\beta}}{x+\beta-1-\frac{\beta}{x}}\right] f(S) \tag{H.7}$$

   Therefore, if $x \le \frac{\beta(\beta+2)}{2\alpha-\beta-2}$, then we have $\frac{2+\beta}{x} + 1 + \frac{2-2\alpha}{\beta} \ge 0$ and the right-hand side of Eqn. (H.7) is maximized when $\sigma = 0$, and otherwise the right-hand side of Eqn. (H.7) is no more than $2[(3+\beta+\frac{2}{\beta})]f(S)$. Therefore, we have

$$f(O) \le 2\max\left\{3+\beta+\frac{2}{\beta}, 2+\beta+\frac{2+\beta+\alpha}{(\alpha-1)(1+\frac{\beta}{\alpha})}\right\} f(S) \tag{H.8}$$

2. $x \ge \frac{\beta}{\beta-1}$: In this case, we have $f(S) \ge x \cdot f(v^*)$ according to Lemma 3, so the right-hand side of Eqn. (H.6) is no more than $2[(3+\beta+\frac{2}{\beta})]f(S)$ since $\beta + 2 \le \frac{2\alpha}{\beta}$ when $\alpha = 1+\psi$ and $\beta = \frac{2}{\psi}$.

Combining the above reasoning, we have

$$f(O) \le 2\max\left\{2+\alpha+\beta, 3+\beta+\frac{2}{\beta}, 2+\beta+\frac{2+\beta+\alpha}{(\alpha-1)(1+\frac{\beta}{\alpha})}\right\} f(S) \tag{H.9}$$

The lemma then follows by substituting $\alpha = 1+\psi$ and $\beta = \frac{2}{\psi}$ into the above inequality. □

Lemma 17. *If Lines 11-18 of Algorithm 4 are executed, then we have $f(O) \le 2(\alpha+\beta+2)\mathbb{E}[f(S)]$ and hence $f(O) \le \frac{4\psi}{\psi-1}f(S)$ by setting $\alpha = 1+\psi$ and $\beta = \frac{2}{\psi}$.*

Proof. For clarity, we use J_1 (resp. J_2) to denote the set A_1 (resp. A_2) at the moment that Algorithm 4 executes Line 11, and define $J^* = \arg\max_{X\in\{J_1,J_2\}} f(X)$. So we have $f(J^*) < f(v^*)$ due to Line 6. It can be seen from Line 17 that v^* is added into A_q only if (1) $\max_{i\in\{1,2\}} f(v^* \mid J_i) \ge 0$ and (2) v^* accepts the price of $B - p(J_q)$. Note that the set T generated by Line 18 must satisfy $f(T) \ge f(J^*)$. In the sequel, we prove that $f(O) \le 2(2+\beta)f(T) + 2\alpha f(v^*)$ by a discussion:

1. $v^* \notin O$: In this case, we can use similar reasoning as Lemma 16 to get

$$\begin{aligned} f(O) = f(O \setminus \{v^*\}) &\le 2(2+\beta)f(J^*) + 2\alpha f(v^*) \\ &\le 2(2+\beta)f(T) + 2\alpha f(v^*) \end{aligned} \tag{H.10}$$

2. $v^* \in O$ and v^* is added into A_q in Line 17; or $v^* \in O$ and $\max_{i\in\{1,2\}} f(v^* \mid J_i) < 0$: In these situations, after Line 18 is executed, we can use similar reasoning as Lemma 16 to get

$$\text{for all } i \in \{1,2\} : f(O \cup A_i) - f(A_i) \le (1+\beta)f(T) + \alpha f(v^*) \tag{H.11}$$

   Note that $f(A_i) \le f(T)$ due to Line 18. So we get

$$f(O) \le \sum_{i\in\{1,2\}} f(O \cup A_i) \le 2(2+\beta)f(T) + 2\alpha f(v^*) \tag{H.12}$$

   where the first inequality is got by similar reasoning as that in Eqn. (H.3).
3. $v^* \in O$ and $\max_{i\in\{1,2\}} f(v^* \mid J_i) \ge 0$ and $\{v^*\}$ rejects the price in Line 17: In this case, we must have $c(v^*) > B - p(J_q)$ and hence

$$c(O \setminus \{v^*\}) \le p(J_q) \le \frac{B \cdot f(J_q)}{f(\emptyset) + \alpha f(v^*)} \le \frac{Bf(J_q)}{\alpha f(v^*)} \le \frac{Bf(J^*)}{\alpha f(v^*)} \le \frac{B}{\alpha} \tag{H.13}$$

where the second inequality is got by Lemma 1. Note that all the elements in $O \setminus \{v^*\}$ have been processed by Lines 3-5. Let O_1 (resp. O_2) denote the set of sellers in $O \setminus \{v^*\}$ who reject the offered prices that are calculated using A_1 (resp. A_2) in Line 4. By similar reasoning with Lemma 16, we get

$$\text{for all } i \in \{1,2\}, \text{for all } u \in O_i : \frac{f(u \mid J_i)}{c(u)} \le \frac{\delta_i(u)}{c(u)} \le \frac{\beta f(J^*) + \alpha f(v^*)}{B}; \tag{H.14}$$

$$\text{for all } i,j \in \{1,2\} \wedge i \neq j : \sum_{u \in O\setminus\{v^*\} \cap J_i} f(u \mid J_j) \le f(J^*); \tag{H.15}$$

where $\delta_i(u)$ is defined in the same way as that in the proof of Lemma 16. Therefore, we can use Eqns. (H.13)-(H.15) to get

$$\begin{aligned} f(O \setminus \{v^*\} \cup J_1) - f(J_1) &\le \sum_{u \in O\setminus\{v^*\} \cap J_2} f(u \mid J_1) + \sum_{u \in O_1 \cup O_2} f(u \mid J_1) \\ &\le f(J^*) + \sum_{u \in O_1} \delta_1(u) + \sum_{u \in O_2} \delta_2(u) \qquad \text{(H.16)} \\ &\le f(J^*) + c(O \setminus \{v^*\}) \cdot \frac{\beta f(J^*) + \alpha f(v^*)}{B} \\ &\le \left(1 + \frac{\beta}{\alpha}\right) f(J^*) + f(v^*), \qquad \text{(H.17)} \end{aligned}$$

where Eqn. (H.16) is due to submodularity and the greedy rule in Line 4. Similarly, we get

$$f(O \setminus \{v^*\} \cup J_2) - f(J_2) \le \left(1 + \frac{\beta}{\alpha}\right) f(J^*) + f(v^*) \tag{H.18}$$

Combining the above equations, we get

$$f(O \setminus \{v^*\}) \le \sum_{i=1}^{2} f(O \setminus \{v^*\} \cup J_i) \le 2\left(2 + \frac{\beta}{\alpha}\right) f(J^*) + 2 f(v^*) \tag{H.19}$$

where the first inequality is got by similar reasoning as that in Eqn. (H.3). Therefore, we have

$$f(O) \le f(O \setminus \{v^*\}) + f(v^*) \le 2\left(2 + \frac{\beta}{\alpha}\right) f(J^*) + 3 f(v^*), \tag{H.20}$$

Therefore, when $\alpha \ge 1.5$, we have

$$f(O) \le 2(2+\beta) f(T) + 2\alpha f(v^*) \tag{H.21}$$

According to Lines 11-18 of Algorithm 4, the algorithm returns T or $\{v^*\}$ with probability of $\frac{\beta+2}{\alpha+\beta+2}$ or $\frac{\alpha}{\alpha+\beta+2}$, respectively. So we get

$$\begin{aligned} \mathbb{E}[f(S)] &= \Pr[S = T] \cdot f(T) + \Pr[S = \{v^*\}] \cdot f(v^*) \\ &= \frac{(\beta+2) f(T)}{\alpha+\beta+2} + \frac{\alpha f(v^*)}{\alpha+\beta+2} \ge \frac{f(O)}{2(\alpha+\beta+2)} \end{aligned} \tag{H.22}$$

The lemma then follows by substituting $\alpha = 1 + \psi$ and $\beta = \frac{2}{\psi}$ into the above inequality. $\square$

Data availability

Data will be made available on request.

References

[1] M. Akbarpour, S. Li, Credible auctions: a trilemma, Econometrica 88 (2020) 425–467.
[2] G. Amanatidis, G. Birmpas, E. Markakis, On budget-feasible mechanism design for symmetric submodular objectives, in: International Conference on Web and Internet Economics (WINE), 2017, pp. 1–15.
[3] G. Amanatidis, P. Kleer, G. Schäfer, Budget-feasible mechanism design for non-monotone submodular objectives: offline and online, in: ACM Conference on Economics and Computation (EC), 2019, pp. 901–919.
[4] G. Amanatidis, P. Kleer, G. Schäfer, Budget-feasible mechanism design for non-monotone submodular objectives: offline and online, Math. Oper. Res. 47 (2022) 2286–2309.
[5] N. Anari, G. Goel, A. Nikzad, Mechanism design for crowdsourcing: an optimal 1-1/e competitive budget-feasible mechanism for large markets, in: IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2014, pp. 266–275.
[6] C. Aslay, F. Bonchi, L.V.S. Lakshmanan, W. Lu, Revenue maximization in incentivized social advertising, Proc. VLDB Endow. 10 (2017) 1238–1249.
[7] L.M. Ausubel, P. Milgrom, et al., The lovely but lonely Vickrey auction, Comb. Auctions 17 (2006) 22–26.
[8] M. Babaioff, N. Immorlica, B. Lucier, S.M. Weinberg, A simple and approximately optimal mechanism for an additive buyer, J. ACM 67 (2020) 1–40.
[9] E. Balkanski, A. Breuer, Y. Singer, Non-monotone submodular maximization in exponentially fewer iterations, in: Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 2359–2370.

[10] E. Balkanski, P. Garimidi, V. Gkatzelis, D. Schoepflin, X. Tan, Deterministic budget-feasible clock auctions, in: ACM-SIAM Symposium on Discrete Algorithms (SODA), 2022, pp. 2940–2963.

[11] X. Bei, N. Chen, N. Gravin, P. Lu, Budget feasible mechanism design: from prior-free to Bayesian, in: ACM Symposium on the Theory of Computing (STOC), 2012, pp. 449–458.

[12] X. Bei, N. Chen, N. Gravin, P. Lu, Worst-case mechanism design via Bayesian analysis, SIAM J. Comput. 46 (2017) 1428–1448.

[13] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: ACM-SIAM Symposium on Discrete Algorithms (SODA), 2014, pp. 946–957.

[14] S. Chawla, J.D. Hartline, D.L. Malec, B. Sivan, Multi-parameter mechanism design and sequential posted pricing, in: ACM Symposium on Theory of Computing (STOC), 2010, pp. 311–320.

[15] N. Chen, N. Gravin, P. Lu, On the approximability of budget feasible mechanisms, in: ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 685–699.

[16] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2010, pp. 1029–1038.

[17] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 199–208.

[18] X. Chen, G. Matikas, D. Paparas, M. Yannakakis, On the complexity of simple and optimal deterministic mechanisms for an additive buyer, in: ACM-SIAM Symposium on Discrete Algorithms (SODA), 2018, pp. 2036–2049.

[19] Z. Cong, X. Luo, J. Pei, F. Zhu, Y. Zhang, Data pricing in machine learning pipelines, Knowl. Inf. Syst. 64 (2022) 1417–1455.

[20] P. Dandekar, N. Fawaz, S. Ioannidis, Privacy auctions for recommender systems, ACM Trans. Econ. Comput. 2 (2014) 12:1–12:22.

[21] S. Dobzinski, C.H. Papadimitriou, Y. Singer, Mechanisms for complement-free procurement, in: Proceedings of the 12th ACM Conference on Electronic Commerce (EC), 2011, pp. 273–282.

[22] K. El-Arini, G. Veda, D. Shahaf, C. Guestrin, Turning down the noise in the blogosphere, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 289–298.

[23] M. Fahrbach, V. Mirrokni, M. Zadimoghaddam, Non-monotone submodular maximization with nearly optimal adaptivity and query complexity, in: International Conference on Machine Learning (ICML), 2019, pp. 1833–1842.

[24] D. Gale, The law of supply and demand, Math. Scand. 3 (1955) 155–169.

[25] G. Goel, A. Nikzad, A. Singla, Mechanism design for crowdsourcing markets with heterogeneous tasks, in: J.P. Bigham, D.C. Parkes (Eds.), Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP), 2014, pp. 77–86.

[26] A. Goyal, W. Lu, L.V.S. Lakshmanan, CELF++: optimizing the greedy algorithm for influence maximization in social networks, in: Proceedings of the 20th International Conference on World Wide Web (WWW), 2011, pp. 47–48.

[27] N. Gravin, Y. Jin, P. Lu, C. Zhang, Optimal budget-feasible mechanisms for additive valuations, ACM Trans. Econ. Comput. 8 (2020) 21:1–21:15.

[28] K. Han, S. Cui, T. Zhu, E. Zhang, B. Wu, Z. Yin, T. Xu, S. Tang, H. Huang, Approximation algorithms for submodular data summarization with a knapsack constraint, Proc. ACM Meas. Anal. Comput. Syst. 5 (2021) 1–31.

[29] K. Han, K. Huang, X. Xiao, J. Tang, A. Sun, X. Tang, Efficient algorithms for adaptive influence maximization, Proc. VLDB Endow. 11 (2018) 1029–1040.

[30] K. Han, B. Wu, J. Tang, S. Cui, C. Aslay, L.V. Lakshmanan, Efficient and effective algorithms for revenue maximization in social advertising, in: ACM Conference on Management of Data (SIGMOD), 2021, pp. 671–684.

[31] K. Han, Y. Wu, H. Huang, S. Cui, Triple eagle: simple, fast and practical budget-feasible mechanisms, in: Advances in Neural Information Processing Systems (NeurIPS), 2023.

[32] H. Huang, K. Han, S. Cui, J. Tang, Randomized pricing with deferred acceptance for revenue maximization with submodular objectives, in: International World Wide Web Conference (WWW), 2023, pp. 3530–3540.

[33] P. Jalaly, É. Tardos, Simple and efficient budget feasible mechanisms for monotone submodular valuations, in: The 14th International Conference on Web and Internet Economics (WINE), 2018, pp. 246–263.

[34] P. Jalaly, É. Tardos, Simple and efficient budget feasible mechanisms for monotone submodular valuations, ACM Trans. Econ. Comput. 9 (2021) 1–20.

[35] J.H. Kagel, R.M. Harstad, D. Levin, Information impact and allocation rules in auctions with affiliated private values: a laboratory study, Econometrica 55 (1987) 1275–1304.

[36] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R.G.L. D'Oliveira, H. Eichner, S.E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P.B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S.U. Stich, Z. Sun, A.T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F.X. Yu, H. Yu, S. Zhao, Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (2021) 1–210.

[37] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: ACM International Conference on Knowledge Discovery and Data Mining (KDD), 2003, pp. 137–146.

[38] A. Kulik, H. Shachnai, T. Tamir, Approximations for monotone and nonmonotone submodular maximization with knapsack constraints, Math. Oper. Res. 38 (2013) 729–739.

[39] J. Leskovec, A. Krevl, Snap datasets: Stanford large network dataset collection, http://snap.stanford.edu/, 2014.

[40] S. Li, Obviously strategy-proof mechanisms, Am. Econ. Rev. 107 (2017) 3257–3287.

[41] X. Liu, H. Chan, M. Li, W. Wu, Y. Zhao, Budget-feasible mechanisms for proportionally selecting agents from groups, Artif. Intell. 323 (2023) 103975.

[42] Y. Luo, N.R. Jennings, A budget-limited mechanism for category-aware crowdsourcing of multiple-choice tasks, Artif. Intell. 299 (2021) 103538.

[43] P. Milgrom, I. Segal, Clock auctions and radio spectrum reallocation, J. Polit. Econ. 128 (2020) 1–31.

[44] R.B. Myerson, Optimal auction design, Math. Oper. Res. 6 (1981) 58–73.

[45] N. Nisan, T. Roughgarden, É. Tardos, V.V. Vazirani (Eds.), Algorithmic Game Theory, Cambridge University Press, 2007.

[46] A. Rubinstein, On the computational complexity of optimal simple mechanisms, in: Innovations in Theoretical Computer Science (ITCS), 2016, pp. 21–28.

[47] A. Rubinstein, J. Zhao, Beyond worst-case budget-feasible mechanism design, in: 14th Innovations in Theoretical Computer Science Conference (ITCS), 2023, pp. 93:1–93:22.

[48] S. Shalev-Shwartz, et al., Online learning and online convex optimization, Found. Trends Mach. Learn. 4 (2012) 107–194.

[49] Y. Singer, Budget feasible mechanisms, in: IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2010, pp. 765–774.

[50] Y. Singer, How to win friends and influence people, truthfully: influence maximization mechanisms for social networks, in: ACM International Conference on Web Search and Data Mining (WSDM), 2012, pp. 733–742.

[51] Y. Singer, M. Mittal, Pricing mechanisms for crowdsourcing markets, in: International World Wide Web Conferences (WWW), 2013, pp. 1157–1166.

[52] A. Singla, S. Tschiatschek, A. Krause, Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), 2016, pp. 2037–2043.

[53] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: International Conference on Management of Data (SIGMOD), 2014, pp. 75–86.

[54] S. Tschiatschek, R.K. Iyer, H. Wei, J.A. Bilmes, Learning mixtures of submodular functions for image collection summarization, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 1413–1421.

[55] X. Tu, K. Zhu, N.C. Luong, D. Niyato, Y. Zhang, J. Li, Incentive mechanisms for federated learning: from economic and game theoretic perspective, IEEE Trans. Cogn. Commun. Netw. 8 (3) (2022) 1566–1593.
[56] A. Young, The Practice Manual: The Ultimate Guide for Golfers, CreateSpace Independent Publishing Platform, 2015.
[57] J. Zhang, Y. Wu, R. Pan, Incentive mechanism for horizontal federated learning based on reputation and reverse auction, in: Proceedings of the Web Conference (WWW), 2021, pp. 947–956.
[58] D. Zhao, X.Y. Li, H. Ma, How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint, in: IEEE Conference on Computer Communications (INFOCOM), 2014, pp. 1213–1221.
[59] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, G. Chen, A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing, IEEE Trans. Mob. Comput. 16 (2017) 2392–2407.