Improved Approximation Algorithms and Lower Bounds for Search-Diversification Problems

Amir Abboud ✉
Weizmann Institute of Science, Rehovot, Israel

Vincent Cohen-Addad ✉
Google Research, Zürich, Switzerland

Euiwoong Lee ✉
University of Michigan, Ann Arbor, MI, USA

Pasin Manurangsi ✉
Google Research, Mountain View, CA, USA

Abstract

We study several questions related to diversifying search results. We give improved approximation algorithms in each of the following problems, together with some lower bounds.

1. We give a polynomial-time approximation scheme (PTAS) for a diversified search ranking problem [9] whose objective is to minimizes the discounted cumulative gain. Our PTAS runs in time $n^{2^{O(\log(1/\epsilon)/\epsilon)}} \cdot m^{O(1)}$ where n denotes the number of elements in the databases and m denotes the number of constraints. Complementing this result, we show that no PTAS can run in time $f(\epsilon) \cdot (nm)^{2^{o(1/\epsilon)}}$ assuming Gap-ETH and therefore our running time is nearly tight. Both our upper and lower bounds answer open questions from [9].
2. We next consider the Max-Sum Dispersion problem, whose objective is to select k out of n elements from a database that maximizes the dispersion, which is defined as the sum of the pairwise distances under a given metric. We give a quasipolynomial-time approximation scheme (QPTAS) for the problem which runs in time $n^{O_\epsilon(\log n)}$. This improves upon previously known polynomial-time algorithms with approximate ratios 0.5 [35, 16]. Furthermore, we observe that reductions from previous work rule out approximation schemes that run in $n^{\tilde{o}_\epsilon(\log n)}$ time assuming ETH.
3. Finally, we consider a generalization of Max-Sum Dispersion called Max-Sum Diversification. In addition to the sum of pairwise distance, the objective also includes another function f. For monotone submodular function f, we give a quasipolynomial-time algorithm with approximation ratio arbitrarily close to $(1-1/e)$. This improves upon the best polynomial-time algorithm which has approximation ratio 0.5 [16]. Furthermore, the $(1-1/e)$ factor is also tight as achieving better-than-$(1-1/e)$ approximation is NP-hard [26].

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation Algorithms, Complexity, Data Mining, Diversification

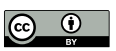
Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.7

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2203.01857 [2]

Funding *Amir Abboud*: Supported by an Alon scholarship and a research grant from the Center for New Scientists at the Weizmann Institute of Science.
Euiwoong Lee: Partially supported by a gift from Google.

Acknowledgements We are grateful to Karthik C.S. for insightful discussions, and to Badih Ghazi for encouraging us to work on the problems.



1 Introduction

A fundamental task in databases in general and in search engines in particular is the selection and ordering of the results to a given query. Suppose that we have already retrieved the set of appropriate answers S_q to a query q by a certain preliminary process. Which item from the (possibly huge) set S_q should be presented *first*? Which should be the first ten?

Besides the obvious approach of ranking the *most relevant* answers first, perhaps the second most important consideration is that the output set should satisfy certain *diversity* requirements. If a user searches for "Barcelona" it would be desirable that the first ten results contain a mix of items containing, e.g. general details of the city, tourist information, and news about the associated soccer team, even though the most relevant items in certain absolute terms may only pertain to the latter. There are various natural ways to formalize what makes a set of results diverse, and much research has gone into this *Search Diversification* topic in the past two and a half decades in various context (see e.g. [19, 3, 33, 15, 9, 39, 50, 16, 24, 12, 31, 46, 34, 1, 36, 25, 52]). Recently, there have also been extensive research efforts into algorithmic fairness (see e.g. a survey [47]). Some of these fairness notions (e.g. [21, 7]) are also closely related to diversity: a set of results that is not diverse enough (e.g. returning only pictures of members of one group when a user searches for "scientists") could be problematic in terms of fairness.

A well-known work on search diversification [19] suggests that a diverse set of results is one that satisfies the following: The k^{th} result in the list should maximize the sum[1] of: (1) the relevance to the query, and (2) the total distance to the first $k-1$ results in the list. The success of this natural notion of diversification may be attributed to the fact that it can be computed efficiently with a greedy algorithm. However, it may be a bit too simplistic and the objectives that real-world search engines seem to optimize for are actually closer to other, more complicated (to compute) notions of diversity that have been proposed in follow-up works (e.g. [9, 33, 16]).

The goal of this paper is to investigate the time complexity of computing these latter, more intricate definitions of the search diversification task. Since such problems are NP-Hard even for restricted settings, and since approximate solutions are typically acceptable in this context, our focus is on understanding their time vs. approximation trade-offs. Our results reduce the gaps in the literature, completely resolving the complexity of some of the most natural notions.

1.1 Diversified Search Ranking

The first problem we study is a diversified search ranking problem formulated by Bansal et al. [9]. Here we are given a collections $\mathcal{S}$ of subsets of $[n]$ and, for each $S \in \mathcal{S}$, a positive integer k_S. Our goal is to find a permutation $\pi : [n] \to [n]$ that minimizes the *discounted cumulative gain (DCG)* defined as

$$\mathrm{DCG}_{\mathcal{S},\mathbf{k}}(\pi) := \sum_{S \in \mathcal{S}} \frac{1}{\log(t_\pi(S)+1)}, \tag{1}$$

where $t_\pi(S)$ is defined as the earliest time the set S is covered k_S times, i.e. $\min\{i \in [n] | S \cap \pi([i])| \geq k_S\}$.

[1] To be more precise, it is a weighted average of the two terms.

This formulation relates to diversification by viewing the output π as the ranking of the documents to be shown, and each topic corresponds to a set S of documents related to that topic. With this interpretation, the DCG favors rankings that display "diverse topics as early in the ranking as possible". Bansal et al. [9] gave a polynomial-time approximation scheme (PTAS) for the problem in the special case that $k_S = 1$ for all $S \in \mathcal{S}$ with running time $n^{2^{O(\log(1/\epsilon)/\epsilon)}} m^{O(1)}$. On the other hand, for the case of general k_S's, they give a quasipolynomial-time approximation scheme with running time $n^{(\log\log n)^{O(1/\epsilon)}} m^{O(1)}$ and left as an open question whether a PTAS exists. We resolve this open question by giving a PTAS for the more general problem; the running time we obtain for this more general problem is similar to the running time obtained by Bansal et al.'s PTAS for the special case $k_S = 1$. We then show that this is indeed the best possible (under some complexity assumption).

▶ **Theorem 1.** *There is a randomized PTAS for maximizing DCG that runs in time* $n^{2^{O(\log(1/\epsilon)/\epsilon)}} \cdot m^{O(1)}$.

The above running time is doubly exponential in $1/\epsilon$, and Bansal et al. [9] asked whether this dependency is necessary even for the special case $k_S = 1$. We also answer this question by showing that the doubly exponential is necessary, assuming the Gap Exponential Time Hypothesis (Gap-ETH)[2]:

▶ **Theorem 2.** *Assuming Gap-ETH, for any function g, there is no PTAS for maximizing DCG that runs in time $g(\epsilon) \cdot (nm)^{2^{o(1/\epsilon)}}$. Moreover, this holds even when restricted to instances with $k_S = 1$ for all $S \in \mathcal{S}$.*

1.2 Max-Sum Dispersion

The second problem we consider is the so-called *Max-Sum Dispersion* problem where we are given a metric space (U, d) where $|U| = n$ and an integer $p \geq 2$. The goal is to select $S \subseteq U$ of size p that maximizes

$$\mathrm{Disp}(S) := \sum_{\{u,v\} \subseteq S} d(u, v).$$

Roughly speaking, if the metric determines how different the items are, then our goal is to pick items that are "as diverse as possible" according to the Disp objective.

The Max-Sum Dispersion problem is a classic problem that has been studied since the 80s [45, 38, 49, 35, 16]. Previous works have given 0.5-approximation algorithm for the problem in polynomial time [35, 16]. We observe that the known NP-hardness reduction, together with newer hardness of approximation results for the Densest k-Subgraph problem with perfect completeness, yields strong lower bounds for the problem. (Details are deferred to the full version [2].) For example, if we assume the Strongish Planted Clique Hypothesis [43], then no $(0.5 + \epsilon)$-approximation algorithm is possible in $n^{o(\log n)}$ time. In other words, to achieve an improvement over the known approximation ratio, the algorithm must run in $n^{\Omega(\log n)}$ time. Complementing this, we provide a quasipolynomial-time approximation scheme that runs in time $n^{O_\epsilon(\log n)}$:

▶ **Theorem 3.** *There is a QPTAS for Max-Sum Dispersion that runs in time* $n^{O(\log n/\epsilon^4)}$.

[2] Gap-ETH [23, 42] asserts that there is no $2^{o(n)}$-time algorithm to distinguish between a satisfiable n-variable 3SAT formula and one which is not even $(1 - \epsilon)$-satisfiable for some $\epsilon > 0$

1.3 Max-Sum Diversification

Finally, we consider a generalization of Max-Sum Dispersion where, in addition to the metric space (U, d), we are now also given a monotone set function f (which we can access via a value oracle) and the goal is to select a set $S \subseteq U$ of size p that maximizes

$$\mathrm{Div}(S) := \mathrm{Disp}(S) + f(S).$$

This problem is referred to as *Max-Sum Diversification*.

The Max-Sum Diversification problem is more expressive than Max-Sum Dispersion. For example, the value $f(S)$ in the objective may be used to encode how relevant the selected set S is to the given query, in addition to the diversity objective expressed by $\mathrm{Disp}(S)$.

Borodin et al. [16] gave a 0.5-approximation algorithm for the problem when f is a monotone submodular function. Since Max-Sum Diversification is a generalization of Max-Sum Dispersion, our aforementioned lower bounds also imply that improving on this 0.5 factor requires at least $n^{\Omega(\log n)}$ time. Furthermore, submodular Max-Sum Diversification is also a generalization of maximizing monotone submodular function subject to a cardinality constraint. For this problem, an $(1 - 1/e)$-approximation algorithm is known and it is also known that achieving better than this ratio is NP-hard [26]. Therefore, it is impossible to achieve a better-than-$(1 - 1/e)$ approximation even in (randomized) quasi-polynomial time, assuming NP $\nsubseteq RTIME(n^{O(\log n)})$. Here we manage to provide such a tight quasi-polynomial time approximation algorithm:

▶ **Theorem 4.** *For any $\epsilon > 0$, there exists a randomized $n^{O(\log n/\epsilon^4)}$-time $(1 - 1/e - \epsilon)$-approximation algorithm for submodular Max-Sum Diversification.*

We remark that an interesting special case of submodular Max-Sum Diversification is when f is linear, i.e. $f(S) = \sum_{u \in S} f(u)$. In this case, Gollapudi and Sharma [33] provided an approximation-preserving reduction from the problem to the Max-Sum Dispersion. Therefore, our QPTAS for the latter (Theorem 3) also yields a QPTAS for this special case of Max-Sum Dispersion.

2 Preliminaries

For a natural number n, we use $[n]$ to denote $\{1, \ldots, n\}$. We say that a randomized algorithm for a maximization problem is an α-approximation if the expected objective of the output solution is at least α times the optimum; note that we can easily get a high-probability bound with approximation guarantee arbitrarily close to α by repeating the algorithm multiple times and pick the best solution.

2.1 Concentration Inequalities

For our randomized approximation algorithms, we will need some standard concentration inequalities. First, we will use the following version of Chernoff bound which gives a tail bound on the sum of i.i.d. random variables. (See e.g. [44] for a proof.)

▶ **Lemma 5** (Chernoff bound). *Let $X_1, \ldots, X_r \in [0, 1]$ be independent random variables, $S := X_1 + \cdots + X_r$ and $\mu := \mathbb{E}[S]$. Then, for any $\delta \in [0, 1]$, we have*

$$\Pr[|S - \mu| > \delta\mu] \leq 2 \exp\left(-\frac{\delta^2 \mu}{3}\right).$$

Furthermore, for any $\delta \geq 0$, we have

$$\Pr[S > (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right).$$

It will also be convenient to have a concentration of sums of random variables that are drawn without replacement from a given set. For this, we will use (a without-replacement version of) the Hoeffding's inequality, stated below. (See e.g. [10].)

▶ **Lemma 6** (Hoeffding's inequality). *Let $X_1, \ldots, X_r$ be random variables drawn without replacement from a multiset $\mathcal{X} \subseteq [0,1]$, $A := \frac{1}{r}(X_1 + \cdots + X_r)$ and $\mu := \mathbb{E}[A]$. Then, for any $\delta \in [0,1]$, we have*

$$\Pr[|A - \mu| > \delta] \leq 2\exp\left(-2\delta^2 r\right).$$

2.2 Densest k-Subgraph

For both our Max-Sum Dispersion and Max-Sum Diversification problems, we will use as a subroutine algorithms for (variants of) the *Densest k-Subgraph (DkS)* problem. In DkS, we are given a set V of nodes, weights $w : \binom{V}{2} \to [0,1]$ and an integer k, the goal is to find a subset $T \subseteq V$ with $|T| = k$ that maximizes $\mathrm{Den}(T) := \frac{1}{|T|(|T|-1)/2}\sum_{\{u,v\}\subseteq T} w(\{u,v\})$. An *additive QPTAS* is an algorithm running in quasipolynomial time for any fixed $\epsilon > 0$ such that its output T satisfies $\mathrm{Den}(T) \geq \mathrm{OPT} - \epsilon$; Barman [11] gave such an algorithm for DkS.

We will in fact use a slightly generalized version of the problem where a subset $I \subseteq V$ of vertices is given as an input and these vertices must be picked in the solution T (i.e. $I \subseteq T$). To avoid cumbersomeness, we also refer to this generalized version as DkS. It is not hard to see[3] that Barman's algorithm [11] extends easily to this setting:

▶ **Theorem 7.** *There is an additive QPTAS for DkS that runs in time $n^{O(\log n/\epsilon^2)}$.*

DkS is a classic problem in approximation algorithms literature, and many approximation algorithms [30, 51, 29, 28, 6, 32, 13, 11] and hardness results [27, 37, 48, 4, 14, 17, 40, 20] have been proved over the years. Most of these works focus on *multiplicative* approximation; the best known polynomial-time algorithm in this setting has an approximation ratio of $n^{1/4+\epsilon}$ for any constant $\epsilon > 0$ [13] and there are evidences that achieving subpolynomial ratio in polynomial time is unlikely [40, 14, 22]. As for *additive* approximation, it is known that an approximation scheme that runs in time $n^{\tilde{o}(\log n)}$ would break the exponential time hypothesis (ETH) [17]; therefore, the running time in Theorem 7 (in terms of n) is tight up to $\mathrm{poly}\log\log n$ factor in the exponent. We provide additional discussions on related results in the full version [2].

2.3 Submodular Maximization over a Matroid Constraint

For our approximation algorithm for Max-Sum Diversification, we will also need an approximation algorithm for *monotone submodular maximization under a matroid constraint*. In this problem, we are given a monotone submodular set function $f : 2^X \to \mathbb{R}_{\geq 0}$ over a ground set X together with a matroid $\mathcal{M} = (X, \mathcal{I})$. The function f is given via a value oracle and $\mathcal{M}$ can be accessed via a membership oracle (which answers questions of the form "does

[3] In fact, in Section 5.1, we also give a more general algorithm than the one stated in Theorem 7 which can also handle an additional monotone submodular function.

S belong to $\mathcal{I}$?"). The goal is to find $S \in \mathcal{I}$ that maximizes $f(S)$. Călinescu et al. gave a randomized algorithm with approximation ratio $(1 - 1/e)$ for the problem, which we will use in our algorithm.

▶ **Theorem 8** ([18]). *There exists a randomized polynomial-time $(1 - 1/e)$-approximation algorithm for maximizing a montone submodular function over a matroid constraint.*

3 Diversified Search Ranking

In this section, we consider the diversified search ranking question as proposed in [9] and prove our upper and lower bounds (Theorems 1 and 2).

3.1 Polynomial-time Approximation Scheme

We will start by presenting our PTAS. At a high-level, our PTAS is similar to that of Bansal et al.'s: our algorithm use bruteforce to try every possible values of $\pi(1), \ldots, \pi(\exp(\tilde{O}(1/\epsilon)))$. Once these are fixed, we solve the remaining problem using linear programming (LP). We use the same LP as Bansal et al., except with a slightly more refined rounding procedure, which allows us to achieve a better approximation guarantee.

The remainder of this section is organized as follows. In Section 3.1.1, we present our LP rounding algorithm and its guarantees. Then, we show how to use it to yield our PTAS in Section 3.1.2.

3.1.1 Improved LP Rounding

For convenience in the analysis below, let us also define a more generic objective function where $\frac{1}{\log(t_\pi(S))+1}$ in Equation (1) can be replaced by any non-increasing function $f : [n] \to (0, 1]$:

$$\mathrm{DCG}^f_{\mathcal{S},\mathbf{k}}(\pi) := \sum_{S \in \mathcal{S}} f(t_\pi(S)).$$

The main result of this subsection is the following polynomial time LP rounding algorithm for the above general version of DCG:

▶ **Lemma 9.** *There exists an absolute constant C such that for any $\alpha \in (0, 0.5)$ the following holds: there is a polynomial-time algorithm that computes a ranking with expected DCG at least $(1 - \alpha) \cdot \tau_{f,\alpha}$ times that of the optimum where*

$$\tau_{f,\alpha} := \min_{t \in [n]} \frac{f\left(\frac{C \log(1/\alpha)}{\alpha} \cdot \frac{t}{f(t)}\right)}{f(t)}.$$

Informally speaking, the term $\tau_{f,\alpha}$ somewhat determines "how fast f increases". In the next section, once we fix the first u elements of the ranking, f will become $f(t) := 1/\log(t+u)$ which is "slowly growing" when u is sufficiently large. This allows us to ensure that the guarantee in Lemma 9 yields an $(1 - O(\epsilon))$-approximation as desired.

3.1.1.1 LP Formulation

To prove Lemma 9, we use the same knapsack constraint-enhanced LP as in [9], stated below. Note that the number of knapsack constraints can be super-polynomial. However, it is known that such an LP can be solved in polynomial time; see e.g. [8, Section 3.1] for more detail.

$$\text{Maximize} \qquad \sum_{S \in \mathcal{S}} \sum_{t \in [n]} (y_{S,t} - y_{S,t-1}) \cdot f(t)$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{e\in[n]} x_{e,t} = 1 && \forall t \in [n] \\
& \sum_{t\in[n]} x_{e,t} = 1 && \forall e \in [n] \\
& \sum_{e\in S\subseteq A} \sum_{t'<t} x_{e,t'} \geq (k_S - |A|)\cdot y_{S,t} && \forall S \in \mathcal{S}, A \subseteq S, t \in [n] \\
& y_{S,t} \geq y_{S,t-1} && \forall S \in \mathcal{S}, t \in \{2,\dots,n\} \\
& x_{e,t}, y_{S,t} \in [0,1] && \forall e,t \in [n], S \in \mathcal{S}.
\end{aligned}$$

3.1.1.2 Rounding Algorithm

Let $\gamma \in (0, 0.1)$ be a parameter to be chosen later. Our rounding algorithm works as follows:

1. $\pi \leftarrow \emptyset$
2. For $i = 1, \dots, \lceil \log n \rceil$ do:
   a. Let $t_i = \min\{n, 2^i\}$.
   b. Let $z_{e,i} = \sum_{t\leq t_i} x^*_{e,t}$ and $p_{e,i} = \min\{1, \frac{z_{e,i}}{\gamma \cdot f(t_i)}\}$ for all $e \in [n]$.
   c. Let A_i be the set such that $e \in [n]$ is independently included w.p. $p_{e,i}$.

Finally, our permutation π is defined by adding elements from $A_1, \dots, A_{\lceil \log n \rceil}$ in that order, where the order within each A_i can be arbitrary and we do not add an element if it already appears in the permutation.

Once again, we remark that our algorithm closely follows that of [9], except that Bansal et al. simply chose their $p_{e,i}$ to be $\min\{1, O(\log^2 n) \cdot z_{e,i}\}$, whereas our $p_{e,i}$ is a more delicate $\min\{1, \frac{z_{e,i}}{\gamma \cdot f(t_i)}\}$. This allows our analysis below to produce a better approximation ratio.

3.1.1.3 Analysis

We will now proceed to analyze our proposed randomized rounding procedure. Let $\eta \in (0, 0.1)$ be a parameter to be chosen later, and let $(\mathbf{x}^*, \mathbf{y}^*)$ denote an optimal solution to the LP. For each S, let $t^*(S)$ be the largest positive integer t^* such that

$$y^*_{S,t^*-1} \leq \eta \cdot f(t^*). \tag{2}$$

We start with the following lemma, which is a refinement of [9, Lemma 1].

▶ **Lemma 10.** $\text{OPT} \leq (1+\eta) \cdot \sum_{S\in\mathcal{S}} f(t^*(S))$.

Proof. We have

$$\begin{aligned}
\text{OPT} &\leq \sum_{S\in\mathcal{S}} \sum_{t\in[n]} (y^*_{S,t} - y^*_{S,t-1}) \cdot f(t) \\
&= \sum_{S\in\mathcal{S}} \left(\sum_{t=1}^{t^*(S)-1} (y^*_{S,t} - y^*_{S,t-1}) \cdot f(t) + \sum_{t=t^*(S)}^{n} (y^*_{S,t} - y^*_{S,t-1}) \cdot f(t) \right) \\
&\leq \sum_{S\in\mathcal{S}} \left(\sum_{t=1}^{t^*(S)-1} (y^*_{S,t} - y^*_{S,t-1}) + \sum_{t=t^*(S)}^{n} (y^*_{S,t} - y^*_{S,t-1}) \cdot f(t^*(S)) \right) \\
&\leq \sum_{S\in\mathcal{S}} \left(y^*_{S,t^*(S)-1} + f(t^*(S)) \right) \\
&\overset{(2)}{\leq} \sum_{S\in\mathcal{S}} (1+\eta) \cdot f(t^*(S)).
\end{aligned}$$

◀

Next, we show via standard concentration inequalities that $|A_i|$'s has small sizes with a large probability.

▶ **Lemma 11.** *With probability* $1 - 2\exp\left(-\frac{1}{3\gamma}\right)$*, we have* $|A_i| \leq \frac{2t_i}{\gamma f(t^*)}$ *for all* $i \in [\lceil \log n \rceil]$.

Proof. Notice that $\sum_{e\in[n]} p_{e,i} \leq \frac{\sum_{e\in[n]} z_{e,i}}{\gamma f(t_i)} = \frac{t_i}{\gamma f(t_i)}$. As a result, by Chernoff bound (Lemma 5), we have

$$\Pr\left[|A_i| > \frac{2t_i}{\gamma f(t^*)}\right] \leq \exp\left(-\frac{t_i}{3\gamma f(t^*)}\right) \leq \exp\left(-\frac{t_i}{3\gamma}\right).$$

By union bound, we thus have $|A_i| \leq \frac{2t_i}{\gamma f(t^*)}$ for all $i \in [\lceil \log n \rceil]$ with probability at least

$$1 - \sum_{i\in[\lceil \log n \rceil]} \exp\left(-\frac{t_i}{3\gamma}\right) \leq 1 - 2\exp\left(-\frac{1}{3\gamma}\right). \qquad \blacktriangleleft$$

Let $i^*(S)$ denote the smallest i such that $t_i \geq t^*(S)$. We now bound the probability that S is covered (k_S times) by the end of the $i^*(S)$-th iteration of the algorithm. Our bound is stated below. We note that our bound here is not with high probability, unlike that of the analysis of [9] which yields a bound of $1 - o(1/n)$. We observe here that such a strong bound is not necessary for the analysis because we are working with a maximization problem and therefore such a high probability bound is not necessary to get a bound on the expectation of the DCG.

▶ **Lemma 12.** *Assume that* $\eta \geq 2\gamma$*. For each* $S \in \mathcal{S}$*, we have* $t_\pi(S) \leq |A_1| + \cdots + |A_{i^*(S)}|$ *with probability* $1 - \exp\left(\frac{\eta}{8\gamma}\right)$.

Proof. It suffices to show that at least k_S elements of S are selected in $A_{i^*(S)}$. Let S_g denote the set of elements $e \in S$ for which $p_{e,i^*(S)} = 1$. If $|S_g| \geq k_S$, then we are done. Otherwise, from knapsack constraint, we have

$$\sum_{e\in S\setminus S_g} z_{e,i^*(S)} \geq (k_S - |S_g|) y^*_{S,t_{i^*(S)}} \geq (k_S - |S_g|) y^*_{S,t^*(S)} \geq \eta \cdot f(t^*(S)) \cdot (k_S - |S_g|)$$
$$\geq \eta \cdot f(t_{i^*(S)}) \cdot (k_S - |S_g|),$$

where the third inequality follows from our choice of $t^*(S)$. This implies that

$$\sum_{e\in S\setminus S_g} p_{e,i^*(S)} \geq \eta/\gamma \cdot (k_S - |S_g|).$$

Recall that $\eta/\gamma \geq 2$. This means that the probability that at least k_S elements of S are selected in $A_{i^*(S)}$ is at least

$$1 - \Pr[|(S \setminus S_g) \cap A_{i^*(S)}| \leq 0.5\eta/\gamma \cdot (k_S - |S_g|)]$$
$$\leq 1 - \exp\left(-\frac{1}{8} \cdot \eta/\gamma \cdot (k_S - |S_g|)\right)$$
$$\leq 1 - \exp\left(-\frac{\eta}{8\gamma}\right),$$

where the first inequality follows from the Chernoff bound. ◀

Applying the union bound to the two previous lemmas, we immediately arrive at the following:

▶ **Lemma 13.** *Assume that $\eta \geq 2\gamma$. For all $S \in \mathcal{S}$, we have*

$$\mathbb{E}_\pi[f(t_\pi(S))] \geq \left(1 - 2\exp\left(-\frac{1}{3\gamma}\right) - \exp\left(\frac{\eta}{8\gamma}\right)\right) \cdot f\left(\frac{8t^*(S)}{\gamma f(t^*(S))}\right)$$

Finally, combining Lemmas 10 and 13 and selecting $\eta = 2\alpha, \gamma = O(\eta/\log(1/\eta))$ yields Lemma 9.

3.1.2 From LP Rounding to PTAS

As stated earlier, we may now use bruteforce to try all possible values of the first few elements in the ranking and then use our LP rounding to arrive at the PTAS:

Proof of Theorem 1. For any $\epsilon < 0.1$, we use bruteforce for the first $u = (4C/\epsilon)^{100/\epsilon}$ elements and then use Lemma 9 on the remaining instance but with $f(t) := \frac{1}{\log(t+u)}$. The expected approximation ratio we have is at least

$$\begin{aligned}
&(1-0.5\epsilon)\cdot\tau_{f,0.5\epsilon}\\
&\geq (1-0.5\epsilon)\cdot\min_{t\in[n]} f\left(\frac{4C\log(1/\epsilon)}{\epsilon}\cdot\frac{t}{f(t)}\right)/f(t)\\
&= (1-0.5\epsilon)\cdot\min_{t\in[n]}\frac{\log(t+u)}{\log\left(\frac{4C\log(1/\epsilon)}{\epsilon}\cdot\frac{t}{f(t)}+u\right)}\\
&\geq (1-0.5\epsilon)\cdot\min_{t\in[n]}\frac{\log(t+u)}{\log\left(\frac{4C\log(1/\epsilon)}{\epsilon}\cdot(t+u)\log(t+u)\right)}\\
&= (1-0.5\epsilon)\cdot\min_{t\in[n]}\frac{1}{1+\frac{\log\left(\frac{4C\log(1/\epsilon)}{\epsilon}\right)}{\log(t+u)}+\frac{\log\log(t+u)}{\log(t+u)}}\\
&= (1-0.5\epsilon)\cdot\frac{1}{1+\frac{\log\left(\frac{4C\log(1/\epsilon)}{\epsilon}\right)}{\log(u)}+\frac{\log\log(u)}{\log(u)}}\\
&\geq (1-0.5\epsilon)\cdot\frac{1}{1+0.1\epsilon+0.1\epsilon}\\
&\geq 1-\epsilon,
\end{aligned}$$

as desired. ◀

3.2 Running Time Lower Bound

To prove our running time lower bound, we will reduce from the *Maximum k-Coverage* problem. Recall that in Maximum k-Coverage, we are given a set $\mathcal{T} \subseteq [M]$ and an integer k; the goal is to find $T_1^*, \cdots T_k^* \in \mathcal{T}$ that maximizes $|T_1^* \cup \cdots \cup T_k^*|$. We write $\mathrm{Cov}(\mathcal{T}, k)$ to denote this optimum. Furthermore, we say that a Maximum k-Coverage is *regular* if $|T| = M/k$ for all $T \in \mathcal{T}$. Finally, we use N to denote $|\mathcal{T}| \cdot M$ which upper bound the "size" of the problem.

Manurangsi [41] showed the following lower bound for this problem:

▶ **Theorem 14** ([41]). *Assuming the Gap Exponential Time Hypothesis (Gap-ETH), for any constant $\delta > 0$, there is no $N^{o(k)}$-time algorithm that can, given a regular instance $(\mathcal{T}, k)$ distinguish between the following two cases:*

- *(YES)* $\mathrm{Cov}(\mathcal{T}, k) \geq M$.
- *(NO)* $\mathrm{Cov}(\mathcal{T}, k) \leq (1 - 1/e + \delta)M$.

Proof of Theorem 2. Fix $\delta = 0.1$. We reduce from the Maximum k-Coverage problem. Suppose that $(\mathcal{T}, k)$ is a regular Maximum k-Coverage instance; we assume w.l.o.g. that k is divisible by 10.

We construct the instance $(\mathcal{S}, \{k_S\}_{S\in\mathcal{S}})$ of the DCG maximization as follows:

- Let $n = |\mathcal{T}|$ where we associate each $j \in [n]$ with $T_j \in \mathcal{T}$.
- Let $\mathcal{S} = \{S_1, \ldots, S_M\}$ where $S_i = \{j \in [n] \mid i \in T_j\}$.
- Let $k_S = 1$ for all $S \in \mathcal{S}$.

In the YES case, let $T_{j_1}, \ldots, T_{j_k}$ be such that $|T_{j_1} \cup \cdots \cup T_{j_k}| = M$. Let $\pi^* : [n] \to [n]$ be any permutation such that $\pi^*(\ell) = j_\ell$ for all $\ell \in [k]$. From regularity of $(\mathcal{T}, k)$, there are exactly $q := M/k$ sets $S \in \mathcal{S}$ such that $t_{\pi^*}(S) = i$. Therefore, we have

$$\mathrm{DCG}_{\mathcal{S},\mathbf{k}}(\pi^*) = \sum_{i\in[k]} \frac{M}{k} \cdot \frac{1}{\log(i+1)}.$$

Let OPT^* denote the RHS quantity. Notice that

$$\mathrm{OPT}^* \le \frac{M}{\log(k+1)}. \tag{3}$$

In the NO case, consider any permutation $\pi : [n] \to [n]$. Let t_i denote the i-th smallest value in the multiset $\{t_\pi(S)\}_{S\in\mathcal{S}}$. Regularity of $(\mathcal{T}, k)$ implies that

$$t_i \ge t_{i-q} + 1 \tag{4}$$

for all $i > q$. This in turn implies that

$$t_i \ge \lceil i/q \rceil . \tag{5}$$

Furthermore, $\mathrm{Cov}(\mathcal{T}, k) \le (1 - 1/e - \delta)M \le 0.8M$ implies that

$$t_{0.8M} > k.$$

Furthermore, applying (4) to the above, we have

$$t_{0.9M} \ge t_{0.8M} + \left\lfloor \frac{0.1M}{q} \right\rfloor = k + 0.1k = 1.1k. \tag{6}$$

With the above notion, we may write $\mathrm{DCG}_{\mathcal{S},\mathbf{k}}(\pi) - \mathrm{OPT}^*$ as

$$\begin{aligned}
\mathrm{DCG}_{\mathcal{S},\mathbf{k}}(\pi) - \mathrm{OPT}^* &= \sum_{i=1}^{M} \frac{1}{\log(t_i+1)} - \sum_{i=1}^{M} \frac{1}{\log(\lceil i/q \rceil + 1)} \\
&\overset{(5)}{\ge} \sum_{i=0.9M}^{M} \left(\frac{1}{\log(t_i+1)} - \frac{1}{\log(\lceil i/q \rceil + 1)} \right) \\
&\overset{(6)}{\ge} \sum_{i=0.9M}^{M} \left(\frac{1}{\log(1.1k+1)} - \frac{1}{\log(\lceil i/q \rceil + 1)} \right) \\
&\ge \sum_{i=0.9M}^{M} \left(\frac{1}{\log(1.1k+1)} - \frac{1}{\log(k+1)} \right) \\
&= 0.1M \cdot \left(\frac{1}{\log(1.1k+1)} - \frac{1}{\log(k+1)} \right) \\
&= \Theta\left(\frac{M}{\log^2 k} \right).
\end{aligned}$$

Finally, observe also that

$$\mathrm{OPT}^* = \frac{M}{k} \cdot \sum_{i \in [k]} \frac{1}{\log(i+1)} = \frac{M}{k} \Theta\left(\frac{k}{\log k}\right) = \Theta\left(\frac{M}{\log k}\right).$$

Combining the above two inequalities, we have

$$\mathrm{DCG}_{\mathcal{S},\mathbf{k}}(\pi) \geq \left(1 + \Theta\left(\frac{1}{\log k}\right)\right) \cdot \mathrm{OPT}^*.$$

Now, suppose that there is a PTAS for maximizing DCG that runs in time $f(\epsilon) \cdot (nm)^{2^{o(1/\epsilon)}}$. If we run the algorithm with $\epsilon = \gamma / \log k$ where $\gamma > 0$ is sufficiently small constant, then we can distinguish between the YES case and the NO case in time $f(1/\log k) \cdot (nm)^{2^{o(\log k)}} \leq f(1/\log k) \cdot (nm)^{o(k)} = g(k) \cdot N^{o(k)}$ which, from Theorem 14, violates Gap-ETH. ◀

4 Max-Sum Dispersion

In this section, we provide a QPTAS for Max-Sum Dispersion (Theorem 3).

As alluded to earlier, our algorithm will reduce to the Densest k-Subgraph (DkS) problem, for which an *additive* QPTAS is known [11]. Notice here that DkS is a generalization of the Max-Dispersion problem because we may simply set $V = U, k = p$ and $w(\{u, v\}) = d(u, v)/D$ where $D := \max_{u,v} d(u, v)$ denote the diameter of the metric space. Note however that we cannot apply Theorem 7 yet because the QPTAS in that theorem offers an *additive* guarantee. E.g. if the optimum is $o(1)$, then the QPTAS will not yield anything at all unless we set $\epsilon = o(1)$, which then gives a running time $n^{\omega(\log n)}$. This example can happen when e.g. there is a single pair u, v that are very far away and then all the other pairs are close to u.

Our main technical contribution is to give a simple structural lemma that allows us to avoid such a scenario. Essentially speaking, it allows us to pick a vertex and selects all vertices that are "too far away" from it. Once this is done, the remaining instance can be reduced to DkS without encountering the "small optimum" issue described in the previous paragraph.

4.1 A Structural Lemma

Henceforth, we write $\mathrm{Disp}(S, T)$ to denote $\sum_{u \in S, v \in T} d(u, v)$ and $\mathrm{Disp}(u, T)$ as a shorthand for $\mathrm{Disp}(\{u\}, T)$. Furthermore, we use $\mathcal{B}(u, D)$ to denote $\{z \in U \mid d(z, u) \leq D\}$ and let $\overline{\mathcal{B}(u, D)} := U \setminus \mathcal{B}(u, D)$.

We now formalize our structural lemma. It gives a lower bound on the objective based on a vertex in the optimal solution and another vertex *not* in the optimal solution. Later on, by guessing these two vertices, we can reduce to DkS while avoiding the "small optimum" issue.

▶ **Lemma 15.** *Let S^{OPT} be any optimal solution of Max-Sum Dispersion and let $u^{\min}$ be the vertex in S^{OPT} that minimizes $\mathrm{Disp}(u^{\min}, S^{\mathrm{OPT}})$. Furthermore, let v be any vertex* not *in S^{OPT} and let $\Delta = d(u^{\min}, v)$. Then, we have*

$$\mathrm{Disp}(S^{\mathrm{OPT}}) \geq \frac{p(p-1)\Delta}{16}.$$

Proof of Lemma 15. Let $S^{\mathrm{OPT}}_{\mathrm{close}} := S^{\mathrm{OPT}} \cap \mathcal{B}(u^{\min}, 0.5\Delta)$. Consider two cases, based on the size of $S^{\mathrm{OPT}}_{\mathrm{close}}$:

- Case I: $|S^{\mathrm{OPT}}_{\mathrm{close}}| \leq p/2$. In this case, we have

$$\mathrm{Disp}(u^{\min}, S^{\mathrm{OPT}}) \geq \mathrm{Disp}(u^{\min}, S^{\mathrm{OPT}} \setminus S^{\mathrm{OPT}}_{\mathrm{close}}) \geq (p/2)(\Delta/2) = \Delta p/4.$$

  Furthermore, by our definition of $u^{\min}$, we have

$$\mathrm{Disp}(S^{\mathrm{OPT}}) = \frac{1}{2}\sum_{u \in S} \mathrm{Disp}(u, S^{\mathrm{OPT}}) \geq \frac{p}{2}\,\mathrm{Disp}(u^{\min}, S^{\mathrm{OPT}}).$$

  Combining the two inequalities, we have $\mathrm{Disp}(S^{\mathrm{OPT}}) \geq p^2\Delta/8$.
- Case II: $|S^{\mathrm{OPT}}_{\mathrm{close}}| > p/2$. In this case, since S^{OPT} is an optimal solution, replacing any $z \in S^{\mathrm{OPT}}_{\mathrm{close}}$ with v must not increase the solution value, i.e.

$$\begin{aligned}\mathrm{Disp}(z, S^{\mathrm{OPT}}) &\geq \mathrm{Disp}(v, S^{\mathrm{OPT}} \setminus \{z\}) \\ &\geq \mathrm{Disp}(v, S^{\mathrm{OPT}}_{\mathrm{close}} \setminus \{z\}) \\ &\geq ((p-1)/2)(0.5\Delta),\end{aligned}$$

  where the second inequality uses the fact that for any $z' \in S^{\mathrm{OPT}}_{\mathrm{close}}$ we have $d(v, z') \geq d(u, v) - d(u, z') \geq \Delta - 0.5\Delta$. From this, we once again have

$$\begin{aligned}\mathrm{Disp}(S^{\mathrm{OPT}}) = \frac{1}{2}\sum_{u \in S} \mathrm{Disp}(u, S^{\mathrm{OPT}}) \geq \frac{1}{2}\sum_{z \in S^{\mathrm{OPT}}_{\mathrm{close}}} \mathrm{Disp}(z, S^{\mathrm{OPT}}) &\geq |S^{\mathrm{OPT}}_{\mathrm{close}}| \cdot \frac{(p-1)\Delta}{8} \\ &> \frac{p(p-1)}{16\Delta},\end{aligned}$$

  where the last inequality follows from our assumption of this case. ◀

4.2 QPTAS for Max-Sum Dispersion

We now present our QPTAS, which simply guesses $u^{\min}$ and $v = \mathrm{argmax}_{z \notin S^{\mathrm{OPT}}} d(z, u)$ and then reduces the problem to DkS. By definition of v, if we let $\Delta = d(u, v)$, every point outside $\mathcal{B}(u^{\min}, \Delta)$ must be in S^{OPT}. The actual reduction to DkS is slightly more complicated than that described at the beginning of this section. Specifically, among points $\overline{\mathcal{B}(u^{\min}, \Delta)}$ that surely belong to S^{OPT}, we ignore all points outside $\mathcal{B}(u^{\min}, 20\Delta/\epsilon)$ (i.e., they do not appear in the DkS instance) and we let $\mathcal{B}(u^{\min}, 20\Delta/\epsilon) \setminus \mathcal{B}(u^{\min}, \Delta)$ be the "must pick" part. Ignoring the former can be done because the contribution to the objective from those points can be approximated to within $(1 \pm O(\epsilon))$ regardless of the points picked in the ball $\mathcal{B}(u^{\min}, \Delta)$. This is not true for the latter, which means that we need to include them in our DkS instance.

Proof of Theorem 3. Our algorithm works as follows:

1. For every distinct $u, v \in U$ do:
   a. Let $\Delta := d(u, v)$ and $\Delta^* = 20\Delta/\epsilon$.
   b. If $|\overline{\mathcal{B}(u, \Delta)}| \geq p$, then skip the following steps and continue to the next pair u, v.
   c. Otherwise, create a DkS instance where $V := \mathcal{B}(u, \Delta^*), I := V \setminus \mathcal{B}(u, \Delta)$, $k = p - |\overline{\mathcal{B}(u, \Delta^*)}|$ and w is defined as $w(\{y, z\}) := 0.5d(y, z)/\Delta^*$ for all $y, z \in V$.
   d. Use the additive QPTAS from Theorem 7 to solve the above instance to within an additive error of $\epsilon' := 0.00005\epsilon^2$. Let T be the solution found.
   e. Finally, let $S^{u,v} := T \cup \overline{\mathcal{B}(u, \Delta^*)}$.
2. Output the best solution among $S^{u,v}$ considered.

It is obvious that the running time is dominated by the running time of the QPTAS which takes $n^{O(\log n/(\epsilon')^2)} = n^{O(\log n/\epsilon^4)}$ as desired.

Next, we show that the algorithm indeed yields a $(1-\epsilon)$-approximation. To do this, let us consider $S^{\mathrm{OPT}}, u^{\min}$ as defined in Lemma 15, and let $u = u^{\min}, v := \operatorname{argmax}_{z \notin S^{\mathrm{OPT}}} d(u,z)$. Let T be the solution found by the DkS algorithm for this u, v and let $T' := T \setminus I$. We have

$$\begin{aligned}
&\mathrm{Disp}(S^{u,v}) \\
&= \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}) + \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}, T) + \mathrm{Disp}(T) \\
&= \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}) + \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}, I) + \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}, T') + \mathrm{Disp}(T). \qquad (7)
\end{aligned}$$

Similarly, letting $S := S^{\mathrm{OPT}} \cap \mathcal{B}(u,\Delta^*)$ and $S' := S^{\mathrm{OPT}} \setminus I$, we have

$$\begin{aligned}
&\mathrm{Disp}(S^{\mathrm{OPT}}) \\
&= \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}) + \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}, I) + \mathrm{Disp}(\overline{\mathcal{B}(u,\Delta^*)}, S') + \mathrm{Disp}(S). \qquad (8)
\end{aligned}$$

Now, observe from the definition of the DkS instance (for this u, v) that for any J such that $I \subseteq J \subseteq V$, we have

$$\mathrm{Den}(J) = \frac{1}{k(k-1)/2} \cdot \frac{0.5}{\Delta^*} \mathrm{Disp}(J).$$

The additive approximation guarantee from Theorem 7 implies that $\mathrm{Den}(T) \geq \mathrm{Den}(S) - \epsilon'$. Using the above equality, we can rewrite this guarantee as

$$\mathrm{Disp}(S) - \mathrm{Disp}(T) \leq \epsilon' \cdot \Delta^* \cdot k(k-1). \qquad (9)$$

Taking the difference between Equation (8) and Equation (7) and applying Equation (9), we have

$$\begin{aligned}
\mathrm{Disp}(S^{\mathrm{OPT}}) - \mathrm{Disp}(S^{u,v}) &\leq \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, S') - \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, T') + \epsilon' \cdot \Delta^* \cdot k(k-1). \\
(\text{Our choice of } \epsilon') &\leq \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, S') - \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, T') + 0.001\epsilon\Delta \cdot p(p-1) \\
(\text{Lemma 15}) &\leq \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, S') - \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, T') + 0.1\epsilon\,\mathrm{Disp}(S^{\mathrm{OPT}}).
\end{aligned}$$

Now, since $|S'| = |T'| \leq p$ and $S', T' \subseteq \mathcal{B}(z,\Delta)$, we have

$$\begin{aligned}
\mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, S') - \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, T') &\leq |\overline{\mathcal{B}(z,\Delta^*)}| \cdot |S'| \cdot ((\Delta^* + \Delta) - (\Delta^* - \Delta)) \\
&\leq 2|\overline{\mathcal{B}(z,\Delta^*)}| \cdot |S'| \cdot \Delta \\
(\text{Our choice of } \Delta^*) &\leq 0.1\epsilon \cdot |\overline{\mathcal{B}(z,\Delta^*)}| \cdot |S'| \cdot (\Delta^* - \Delta) \\
&\leq 0.1\epsilon \cdot \mathrm{Disp}(\overline{\mathcal{B}(z,\Delta^*)}, S') \\
&\leq 0.1\epsilon \cdot \mathrm{Disp}(S^{\mathrm{OPT}}).
\end{aligned}$$

Combining the above two inequalities, we get $\mathrm{Disp}(S^{u,v}) \geq (1 - 0.2\epsilon) \cdot \mathrm{Disp}(S^{\mathrm{OPT}})$, as desired. ◀

5 Max-Sum Diversification

In this section, we give our quasipolynomial-time approximation algorithm for the Max-Sum Diversification with approximation ratio arbitrarily close to $(1 - 1/e)$ (Theorem 4). In fact, we prove a slightly stronger version of the theorem where the approximation ratio for the dispersion part is arbitrarily close to 1 and that of the submodular part is arbitrarily close to $1 - 1/e$. This is stated more precisely below; note that this obviously implies Theorem 4.

▶ **Theorem 16.** *Let S^{OPT} be any optimal solution of Max-Sum Diversification. There exists a randomized $n^{O(\log n/\epsilon^4)}$-time algorithm that finds a p-size set S such that*

$$\mathbb{E}[\mathrm{Div}(S)] \geq (1-\epsilon)\,\mathrm{Disp}(S^{\mathrm{OPT}}) + (1-1/e-\epsilon)f(S^{\mathrm{OPT}}).$$

At a high-level, our algorithm for Max-Sum Diversification is very similar to that of Max-Sum Dispersion presented in the previous section. Specifically, we use a structural lemma (akin to Lemma 15) to reduce our problem to a variant of DkS. This variant of DkS additionally has a submodular function attached to it. Using techniques from DkS approximation literature, we give an algorithm for this problem by in turn reducing it to the submodular maximization problem over a partition matroid, for which we can appeal to Theorem 8.

5.1 Approximating Densest Subgraph and Submodular Function

We will start by giving an algorithm for the aforementioned extension of the DkS problem, which we call *Submodular DkS*:

▶ **Definition 17** (Submodular DkS)**.** *Given (V, I, w, k) (similar to DkS) together with a monotone submodular set function h on the ground set V (accessible via a value oracle), the goal is to find a size-k subset T where $I \subseteq T \subseteq V$ that maximizes $h(T) + \mathrm{Den}(T)$.*

We give a quasipolynomial-time algorithm with an approximation guarantee similar to QPTAS for the original DkS (i.e. Theorem 7) while also achiving arbritrarily close to $(1-1/e)$ approximation ratio for the submodular part of the objective:

▶ **Theorem 18.** *For any set T^{OPT} of size k such that $I \subseteq T^{\mathrm{OPT}} \subseteq V$, there is an $n^{O(\log n/\gamma^2)}$-time algorithm that output a size-k T such that $I \subseteq T \subseteq V$ and*

$$\mathbb{E}[h(T) + \mathrm{Den}(T)] \geq (1-1/e-\gamma)\,h(T^{\mathrm{OPT}}) + \mathrm{Den}(T^{\mathrm{OPT}}) - \gamma. \tag{10}$$

In order to facilitate the subsequent discussion and proof, it is useful to define additional notations. (Throughout, we view vectors as column vectors.)

- Let $\mathbf{W} \in \mathbb{R}^{V\times V}$ denote the vector where $\mathbf{W}_{u,v} = w(\{u,v\})$ for $u \neq v$ and $\mathbf{W}_{u,u} = 0$.
- For every $U \subseteq V$, let $\mathbf{1}(U) \in \mathbb{R}^V$ denote the indicator vector of U, i.e.

$$\mathbf{1}(U)_v = \begin{cases} 1 & \text{if } v \in U, \\ 0 & \text{otherwise.} \end{cases}$$

- For every $U \subseteq V$, let $\mathbf{w}(U) = \mathbf{W} \cdot \mathbf{1}(U) \in \mathbb{R}^V$.
- Finally, for every non-empty $U \subseteq V$, let $\overline{\mathbf{w}}(U) := \frac{1}{|U|} \cdot w(U)$ and $\overline{\mathbf{1}}(U) := \frac{1}{|U|} \cdot \mathbf{1}(U)$.

To understand our reduction, we must first describe the main ideas behind the QPTAS of [11]. (Some of these ideas also present in previous works, e.g. [5].) Let us assume for simplicity of presentation that $I = \emptyset$. Observe that DkS is, up to an appropriate scaling, equivalent to find a size-k subset T that maximizes $\overline{\mathbf{1}}(T)^T \cdot \mathbf{W} \cdot \overline{\mathbf{1}}(T) = \overline{\mathbf{1}}(T)^T \overline{\mathbf{w}}(T)$. The main observation is that, if we randomly pick a subset $U \subseteq T^{\mathrm{OPT}}$ of size $\Theta_\gamma(\log n)$, then with high probability $\|\overline{\mathbf{w}}(U) - \overline{\mathbf{w}}(T^{\mathrm{OPT}})\|_\infty \leq O(\gamma)$ and $|\overline{\mathbf{1}}(T^{\mathrm{OPT}})^T \overline{\mathbf{w}}(T^{\mathrm{OPT}}) - \overline{\mathbf{1}}(U)^T \overline{\mathbf{w}}(U)| < O(\gamma)$. Roughly speaking, [11] exploits this by "guessing" such a set U and then solves for T such that $\|\overline{\mathbf{w}}(U) - \overline{\mathbf{w}}(T)\|_\infty \leq O(\gamma)$ and $|\overline{\mathbf{1}}(T)^T \overline{\mathbf{w}}(U) - \overline{\mathbf{1}}(U)^T \overline{\mathbf{w}}(U)| < O(\gamma)$; note that (the fractional version of) this is a linear program and can be solved efficiently. [11] then shows that a fractional solution to such a linear program can be rounded to an actual size-k set without any loss in the objective function.

We further push this idea by noting that, if we randomly partition V into $V_1, \ldots, V_s$ part where $s = O_\gamma(k/\log n)$, then the intersections $U_i^{\mathrm{OPT}} := V_i \cap T^{\mathrm{OPT}}$ satisfy the two conditions from the previous paragraphs (for $T = U_i^{\mathrm{OPT}}$). Therefore, we may enumerate all sets $U_i \subseteq V_i$ of roughly expected size to construct a collection $\mathcal{P}_i$ of subsets that satisfies these two conditions. Our goal now become picking $U_1 \in \mathcal{P}_1, \ldots, U_s \in \mathcal{P}_s$ that maximizes $h(U_1 \cup \cdots \cup U_s)$. This is simply monotone submodular maximization subject to a partition matroid constraint and therefore we may appeal to Theorem 8. We remark here that the two conditions that all subsets in $\mathcal{P}_i$ satisfy already ensure that the DkS objective is close to optimum. The full proof of Theorem 18 is deferred to the full version [2].

5.2 From Submodular DkS to Max-Sum Diversification

Having provided an approximation algorithm for Submodular DkS, we can use it to approximate Max-Sum Diversification via a similar approach to the reduction from Max-Sum Dispersion to DkS in the previous section. In particular, we can prove a structural lemma for Max-Sum Diversification that is analogous to Lemma 15 for Max-Sum Dispersion. We can then use the reduction nearly identical to the one in the proof of Theorem 3 to arrive at Theorem 16. The full details are deferred to the full version [2].

6 Conclusion

In this work, we consider three problems related to diversification: DCG in diversified search ranking, Max-Sum Dispersion and Max-Sum Diversification. For DCG, we give a PTAS and prove a nearly matching running time lower bound. For Max-Sum Dispersion, we give a QPTAS and similarly provide evidence for nearly matching running time lower bounds. Finally, we give a quasi-polynomial time algorithm for Max-Sum Diversification that achieves an approximation ratio arbitrarily close to $(1 - 1/e)$, which is also tight given the $(1 - 1/e + o(1))$ factor NP-hardness of approximating Maximum k-Coverage [26]. Our algorithms for DCG and Max-Sum Diversification are randomized and it remains an interesting open question whether there are deterministic algorithms with similar running times and approximation ratios.

References

1 Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 32–40. ACM, 2013. doi:10.1145/2487575.2487636.

2 Amir Abboud, Vincent Cohen-Addad, Euiwoong Lee, and Pasin Manurangsi. Improved approximation algorithms and lower bounds for search-diversification problems. *CoRR*, abs/2203.01857, 2022. doi:10.48550/arXiv.2203.01857.

3 Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In Ricardo Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu, editors, *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, pages 5–14. ACM, 2009. doi:10.1145/1498759.1498766.

4 Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. Inapproximability of densest κ-subgraph from average case hardness, 2011.

5 Noga Alon, Troy Lee, Adi Shraibman, and Santosh S. Vempala. The approximate rank of a matrix and its algorithmic applications: approximate rank. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 675–684. ACM, 2013. doi:10.1145/2488608.2488694.

6 Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. Complexity of finding dense subgraphs. *Discrete Applied Mathematics*, 121(1–3):15–26, 2002. doi:10.1016/S0166-218X(01)00243-8.

7 Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 405–413. PMLR, 2019. URL: http://proceedings.mlr.press/v97/backurs19a.html.

8 Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *SODA*, pages 1539–1545, 2010. doi:10.1137/1.9781611973075.125.

9 Nikhil Bansal, Kamal Jain, Anna Kazeykina, and Joseph Naor. Approximation algorithms for diversified search ranking. In *ICALP*, pages 273–284, 2010. doi:10.1007/978-3-642-14162-1_23.

10 Rémi Bardenet and Odalric-Ambrym Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

11 Siddharth Barman. Approximating nash equilibria and dense subgraphs via an approximate version of carathéodory's theorem. *SIAM J. Comput.*, 47(3):960–981, 2018. doi:10.1137/15M1050574.

12 Julien Baste, Lars Jaffke, Tomás Masarík, Geevarghese Philip, and Günter Rote. FPT algorithms for diverse collections of hitting sets. *Algorithms*, 12(12):254, 2019. doi:10.3390/a12120254.

13 Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 201–210, 2010. doi:10.1145/1806689.1806718.

14 Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. Polynomial integrality gaps for strong SDP relaxations of densest k-subgraph. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 388–405, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2095116.2095150.

15 Aditya Bhaskara, Mehrdad Ghadiri, Vahab S. Mirrokni, and Ola Svensson. Linear relaxations for finding diverse elements in metric spaces. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4098–4106, 2016. URL: https://proceedings.neurips.cc/paper/2016/hash/d79c6256b9bdac53a55801a066b70da3-Abstract.html.

16 Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 13(3):41:1–41:25, 2017. doi:10.1145/3086464.

17 Mark Braverman, Young Kun-Ko, Aviad Rubinstein, and Omri Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. In *SODA*, pages 1326–1341, 2017.

18 Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. doi:10.1137/080733991.

19 Jaime G. Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 335–336. ACM, 1998. doi:10.1145/290941.291025.

20 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020. doi:10.1137/18M1166869.

21 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5029–5037, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/978fce5bcc4eccc88ad48ce3914124a2-Abstract.html.

22 Eden Chlamtác, Pasin Manurangsi, Dana Moshkovitz, and Aravindan Vijayaraghavan. Approximation algorithms for label cover and the log-density threshold. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 900–919. SIAM, 2017. doi:10.1137/1.9781611974782.57.

23 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

24 Marina Drosou, H. V. Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. Diversity in big data: A review. *Big Data*, 5(2):73–84, 2017. doi:10.1089/big.2016.0054.

25 Alessandro Epasto, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Scalable diversity maximization via small-size composable core-sets (brief announcement). In Christian Scheideler and Petra Berenbrink, editors, *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24, 2019*, pages 41–42. ACM, 2019. doi:10.1145/3323165.3323172.

26 Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

27 Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 534–543, New York, NY, USA, 2002. ACM. doi:10.1145/509907.509985.

28 Uriel Feige, Guy Kortsarz, and David Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001. doi:10.1007/s004530010050.

29 Uriel Feige and Michael Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *J. Algorithms*, 41(2):174–211, November 2001. doi:10.1006/jagm.2001.1183.

30 Uriel Feige and Michael Seltser. On the densest k-subgraph problem. Technical report, Weizmann Institute of Science, Rehovot, Israel, 1997.

31 Fedor V. Fomin, Petr A. Golovach, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. Diverse collections in matroids and graphs. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 31:1–31:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.31.

32 Doron Goldstein and Michael Langberg. The dense k subgraph problem. *CoRR*, abs/0912.5327, 2009. URL: http://arxiv.org/abs/0912.5327, arXiv:0912.5327.

33 Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009. doi:10.1145/1526709.1526761.

34 Tesshu Hanaka, Yasuaki Kobayashi, Kazuhiro Kurita, See Woo Lee, and Yota Otachi. Computing diverse shortest paths efficiently: A theoretical and experimental study. *CoRR*, abs/2112.05403, 2021. arXiv:2112.05403.

35 Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997. doi:10.1016/S0167-6377(97)00034-5.

36 Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S. Mirrokni. Composable core-sets for diversity and coverage maximization. In Richard Hull and Martin Grohe, editors, *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108. ACM, 2014. doi:10.1145/2594538.2594560.

37 Subhash Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM J. Comput.*, 36(4):1025–1071, 2006. doi:10.1137/S0097539705447037.

38 Michael J. Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987. doi:10.1111/j.1538-4632.1987.tb00133.x.

39 Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

40 Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961, 2017. doi:10.1145/3055399.3055412.

41 Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In *SODA*, pages 62–81, 2020. doi:10.1137/1.9781611975994.5.

42 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. In *ICALP*, pages 78:1–78:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.

43 Pasin Manurangsi, Aviad Rubinstein, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In *ITCS*, pages 10:1–10:21, 2021. doi:10.4230/LIPIcs.ITCS.2021.10.

44 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005. doi:10.1017/CBO9780511813603.

45 I. Douglas Moon and Sohail S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30(3):290–307, 1984. URL: http://www.jstor.org/stable/2631804.

46 Zafeiria Moumoulidou, Andrew McGregor, and Alexandra Meliou. Diverse data selection under fairness constraints. In Ke Yi and Zhewei Wei, editors, *24th International Conference on Database Theory, ICDT 2021, March 23-26, 2021, Nicosia, Cyprus*, volume 186 of *LIPIcs*, pages 13:1–13:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICDT.2021.13.

47 Dana Pessach and Erez Shmueli. Algorithmic fairness. *CoRR*, abs/2001.09784, 2020. arXiv:2001.09784.

48 Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 755–764, New York, NY, USA, 2010. ACM. doi:10.1145/1806689.1806792.

49 S. S. Ravi, Daniel J. Rosenkrantz, and Giri Kumar Tayi. Heuristic and special case algorithms for dispersion problems. *Oper. Res.*, 42(2):299–310, 1994. doi:10.1287/opre.42.2.299.

50 LT Rodrygo, Craig Macdonald, and Iadh Ounis. Search result diversification. *Foundations and Trends in Information Retrieval*, 9(1):1–90, 2015.

51 Anand Srivastav and Katja Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '98, pages 181–191, London, UK, UK, 1998. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646687.702946.

52 Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Scalable feature selection via distributed diversity maximization. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2876–2883. AAAI Press, 2017. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14914.