Approximation algorithms for geometric dispersion

THÈSE N^{O} 7291 (2016)

PRÉSENTÉE LE 18 NOVEMBRE 2016

À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE D'OPTIMISATION DISCRÈTE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alfonso Bolívar CEVALLOS MANZANO

acceptée sur proposition du jury:

Prof. J. Pach, président du jury
Prof. F. Eisenbrand, directeur de thèse
Prof. R. Zenklusen, rapporteur
Prof. N. Mustafa, rapporteur
Prof. O. Svensson, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2016

Acknowledgements

I would like to express my heartfelt gratitude to my advisor Fritz Eisenbrand. His guidance and knowledge enriched me greatly, and his encouragement brought out the best in me and helped me climb my mountain. It has been a true pleasure working in his group. I thank the Mathematics Doctoral School at EPFL and the Swiss National Science Foundation for their financial support.

I appreciate the deeply instructive research discussions I had with Andreas Haupt, Yuri Faenza and Justin Ward. Most importantly, it was a privilege to collaborate with Rico Zenklusen, and I am thankful for his many contributions to my thesis. My sincere thanks to the jury members János Pach, Ola Svensson and Nabil Mustafa, three of the smartest and most approachable researchers I have the pleasure to know, for their guidance and constructive criticism. Special thanks go to Linda Farczadi, Christoph Hunkenschröder and my esteemed office mate Manuel Aprile, for their careful proofreading and suggestions in the editing of this work.

Mein dank gilt meinem treusten Bürokollegen Adrian Bock, who was my mentor in the first part of my doctoral studies. A big thanks to our charismatic secretary Jocelyne Blanc for being always there for us. My gratitude to all the wonderful members of DISOPT and DCG for those days in the mountains, those trips around Europe, those movie nights, and so much more. Thank you Switzerland for those paths, trees and flowers.

Above all, I thank my parents for their unconditional support, and for being my favorite role models. To them I dedicate this work.

Lausanne, October 2016

Alfonso Cevallos

Abstract

The most basic form of the *max-sum dispersion* problem (MSD) is as follows: given n points in $\mathbb{R}^q$ and an integer k, select a set of k points such that the sum of the pairwise distances within the set is maximal. This is a prominent diversity problem, with wide applications in web search and information retrieval, where one needs to find a small and diverse representative subset of a large dataset. The problem has recently received a great deal of attention in the computational geometry and operations research communities; and since it is NP-hard, research has focused on efficient heuristics and approximation algorithms.

Several classes of distance functions have been considered in the literature. Many of the most common distances used in applications are induced by a norm in a real vector space. The focus of this thesis is on MSD over these geometric instances. We provide for it simple and fast polynomial-time approximation schemes (PTASs), as well as improved constant-factor approximation algorithms. We pay special attention to the class of *negative-type distances*, a class that includes Euclidean and Manhattan distances, among many others. In order to exploit the properties of this class, we apply several techniques and results from the theory of isometric embeddings.

We explore the following variations of the MSD problem: matroid and matroid-intersection constraints, knapsack constraints, and the mixed-objective problem that maximizes a combination of the sum of pairwise distances with a submodular monotone function. In addition to approximation algorithms, we present a core-set for geometric instances of low dimension, and we discuss the efficient implementation of some of our algorithms for massive datasets, using the streaming and distributed models of computation.

Key words
Combinatorial optimization, computational geometry, approximation algorithms, max-sum dispersion, remote clique, distances of negative type, theory of embeddings, convex programming, local search, core-sets.

Résumé

Dans sa forme la plus rudimentaire, le problème nommé *max-sum dispersion* (MSD) est défini comme suit : étant donnés n points sur $\mathbb{R}^q$ et un nombre entier k, choisir un ensemble de k points de façon à ce que la somme de toutes les distances par paires dans l'ensemble soit maximale. Ceci est un célèbre problème de diversité, avec un grand rang d'applications dans des domaines tels que la recherche d'information et les systèmes de recommandation, où l'on vise à extraire d'un ensemble de données un échantillon petit et divers. Récemment ce problème a reçu beaucoup d'attention dans les communautés de géométrie algorithmique et recherche opérationnelle ; et puisqu'il est NP-difficile, la recherche s'est concentrée sur l'heuristique et les algorithmes d'approximation.

Plusieurs classes de distances ont été considérées dans la littérature. De nombreuses distances utilisées dans la pratique sont associées à une norme dans un espace vectoriel. Cette thèse est axée sur le problème MSD restreint à de tels instances géométriques. On fournit pour celui-ci des schémas d'approximation en temps polynomial (PTAS), ainsi que des algorithmes d'approximation de facteur constant améliorés. On prête une attention particulière à la classe de distances *de type négatif*, classe qui inclut les distances euclidiennes et les distances de Manhattan, parmi beaucoup d'autres. Dans le but exploiter les proprietés de cette classe, on utilise plusieurs techniques et resultats provenant de la théorie de plongement isométriques.

Nous explorons les variations suivantes du problème : des contraintes définies par une matroïde ou par l'intersection de deux matroïdes, des contraintes du type sac à dos (knapsack), et le problème d'objectif mixte qui maximise une combinaison de la somme de distances avec une fonction sous-modulaire monotone. Outre des algorithmes d'approximation, on présente un core-set pour des instances géométriques de dimension basse, et on touche aussi sur la mise en œuvre efficace de quelques uns de nos algorithmes pour des ensembles de données massifs, dans les modèles de calul streaming et distribué.

Mots clefs
Optimisation combinatoire, géométrie algorithmique, algorithmes d'approximation, max-sum dispersion, remote clique, distances de type négatif, théorie de plongements, programmation convexe, recherche locale, core-sets.

Contents

1 Introduction

Diversity-maximization problems seek to retrieve a small representative sample of a large database, that is as diverse as possible. They have recently received a lot of attention due to their many applications in information retrieval, web search, recommender systems, text summarization, facility location, operations research, etc. One of the most popular functions to measure diversity is dispersion, which for an inherent dissimilarity or distance measure between pairs of data items, considers the total sum of pairwise distances between chosen items. Our definition of dispersion of a set A is thus $d(A) = \sum_{\{a,b\} \subset A} d(a,b)$. The maximization of this function, over sets restricted by a cardinality threshold or by further constraints, is known as max-sum dispersion, or MSD. Frequently, instances of this problem are geometric in nature, or a geometric interpretation can be given to the distances in the dataset. These geometric instances are the focus of this thesis.

Being a generalization of densest k-subgraph, the MSD problem is particularly hard to approximate. However, as the most common applications observe the triangle inequality, the search for approximation algorithms has concentrated on metric instances. This led to fast, greedy-based algorithms, offering an approximation ratio of $\frac{1}{2}$, which is tight under mild complexity assumptions. Yet, the theoretically hard metric instances do not play a prominent role as a dissimilarity measure, and the geometric structure of the most common distances, such as Euclidean and Manhattan, was not fully exploited in the analyses found in the literature. This fact motivated us to consider the class of distances of negative type, with its rich supporting theory of embeddings, that started with the work of Schoenberg in the 1930s.

Hence, in this work we focus mostly on negative-type distances, a class which contains Manhattan, Euclidean, Euclidean-squared, Jaccard, cosine, and many other distances that are prominent in practical applications. Our main contribution is to prove that MSD admits polynomial-time approximation schemes (PTAS) for these distances. Our algorithms work even if the embedding dimension is part of the input, hence they represent a result much stronger than anything that was previously known for distances in this class. And they work even under the constraints defined by a general matroid. Matroid constraints are particularly relevant in this context, as they model several natural restrictions expected from a small and diverse sample.

We present two procedures to achieve such a PTAS, that require very different techniques and analysis, and have different strengths. The first one uses quadratic programming and a careful randomized rounding procedure. The algorithm offers great flexibility to deal with general linear constraints. In particular, besides ensuring the satisfaction of a matroid constraint, it outputs a solution that observes concentration bounds, and hence will not violate any additional linear constraint by a large margin, with high probability.

The second algorithm is based on a standard local search. The simplicity of the algorithm makes it less flexible in terms of constraints, but very efficient and practical even for very large datasets. Furthermore, in contrast to its apparent simplicity (or rather thanks to it), this method proves to be very malleable, so that implementations for different objectives can be merged with ease into new algorithms to handle mixed objectives. We formalize this technique, and use it to obtain improved approximations for an objective that combines dispersion with a submodular monotone function. This objective models an even larger number of real-life applications, where one needs a small representative set of datapoints that maximizes both a diversity objective and a relevance objective.

Another contribution of this thesis is a PTAS for the cardinality-constrained MSD problem over distances induced by an arbitrary norm in fixed dimension. Very little was previously known about the approximability of the problem in this scenario, despite the fact that it covers several natural applications. Our result thus adds up to our understanding of the problem. The simplicity of this algorithm makes it also very applicable, even for large instances, as the computational task can be easily fragmented and handled in a distributed system.

This last algorithm is built over a core-set, with a very specific structure. This core-set has very desirable properties, as it reduces the input to a number of points linear in the output size while preserving an almost optimal solution, and it can be computed with a single-pass streaming algorithm. Finally, the combination of this core-set with the local-search algorithm results in an extremely efficient procedure, that can be implemented under streaming and distributed models.

Contribution and organization

The theoretical contributions of this thesis are found in Chapters 3, 4 and 5. These chapters are only weakly linked and do not require a sequential reading. However, we recommend the reader to start with Chapter 2, which compiles a considerable amount of required background information and notation. Each chapter starts with a summary, to facilitate its study.

In Chapter 2, we present all the needed background on distances of negative type and embeddability theory, as well as key definitions, and an extended literature review. We also prove there that the max-sum dispersion problem (MSD) is strongly NP-hard on distances of negative type, and distances in ℓ_p, for any $1 \leq p \leq \infty$ (Thm. 2.18).

Chapter 3 is based on joint work with my advisor Friedrich Eisenbrand, and Rico Zenklusen. Our work is reflected in the publication [32]. We prove that the usually non-convex quadratic relaxation of MSD can be convexified when the distances are of negative type (Thm. 3.2). Through convex optimization and a deterministic rounding procedure, we obtain a $(1 - O(\frac{\log k}{k}))$-approximation algorithm for MSD over these distances, constrained by a general matroid of rank k. (Thm. 3.4). This algorithm immediately implies a PTAS. Finally, by randomizing the rounding procedure, we extend the PTAS to the case of a matroid constraint and an additional constant number of knapsack constraints (Thm. 3.11).

Chapter 4 is also based on joint work with Friedrich Eisenbrand and Rico Zenklusen, and corresponds to the paper [31]. There, we analyze a generic non-oblivious local-search algorithm, for the maximization of a monotone increasing objective, constrained by a matroid. We study the cases where this objective is a submodular function, our dispersion function, or a linear combination of these two types of functions. As a result, we obtain a fast and simple PTAS for negative-type MSD with a matroid constraint (Thm. 4.10), as well as an asymptotically optimal $O(1)$-approximation for the mixed-objective problem (Thms. 4.7 and 4.13). We also provide a more involved PTAS for negative-type MSD constrained by a matroid intersection (Thm. 4.19).

Chapter 5 presents a PTAS for cardinality-constrained MSD over distances induced by a norm of fixed dimension q (Thm. 5.17). The algorithm performs exhaustive search over a certain collection of subsets, which is guaranteed to contain a good approximation to the problem. Its analysis exploits a quality of hollowness of the optimal solution, and the notion of subgradient of the norm function. A consequence of this analysis is the existence of a core-set for MSD over these geometric instances. This core-set offers an approximation ratio of $(1-\varepsilon)$ and has size $\tilde{O}(k)$,[1] for any $\varepsilon > 0$ and where k is the cardinality threshold; furthermore, it can be computed in a distributed fashion, or with a single-pass stream requiring space $\tilde{O}(k)$ and update time $\tilde{O}(1)$ (Thm. 5.18).

Finally, we prove that for the cardinality-constrained MSD over Manhattan, Euclidean, or Euclidean-squared distances of fixed dimension q, the construction of the previous core-set, followed by a standard local search, offers an approximation ratio of $(1-\frac{4}{k}-\varepsilon)$, in time $\tilde{O}(n+k^3)$ and space $\tilde{O}(k)$, for any $\varepsilon > 0$ and where k is the cardinality threshold and n is the size of the input set (Thm. 5.19).

[1] The notation $\tilde{O}(\cdot)$ hides terms logarithmic in k, and constants that depend on ε and q. It also hides the complexity of distance evaluations and inner products between two vectors in $\mathbb{R}^q$.

2 Preliminaries

2.1 Chapter overview

In this chapter, we present the most important background definitions and results needed in this thesis, in particular in the theory of embeddings for distance spaces. The highlights of the chapter are as follows. We present the definition and several characterizations of distances of negative type (Section 2.3.1). We include an extended list of examples of distances in this class, that are both of theoretical and practical interest. We then list several important properties of this class, in terms of the dispersion function and the auxiliary cross-dispersion function (2.3.2). And finally, we define the various versions of the max-sum dispersion problem that we will deal with in this thesis (2.3.3).

Section 2.4 contains an extended literature review on the problem. It includes a proof of the NP-hardness of max-sum dispersion over distances of negative type and several geometric instances; and it also includes a table that summarizes the state of the art in approximability for the problem.

2.2 Basic definitions and notation

Throughout this thesis, we consider a finite ground set X, with n elements. Given a set $A \subset X$ and an element $a \in X$, for brevity we use the shorthands $A + a$ for $A \cup \{a\}$, and $A - a$ for $A \setminus \{a\}$. The sets $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ are the integer numbers, rational numbers and real numbers, respectively, and to each one of these sets we add the subscript $_+$ to signify their restriction to non-negative values; for instance, $\mathbb{Z}_+ = \{0, 1, 2, \cdots\}$, and so on. The all-ones vector in $\mathbb{R}^q$ is represented as $\mathbb{1} = (1, \cdots, 1)^T$; and the set $\mathbb{S}^{q-1} = \{x \in \mathbb{R}^q : \|x\|_2^2 = 1\}$ corresponds to the unit sphere in $\mathbb{R}^q$.

We denote the symmetric difference of two sets by $A \triangle B = (A \setminus B) \cup (B \setminus A)$. For any set $A \subset X$ and vector $x \in \mathbb{R}^X$, the restricted vector $x^A \in \mathbb{R}^X$ has components $x^A_a = x_a$ if $a \in A$, and 0 otherwise. In particular, the characteristic vector of set A is $\mathbb{1}^A$.

A real symmetric matrix $Q \in \mathbb{R}^{n \times n}$ is called *positive semidefinite* if

$$x^T Q x \geq 0 \quad \text{for all } x \in \mathbb{R}^n.$$

And it is called *negative semidefinite* if the previous inequalities hold in the opposite direction, or equivalently, if $-Q$ is positive semidefinite.

A non-zero vector $x \in \mathbb{R}^n$ is an *eigenvector* of matrix Q if there is a (possibly complex) coefficient λ such that $Qx = \lambda x$. In this case, such coefficient is unique, and is called the *eigenvalue* of the corresponding eigenvector. For a real symmetric matrix, all of its eigenvalues are real. Moreover, Q is positive semidefinite if and only if all of its eigenvalues are non-negative, and similarly it is negative semidefinite if and only if all of its eigenvalues are non-positive.

Submodular functions and matroids

Consider the set function $f : 2^X \to \mathbb{R}_+$ over the ground set X. For singletons, we will write $f(a)$ as short-hand for $f(\{a\})$. The function is *monotone increasing*, or simply *monotone*, if $f(A+a) \geq f(A)$ for any $A \subset X$ and $a \in X$. It is *normalized* if $f(\emptyset) = 0$. And it is *submodular* if, for any sets $B \subset A \subset X$ and any element $a \in X \setminus A$, we have

$$f(A+x) - f(A) \leq f(B+x) - f(B).$$

If the inequalities above hold in the opposite sense, f is *supermodular*. A function is *linear* if and only if it is both submodular and supermodular, in which case the above inequalities hold with equality. If f is linear, then there are coefficients $w(a) = f(a) - f(\emptyset)$ for each $a \in A$, such that $f(A) = f(\emptyset) + \sum_{a \in A} w(a)$ for all $A \subset X$.

Next, we provide an overview of the basics of matroid theory. For further information, we refer to [114]. A *matroid* $(X, \mathscr{I})$ over the ground set X consists of a non-empty family $\mathscr{I} \subset 2^X$ of subsets, called *independent sets*, satisfying:

1. if $A \in \mathscr{I}$ and $B \subset A$, then $B \in \mathscr{I}$; and
2. if $A, B \in \mathscr{I}$ and $|A| > |B|$ then there is an element $a \in A \setminus B$ such that $B + a \in \mathscr{I}$.

The matroid $(X, \mathscr{I})$ defines a *rank* function $r : 2^X \to \mathbb{Z}_+$, where $r(A) = \max\{B \in \mathscr{I} : B \subset A\}$, i.e., it is equal to the largest cardinality of an independent subset of A. The rank function characterizes its matroid, and it is always submodular, monotone and normalized. The value $r(X)$ is called the rank of the matroid, and we will usually denote it by k. Inclusion-wise maximal independent sets are called *bases*, and it is a consequence of the definition of matroid that all bases are of equal cardinality k.

We list some common examples of matroids. In a *uniform matroid* of rank k, $\mathscr{I}$ consists of all sets in X of cardinality at most k. A *partition matroid* is defined in terms of a partition of the ground set $X = \cup_{i=1}^{p} X_i$, and integers $k_1, \cdots, k_p$, and a set $A \subset X$ is in $\mathscr{I}$ if $|A \cap X_i| \leq k_i$ for all $1 \leq i \leq p$. In a *graphical* matroid, the ground set X corresponds to the edges of a given graph G, and an edge set is independent if it contains no cycles. Finally, in a *linear matroid*, the ground set X contains vectors in a vector space, and a vector set is in $\mathscr{I}$ if it is linearly independent.

The *matroid polytope* $P(\mathscr{I}) \subset \mathbb{R}^X$ of matroid $(X, \mathscr{I})$ is the convex hull of the characteristic vectors $\mathbb{1}^A$ of all independent sets $A \in \mathscr{I}$. It can be described as follows:

$$P(\mathscr{I}) = \left\{ x \in \mathbb{R}_+^X : \mathbb{1}^T x^A \leq r(A),\ \forall A \subset X \right\},$$

where $\mathbb{1}^T x^A = \sum_{a \in A} x_a$. The *base polytope* $P_B(\mathscr{I}) \subset \mathbb{R}^X$ is the convex hull of the characteristic vectors $\mathbb{1}^A$ of all bases in A, and it can be described by

$$P_B(\mathscr{I}) = P(\mathscr{I}) \cap \left\{ x : \mathbb{1}^T x = k \right\}.$$

Complexity and approximation algorithms

In this thesis we restrict our attention to a specific type of combinatorial optimization problems – namely, to the constrained maximization of set functions. We remark that the definitions presented here have been narrowed to this context, and are hence different from standard definitions. We consider problems where an instance is defined by

1. a ground set X on n elements,
2. an objective function $f : 2^X \to \mathbb{R}_+$ to be maximized, and
3. a family $\mathscr{F} \subset 2^X$ of *feasible solutions*.

The goal of the problem is to find the feasible set $A \in \mathscr{F}$ that maximizes the objective function $f(A)$. We denote the optimal set in $\mathscr{F}$ by O, and its value by $opt = f(O)$. In fact, in the present work we will only deal with functions that are monotone.

We assume that the instance $(X, f, \mathscr{F})$ is represented in a compact way. To achieve this, the family $\mathscr{F}$ may be defined indirectly via a *membership oracle*,[1] which answers whether or not a set A is in $\mathscr{F}$; and the function f may be defined by a *value oracle*, which returns the value $f(A)$ of any set A. We similarly assume that all evaluations of $f(\cdot)$ have a small binary encoding.

As we will deal with NP-hard problems, we focus on approximation algorithms. For a problem $\mathbb{P}$ containing instances $(X, f, \mathscr{F})$ as above, an *approximation algorithm* is an always-halting process, that takes as input an instance $(X, f, \mathscr{F})$ from $\mathbb{P}$, performs a number of operations that is polynomial in the size of the representation of the instance, including all necessary oracle calls, and outputs a feasible set $S \in \mathscr{F}$. Such an algorithm has an *approximation ratio* $\alpha \geq 0$ if $f(S)/f(O) \geq \alpha$ for all instances of problem $\mathbb{P}$. For a *randomized* approximation algorithm, where the output S is a random set in 2^X, we say that it has an approximation ratio of α if, for all instances of problem $\mathbb{P}$, with probability at least $1/2$ we will have that S is feasible and $f(S)/f(O) \geq \alpha$. The coefficient α may be constant or depend on the instance; in particular, we will consider cases where α depends on $|X| = n$, on $|O|$, and even on O.

We aim to design algorithms with approximation ratios that are as high as possible. An approximation ratio can have value at most 1, and is strictly smaller than 1 if the problem is NP-hard

[1] This is also called an *independence oracle* when $\mathscr{F}$ is defined by a matroid.

(unless P=NP). A problem $\mathbb{P}$ may have a *hardness of approximation* result, which establishes a bound $\beta < 1$ and gives evidence against the existence of an algorithm with $\alpha \geq \beta$. If the approximation ratio of an algorithm is equal to a hardness bound for the problem, we say that it is *tight*.

For a problem $\mathbb{P}$, an approximation algorithm is a *polynomial time approximation scheme* (PTAS) if, for any constant $\varepsilon > 0$, it can be calibrated to achieve an approximation ratio of $1 - \varepsilon$. The running time of such algorithm must be polynomial in the size of the input instance, but may have any kind of dependency on $1/\varepsilon$. If the running time is also polynomial in $1/\varepsilon$, the algorithm is a *fully polynomial time approximation scheme* (FPTAS). Finally, a randomized PTAS is also called a PRAS.

Remark 2.1. In this thesis, we will use the fact that if the approximation ratio of an algorithm is $\alpha = 1 - o(1)$, as $|O|$ increases, then the algorithm immediately defines a PTAS. Indeed, for any constant $\varepsilon > 0$, if $\alpha \geq 1 - \varepsilon$ then the algorithm achieves the desired ratio of $1 - \varepsilon$. Otherwise, the optimal solution O has a size bounded by a constant (that depends only on ε), so it can be found in polynomial time by an exhaustive search.

2.3 Distance and dispersion

An important part of the theoretical background needed in this thesis comes from the theory of embeddings. We review some relevant notions and results, and refer to [44, 97] for a thorough account. A finite *distance space* (X, d) over the ground set X is defined by a symmetric function

$$d : X^2 \to \mathbb{R}_+,$$

with the property that $d(a, a) = 0$ for all $a \in X$. We call $d(a, b)$ the *distance* between a and b. However, for brevity and when there is no risk of ambiguity, we will use the term *distance* as short-hand for finite distance space. The distance (X, d) is called *metric* if it observes the triangle inequality: $d(a, b) + d(b, c) \geq d(a, c)$ for all $a, b, c \in X$.[2]

For a real vector space $\mathbb{R}^q$, a *norm* $\|\cdot\|_* : \mathbb{R}^q \to \mathbb{R}_+$ is a function such that for all $x, y \in \mathbb{R}^q$ and $\lambda \in \mathbb{R}$: a) $\|\lambda x\|_* = |\lambda| \cdot \|x\|_*$, b) if $x \neq 0$ then $\|x\|_* > 0$, and c) $\|x + y\|_* \leq \|x\|_* + \|y\|_*$ (this last property is called subadditivity). In particular, the ℓ_p norm is defined as

$$\|x\|_\infty = \max_{1 \leq i \leq q} |x_i| \quad \text{and} \quad \|x\|_p = \left[\sum_{i=1}^{q} |x_i|^p \right]^{1/p} \text{for } p \geq 1.$$

We extend this last definition for values $0 < p < 1$, even though for these values the function $\|\cdot\|_p$ is not a proper norm as it does not respect the subadditivity property (this is known as a *quasi-norm*).

[2] We highlight that our definition of metric distance space, sometimes called *semi-metric*, is weaker than the standard definition, which also establishes the property that $d(a, b) > 0$ whenever $a \neq b$. In general we have no need to assume this last property in this thesis.

A distance (X,d) is *isometrically embeddable* into $(\mathbb{R}^q, \|\cdot\|_*)$ if there is an *embedding* $\rho : X \to \mathbb{R}^q$ such that for all $a,b \in X$, we have $d(a,b) = \|\rho(a) - \rho(b)\|_*$. In this case, we may also say that (X,d) is *induced* by norm $\|\cdot\|_*$. Norm-induced distances are always metric, as a consequence of the subadditivity property. In particular, for $0 < p \leq \infty$, a distance (X,d) is ℓ_p-embeddable, or is *in* ℓ_p, if there is a dimension q such that it is isometrically embeddable into $(\mathbb{R}^q, \|\cdot\|_p)$.[3] Distances in ℓ_1 and ℓ_2 are usually called *Manhattan* and *Euclidean* distances, respectively. It will also be convenient to make an extra, non-standard definition: we say that (X,d) is *Euclidean-squared* if there is a dimension q and an embedding $\rho : X \to \mathbb{R}^q$ such that $d(a,b) = \|\rho(a) - \rho(b)\|_2^2$ for all $a,b \in X$. This class of distances is in general not metric;[4] however, as we will show, it is a very interesting class from both theoretical and practical viewpoints.

We cite two classic results in embeddability theory. They establish some inclusions among the classes of ℓ_p-embeddable distances.

Proposition 2.2 (Fréchet [60])**.** *If a distance (X,d) is metric, then it is in ℓ_∞, with embedding $\rho : X \to \mathbb{R}^X$, where $\big(\rho(a)\big)_b = d(a,b)$ for all $a,b \in X$.*

Proposition 2.3 (Dor [46])**.** *If (X,d) is in ℓ_2, then it is in ℓ_p for all $1 \leq p \leq \infty$.*

2.3.1 Distances of negative type

A distance (X,d) can be represented in a compact way by the symmetric matrix $D \in \mathbb{R}_+^{X\times X}$, where $D_{a,b} = d(a,b)$ for all $a,b \in X$, known as the *distance matrix*. Distance (X,d) is *of negative type* if

$$x^T D x \leq 0 \quad \text{for all } x \in \mathbb{Z}^X \text{ with } \sum_{a\in X} x_a = 0.$$

We present some alternative characterizations of this definition. Recall that $\mathbb{1} = (1,\cdots,1)^T$ is the all-ones vector in $\mathbb{R}^X$.

Lemma 2.4. *(X,d) is of negative type if and only if*

$$x^T D x \leq 0 \quad \textit{for all } x \in \mathbb{R}^X \textit{ with } \mathbb{1}^T x = 0. \tag{2.1}$$

Proof. It is clear that $\sum_{a\in X} x_a = \mathbb{1}^T x$, so the conditions are equivalent for integer-valued vectors. This proves the *if* part. For the *only if* part, assume the distance is of negative type. Inequality (2.1) on integer-valued vectors implies the same condition on rational-valued vectors – this is verified simply by multiplying a vector x by the lowest common denominator of its coordinates. And finally, if we consider the function $x^T Dx$ defined over the subspace $\{x : \mathbb{1}^T x = 0\}$, the fact that it is continuous and non-positive over all rational points implies that it is also non-positive over its entire domain. This completes the proof. □

[3] We stress the fact that the corresponding dimension q and embedding ρ need not be known. For instance, finding an embedding for an ℓ_2-embeddable distance is polynomial-time solvable, while the respective task for an ℓ_1-embeddable distance is NP-complete [13]. If (X,d) is ℓ_p-embeddable, with $|X| = n$, then the minimum necessary dimension q is at most $n-1$ for $p = 2$ and $p = \infty$, and at most $\binom{n}{2}$ for all $p \geq 1$ [16, 57].

[4] The simplest example of a Euclidean-squared distance that is not metric is the triple $X = \{a,b,c\}$ with $d(a,b) = d(b,c) = 1$ and $d(a,c) = 4$, that is embeddable into a line.

Remark 2.5. Inequalities of the form (2.1) are called *negative-type inequalities*, and they imply that matrix D is *conditionally negative semidefinite*. This term means that it satisfies the conditions of a negative semidefinite matrix over a proper subspace, in this case the $(n-1)$-dimensional subspace orthogonal to $\mathbb{1}$. A real symmetric matrix D is the distance matrix of a negative-type distance if and only if it has zero diagonal entries, non-negative off-diagonal entries, and satisfies (2.1) (see [24]). Furthermore, such a matrix will have exactly one positive eigenvalue, provided it is not the zero matrix [44, Theorem 6.2.16].

Lemma 2.6. *(X,d) is of negative type if and only if*

$$\frac{(y+z)^T D(y+z)}{\mathbb{1}^T(y+z)} \geq \frac{y^T Dy}{\mathbb{1}^T y} + \frac{z^T Dz}{\mathbb{1}^T z} \quad \text{for all } y,z \in \mathbb{R}_+^X \setminus \{0\}. \tag{2.2}$$

Proof. For the *only if* part, assume the distance is of negative type, and consider some non-zero vectors y,z in $\mathbb{R}_+^X$. If we define $x = \sqrt{(\mathbb{1}^T z)/(\mathbb{1}^T y)}\,y - \sqrt{(\mathbb{1}^T y)/(\mathbb{1}^T z)}\,z$, then $\mathbb{1}^T x = 0$, and inequality (2.1) gives

$$0 \geq x^T Dx = \frac{\mathbb{1}^T z}{\mathbb{1}^T y} y^T Dy + \frac{\mathbb{1}^T y}{\mathbb{1}^T z} z^T Dz - 2y^T Dz. \tag{2.3}$$

On the other hand, we have the equation

$$(y+z)^T D(y+z) = y^T Dy + z^T Dz + 2y^T Dz, \tag{2.4}$$

and summing up the last two lines yields the result:

$$\begin{aligned}(y+z)^T D(y+z) &\geq \left(1 + \frac{\mathbb{1}^T z}{\mathbb{1}^T y}\right) y^T Dy + \left(1 + \frac{\mathbb{1}^T y}{\mathbb{1}^T z}\right) z^T Dz \\ &= \left(\mathbb{1}^T y + \mathbb{1}^T z\right)\left[\frac{y^T Dy}{\mathbb{1}^T y} + \frac{z^T Dz}{\mathbb{1}^T z}\right].\end{aligned}$$

Now, for the *if* part, assume that inequality (2.2) holds, and consider a vector $x \in \mathbb{Z}^X$ with $\mathbb{1}^T x = 0$. As the case $x = 0$ is trivial, we assume that x is non-zero. Let $y = \max\{x, 0\}$ and $z = \max\{-x, 0\}$, where the max function is taken component-wise. Then y and z are non-zero vectors in $\mathbb{R}_+^X$, $x = y - z$, and the condition $\mathbb{1}^T x = 0$ implies that $\mathbb{1}^T y = \mathbb{1}^T z$. The proof now works backwards. Using equation (2.4) over inequality (2.2), we obtain

$$\frac{y^T Dy + z^T Dz + 2y^T Dz}{2} \geq y^T Dy + z^T Dz.$$

And if we multiply by 2 and move all terms to the right-hand side,

$$0 \geq y^T Dy + z^T Dz - 2y^T Dz = (y-z)^T D(y-z) = x^T Dx.$$

□

Proposition 2.7 (Schoenberg [113])**.** *(X,d) is of negative type if and only if it is Euclidean-squared.*

In Remark 2.14 we will provide some intuition behind this surprising geometric characterization.[5] The previous result was proven by Isaac Schoenberg, who in the 1930s developed a deep theory over distances of negative type and isometric embeddings. With this result, it is easy to check that these distances are in general not metric (or vice versa), hence results for these two distance classes are not directly comparable.

Examples of distances of negative type

We present several examples of distances of negative type, as well as several distance transformations that either produce or preserve this property. For a distance d and a transformation function $f : \mathbb{R}_+ \to \mathbb{R}_+$ with $f(0) = 0$, we represent by $f(d)$ the distance defined by $f(d)(a,b) = f(d(a,b))$.

Proposition 2.8 (Schoenberg [113])**.** *For any $0 < \alpha < p \leq 2$, if (X,d) is in ℓ_p, then (X,d^α) is of negative type.*

A direct consequence of Proposition 2.8 (setting $\alpha = 1$) is that Euclidean and Manhattan distances are of negative type, as are all distances in ℓ_p for $1 \leq p \leq 2$. Euclidean distances are widely popular in several diversity maximization problems, especially in facility location (see Section 2.4). And Manhattan distances are a prominent similarity measure in information retrieval [96], in particular when using sketching techniques [94] where data points are represented by small-dimensional bit-vectors whose Hamming distance approximates the distance of the corresponding points.

In many applications, the data objects that populate the ground set X are described in terms of features, and a dissimilarity or distance function measures how different these features are between two objects. When these features are binary, each object in X is represented by a set $P \subset U$, where U is the collection of all features. There exist several popular definitions of distances $d(P,Q)$ between two sets $P,Q \subset U$, which are of negative type, such as the *Jaccard* distance $\frac{|P \triangle Q|}{|P \cup Q|}$ [68], *Simple Matching* $\frac{|P \triangle Q|}{|U|}$, *Russel and Rao* $1 - \frac{|P \cap Q|}{|U|}$, *Dice* $\frac{|P \triangle Q|}{|P|+|Q|}$, etc. We remark that the Dice distance is not metric. We refer to [102, Table 5.1] for an extensive list of binary dissimilarity measures, classified by metricity and negative type.

For more general data, the features are quantitative. This naturally leads to a representation of the objects in X as vectors in space $\mathbb{R}^q$, where q is the number of features. In this case, norm-induced distances are usually applied [96, 111]. However, sometimes one may want to ignore the absolute magnitudes of the features, and rather compare objects by the relative weights of their features. For vectors $x, y \in X \subset \mathbb{R}^q \setminus \{0\}$, we achieve this goal by defining a distance $d(x,y)$ that depends exclusively on the angle $\theta_{x,y}$ between x and y. The *spherical*

[5] Because of this geometric connection, the distance matrix of a negative-type distance is also known in the literature as a *Euclidean distance matrix* (see, e.g., [24]).

distance is defined by $d(x,y) = \theta_{x,y}$,[6] and it is known to be ℓ_1-embeddable [20, 44], hence it is of negative type. The *cosine distance*, on the other hand, is defined by $d'(x,y) = 1 - \cos\theta_{x,y}$, and it is easy to prove that it is Euclidean-squared: simply verify that $d'(x,y) = \frac{1}{2}\|\frac{x}{\|x\|_2} - \frac{y}{\|y\|_2}\|_2^2$. Hence, it is also of negative type (and non-metric).

The following results will be useful for problem reductions.

Proposition 2.9 (Deza and Maehara [45])**.** *If (X,d) is metric, with $|X| = n$, then $(X, d^{\log_2(\frac{n}{n-1})})$ is of negative type.*

Lemma 2.10. *Let (X,d) be a metric space with $|X| = n$, such that all distances between distinct points are in the range $[1,c]$.*

- *If $c = 2$, then (X,d) is metric;*
- *If $c = \frac{n}{n-1}$, then (X,d) is of negative type; and*
- *If $c = \sqrt{\frac{n}{n-1}}$, then (X,d) is Euclidean.*

Proof. If (X,d) is such that all distances are in the range $[1,2]$, the triangle inequality can be easily verified, so it is metric. By applying Proposition 2.9 and the transformation $f(d) = d^{\log_2(\frac{n}{n-1})}$, we obtain the second claim. And the third claim follows from Proposition 2.7 and the transformation $f(d) = \sqrt{d}$. □

We present some additional examples and transformations which are of theoretical interest.

Proposition 2.11 (Schoenberg [112])**.** *If (X,d) is of negative type, then $(X, f(d))$ remains of negative type under any of the following transformations: $f(d) = \frac{d}{1+d}$, $\ln(1+d)$, $1 - e^{-\lambda d}$ for $\lambda > 0$, and d^α for $0 \le \alpha \le 1$.*

Proposition 2.12 (See Theorem 3.6 in [98])**.** *The following distance classes are of negative type: distances induced by a two-dimensional norm, metric distances of up to four points, ultrametric distances, hyperbolic distances, and weighted tree distances.*

2.3.2 The dispersion and cross-dispersion functions

We abuse notation by a great deal in this thesis, and use $d(\cdot)$ to represent several related functions. We do it to keep notation simple, and because we believe that doing so will simplify and bring intuition to many of this work's proofs.

Given a distance (X,d), we define the *dispersion* function $d : 2^X \to \mathbb{R}_+$ as

$$d(A) = \sum_{\{a,a'\}\subset A} d(a,a') = \frac{1}{2}\sum_{a,a'\in A} d(a,a') \quad \forall A \subset X.$$

[6] If one desires to have the property that distinct elements have non-zero distances, then the ground set must be restricted to the unit sphere $\mathbb{S}^{q-1}$ in $\mathbb{R}^q$.

The dispersion of A corresponds to the sum of pairwise distances within A. We also define the auxiliary *cross-dispersion* function as

$$d(A,B) = \sum_{a\in A, b\in B} d(a,b) \quad \forall A,B \subset X.$$

We write (A,b) as short-hand for $(A,\{b\})$. Some easy-to-check properties of these functions are

$$d(A\cup B) = d(A) + d(B) + d(A,B), \quad d(A,A) = 2d(A), \quad d(A,a) = d(A-a,a),$$

for all disjoint sets $A,B \subset X$, and element $a \in A$.

Both of these functions can be written in terms of the distance matrix D and characteristic vectors. Namely, $d(A) = \frac{1}{2}(\mathbb{1}^A)^T D(\mathbb{1}^A)$, and $d(A,B) = (\mathbb{1}^A)^T D(\mathbb{1}^B)$, for any sets $A,B \subset X$. This fact motivates us to further define the *extended* dispersion and cross-dispersion functions for arbitrary vectors in $\mathbb{R}^X$, in such a way that $d(A) = d(\mathbb{1}^A)$ and $d(A,B) = d(\mathbb{1}^A,\mathbb{1}^B)$. Let

$$d(x) = \frac{1}{2}x^T Dx \quad \text{and} \quad d(x,y) = x^T Dy \quad \forall x,y \in \mathbb{R}^X.$$

These extended functions have the properties

$$d(x+y) = d(x) + d(y) + d(x,y), \quad d(\lambda x) = \lambda^2 d(x), \quad d(\lambda x, y) = \lambda d(x,y), \tag{2.5}$$

for all vectors $x,y \in \mathbb{R}^X$ and scalar $\lambda \in \mathbb{R}$.

When the distance (X,d) is of negative type, then $d(x) \le 0$ for all $x \in \mathbb{R}^X$ with $\mathbb{1}^T x = 0$. Lemma 2.6 implies the following additional properties, which are key for the analysis of the approximation algorithms that we will present for this class of distances. They correspond to inequalities (2.2) and (2.3) in Lemma 2.6.

Lemma 2.13. *If (X,d) is of negative type, then for any non-zero vectors $x,y \in \mathbb{R}^X_+$,*

$$d(x,y) \ge \frac{\mathbb{1}^T y}{\mathbb{1}^T x} d(x) + \frac{\mathbb{1}^T x}{\mathbb{1}^T y} d(y) \quad \textit{and} \tag{2.6}$$

$$\frac{d(x+y)}{\mathbb{1}^T(x+y)} \ge \frac{d(x)}{\mathbb{1}^T x} + \frac{d(y)}{\mathbb{1}^T y}. \tag{2.7}$$

Consequently, for any non-empty sets $A,B \in X$,

$$d(A,B) \ge \frac{|B|}{|A|} d(A) + \frac{|A|}{|B|} d(B), \tag{2.8}$$

and if additionally A and B are disjoint,

$$\frac{d(A\cup B)}{|A\cup B|} \ge \frac{d(A)}{|A|} + \frac{d(B)}{|B|}. \tag{2.9}$$

Remark 2.14. To provide an intuition to Schoenberg's geometric characterization of negative-type distances (Proposition 2.7), we mention that inequality (2.9) can also be proved from a Euclidean-squared embedding of the distance space, and the notion of centroid. Given a finite, non-empty set of points $A \subset \mathbb{R}^q$, the centroid of A is defined as $c_A = \frac{1}{|A|}\sum_{a\in A} a$. This point observes the following two properties, which will be proved in Lemma 5.1.

$$d(A, c_a) = \frac{d(A)}{|A|} \quad \text{and} \quad d(A, c_A) = \min_{c\in\mathbb{R}^q} d(A, c).$$

Then, for two finite, disjoint and non-empty sets A, B in $\mathbb{R}^q$, inequality (2.9) simply states that

$$d(A\cup B, c_{A\cup B}) = d(A, c_{A\cup B}) + d(B, c_{A\cup B}) \geq d(A, c_A) + d(A, c_B).$$

Remark 2.15. For metric distances, the following inequality similar to (2.8) holds:[7]

$$d(A, B) \geq \frac{|A|}{|B|-1} d(B), \quad \text{for all } A, B \subset X \text{ with } |B| \geq 2. \tag{2.10}$$

This property has often been used in the analysis of approximation algorithms, see [108, 23]. Intuitively, inequality (2.8) is weaker than inequality (2.10) for small sets, but (much) stronger for large sets.

2.3.3 The max-sum dispersion problem

This thesis focuses on the problem of *max-sum dispersion*, or MSD for short. It is

$$\max\{d(A) :\ A \in \mathscr{F}\},$$

where (X, d) is a distance space, and $\mathscr{F} \subset 2^X$ is a given family of *feasible* subsets. In words, we want to maximize the dispersion of a feasible subset. Some examples of feasible families $\mathscr{F}$ that will consider are those defined by a cardinality constraint, a matroid, the intersection of two matroids, and knapsack constraints. We will specify what $\mathscr{F}$ is in every case, except for the cardinality-constrained problem $\max\{d(A) :\ A \subset X,\ |A| \leq k\}$, which receives the special label MSD_k. We will also in general clarify the class of distances we are restricting the problem to, by names such as *metric* MSD, *negative-type* MSD, and so on.

For a set function $f : 2^X \to \mathbb{R}_+$ that is submodular and monotone, we will also consider the maximization problem with combined objective

$$\max\{g(A) :\ A \in \mathscr{F}\}, \quad \text{where } g(A) = d(A) + f(A) \quad \forall A \subset X.$$

For convenience, we denote this problem by MSD + f. If, in addition, f is linear, we denote the corresponding problem by MSD + l. As all considered objectives are monotone, we will always restrict our attention to feasible solutions that are inclusion-wise maximal sets.

[7] This inequality is proven by Ravi et al. [108], originally only for disjoint sets, but their proof extends to the general case.

2.4 Literature review

The max-sum dispersion problem, in particular in its cardinality-constrained version (MSD_k), is one of the most prominent *diversity maximization* problems. This type of problems seeks to maximize a diversity function over a subset of specified cardinality, from a large ground set. They include, for instance, the also popular *max-min dispersion* problem, where the diversity function is $div(A) = \min\{d(a,a') : a,a' \in A,\ a \neq a'\}$.

Remark 2.16. A related problem, the *max-mean dispersion* [29], is defined as $\max_{A\subset X,|A|\leq k} \frac{d(A)}{|A|}$. Even though the objective function is not monotone in general, from inequalities (2.9) and (2.10) it can be easily proved to be monotone for both negative-type and metric distances; hence for these distances the problem is equivalent to MSD_k in terms of approximability. In another related problem, called the *p-maxian problem* [49], the ground set $X = Y \cup Z$ is partitioned into unselected points Y and preselected points Z, and the goal is to find a subset $A \subset Y$ of bounded size that maximizes $d(A \cup Z)$. We note that this can be easily converted into an MSD + l instance.

The problem of MSD_k receives several names in the literature, such as *maxisum dispersion*, *max-avg dispersion*, *maximum diversity*, and *remote clique*. Its origin can be traced to several, separate research communities.

In the context of facility location, max-sum dispersion and max-min dispersion were introduced by Kuby [91]. Several applications have been proposed, where one must select locations that are far from each other. For instance, strategic facilities such as oil tanks [99] and ammunition dumps [51] should be kept separated from each other, in order to minimize the damage of a localized attack. Hazardous equipment susceptible to fire or other accidents should also be properly spaced to minimize the risk of spread [91]. Location diversity is desirable for business franchises, seeking to avoid mutual competition; or for the placement of firehouses and ambulance stations, in order to obtain an efficient an fair coverage of a city [125].

Many other applications in facility location have been suggested [33], and the work was followed by [50, 48, 85, 74, 104], where heuristics are considered for several scenarios and applications, and where the focus is almost exclusively on Euclidean distances.

In the context of representing a large database by a small and diverse sample, MSD_k was introduced by Glover et al. [65], for an application in biological diversity preservation. Examples of application scenarios range from agricultural breeding stocks, to composing jury panels, to very-large-scale integration (VLSI) [86]. It received immediate attention in the operations research community in the 1990s and 2000s, and many heuristics were applied to it, such as linearization of the quadratic formulation, local-search and greedy algorithms, GRASP, tabu search, simulated annealing, Lagrangian relaxation, etc. [63, 66, 124, 12, 115, 43, 47, 62]. In most of these papers, the experimental results are either performed on Euclidean distances, or on randomly generated non-metric distances.

Approximation algorithms for MSD_k

When no assumption is made on the distance class, the cardinality-constrained problem (MSD_k) corresponds to a weighted version of the *densest k-subgraph* problem (DkS), which is notoriously hard. For a graph G and a number k, the goal of DkS is to find a set of k vertices in G that induce the largest number of edges. This case is outside of the scope of the present work, hence we skip the literature survey and refer the reader to the relevant work [88, 9, 55, 117, 11, 54, 53]. We only mention that DkS is known to be strongly NP-hard and not admitting a PTAS [84], its current best approximation ratio is (only) of $\frac{1}{O(n^{1/4-\varepsilon})}$ for any $\varepsilon > 0$ [17], and the problem admits no constant-factor approximation under the assumption that the *planted clique* problem is hard [8].

As most applications work with distances that are metric, or even geometric (induced by a norm), the search for heuristics with provably good performance has focused on these distance classes. For metric MSD_k, Ravi et al. [108] obtained the first constant-factor approximation, showing that the standard greedy algorithm (that iteratively selects the item maximizing the marginal gain until k items are selected) has an approximation ratio of $\frac{1}{4}$. Hassin et al. [77] then presented a $\frac{1}{2}$-approximation, obtained by a somewhat slower greedy algorithm (that selects two new items of maximal mutual distance at each step). Birnbaum and Goldman [19] finally showed that a ratio of $\frac{1}{2}$ is also guaranteed by the standard greedy studied in [108]. This ratio is tight, as it is shown in [23] that the problem admits no approximation factor of $\frac{1}{2}+\varepsilon$, for any constant $\varepsilon > 0$, again under the assumption that the planted clique problem is hard.

For the case of distances induced by a norm in fixed dimension, Ravi et al. [108] presented an efficient exact algorithm for dimension 1,[8] and they used it to obtain a $\frac{2}{\pi}$-approximation for planar Euclidean distances, based on the idea of projecting the points into a random 1-dimensional subspace. Fekete and Meijer [56] then presented a PTAS for Manhattan distances on any fixed dimension; and they observed that their result implies an approximation ratio of $(\frac{1}{\sqrt{2}}-\varepsilon)$ for planar Euclidean distances, for any $\varepsilon > 0$. The authors in [108] and [56] both remark that the NP-hardness of MSD_k is open over fixed-dimensional Manhattan and Euclidean distances.

Diversification, matroid constraints, and mixed objectives

There is growing interest in the application of MSD_k and other related problems in information retrieval [21, 128, 38, 121, 129, 105, 106]. These instances are mostly related to Internet applications with very large databases, from which a small sample of items must be presented to the users, based on a query or known user attributes. Contrary to the aforementioned applications, here the main objective is to maximize a *relevance* function, which is a priori unrelated to any notion of diversity. However, an additional measure of diversity is introduced, in order to avoid redundancy, to satisfy a maximum number of users, or to increase the chance of relevance when a query is ambiguous. Problems that combine these two objectives are

[8] In fact, Tamir [120] later remarked that optimal solutions in dimension 1 always have a simple structure, and hence can be found by a trivial algorithm (see Section 5.4.1).

referred to as *diversification* problems, and application examples consist of online shopping, web search, document summaries, etc.

For instance, in web search, diversification is considered to be an effective solution to minimize query abandonment, which is the problem of a user not finding any relevant result within the returned items [7, 39, 42]. Furthermore, diversification has been recently explored as an aid in recommender systems [130, 126]. Recommender systems [109] provide customers with recommendations of products they might be interested in, based on their past purchases or preferences, and demographic information. In this context, diversification decreases redundancy, and increases customer satisfaction under the natural assumptions that users have a wide range of interests and enjoy to explore new products.

In the context of document summary, Carbonell and Goldstein [28] modeled diversification by defining an objective function which is the convex combination of two set functions: a relevance measure that depends on the query, and a diversity measure that depends only on the chosen set. A parameter controls the degree of the trade-off. Later on, Gollapudi and Sharma [67] and Bhattacharya et al. [18] took this model and considered the dispersion function over a metric distance to measure diversity, and a monotone linear function to measure relevance. That is, they considered the metric, cardinality-constrained MSD + l problem. And they presented an approximation ratio of $\frac{1}{2}$ for it. It is worth mentioning that Gollapudi and Sharma select the dispersion function (the sum of pairwise distances) as an optimal measure of diversity for web search, after a detailed study of desirable properties for an objective function; and they consider Jaccard and weighted tree distances, both of which are of negative type. On the other hand, Bhattacharya et al. argue that the choice of the dispersion function is desirable in the context of e-commerce, and validate this claim through a user study.

Motivated by an application in news aggregator services, Abbassi, Mirrokni and Thakur [2] introduced matroid constraints to the study of this problem, as an additional means to ensure diversity. In particular, for news articles that are classified into categories, they highlight the use of a partition matroid constraint in order to limit the number of results within the same category. For metric MSD under a matroid constraint, they prove that a standard local-search algorithm also achieves an approximation ratio of $\frac{1}{2}$. Since a cardinality constraint can be represented by a uniform matroid, the $\frac{1}{2}$-hardness that is known for metric MSD$_k$ still holds for this new constraint, hence their approximation ratio is tight.

Finally, Borodin et al. [23] studied the more general scenario where the relevance function is a monotone and submodular function, i.e., they consider the metric MSD + f problem, also under a matroid constraint. They prove that the standard local-search algorithm achieves an approximation of $\frac{1}{2}$, which is known to be tight for both functions in the objective. They argue that monotone submodular functions naturally model the relevance of the set of returned items for a keyword-based search in a database.

Distance class	Hardness result	Approximation ratio
Metric / ℓ_∞	No $1/2 - \varepsilon$ † [23]	$1/2$ [77]
Negative type	No FPTAS (Thm. 2.18)	PTAS (Thms. 3.4, 4.10)
ℓ_p for $1 \le p \le 2$	No FPTAS (Thm. 2.18)	PTAS (Thms. 3.4, 4.10)
ℓ_p for $2 < p < \infty$	No FPTAS (Thm. 2.18)	$1/2$ [77]
One-dimensional	1	1 [108]
q-dimensional, fixed $q \ge 2$ (all norms)	1	PTAS (Thm. 5.17)

Table 2.1: This table presents the current best hardness results and approximation ratios for MSD_k over several distance classes. The sign † indicates that the result assumes hardness of the planted clique problem.

Remark 2.17. While the max-sum dispersion objective maximizes the average distance in the output set, it may select points that are clustered together in some regions. If such clusters are undesirable, a possible solution is to maximize the function $d(A) + f(A)$, where $f(A)$ equals the total number of elements in the ground set X that are covered by balls of radius r and centered in elements of A, for a fixed $r > 0$. It can be checked that $f(A)$ is a monotone submodular function, and thus this is an MSD + f problem.

State of the art in approximability

We prove below that the max-sum dispersion problem remains strongly NP-hard, even when restricted by a cardinality constraint, over distances of negative type, as well as general geometric instances.

Theorem 2.18. MSD_k *is strongly NP-hard when restricted to any of the following distance classes: metric distances, distances of negative type, and distances in* ℓ_p*, for any* $1 \le p \le \infty$*. In particular, for these classes it does not admit fully polynomial time approximation schemes.*

Proof. We present a reduction from DkS, which is strongly NP-hard. For a DkS instance given by $G = (V, E)$ and k, we define the distance (V, d) where two distinct vertices are at distance $\sqrt{\frac{n}{n-1}}$ if they are adjacent, and at distance 1 if they are not. By Lemma 2.10, this distance is metric, of negative type, and Euclidean. And by Proposition 2.3, it is also in ℓ_p for all $1 \le p \le \infty$. If the optimal solution to the DkS instance has value opt, then the optimal solution to MSD_k over (V, d) with parameter k will be exactly $(\sqrt{\frac{n}{n-1}} - 1)opt + \binom{k}{2}$. This proves NP-hardness. Finally, as is a typical argument for strongly NP-hard problems, we remark that if there was an FPTAS for this MSD_k instance, by choosing the error parameter ε sufficiently small we could transform it into an exact algorithm. □

We present in Table 2.1 the current state of the art in approximation and inapproximability results for MSD_k, over several distance classes. The table showcases some of the results presented in this thesis, which greatly improve our understanding of this problem. However, we notice that the NP-hardness of the problem remains unproven for distances induced by a norm in fixed dimension; and the same goes for the existence of a PTAS for variable-dimension distances in ℓ_p, for $2 < p < \infty$.

Local search in related geometric clustering problems

As this thesis showcases the use of local search[9] for a geometric optimization problem, it is worth mentioning other examples of such problems where this technique has been applied successfully. In particular, there have been recent breakthroughs in the approximability of the geometric clustering problems of *k-median* and *k-means*. Given a metric distance (X, d) and a number k, these problems are defined as

$$\min_{A \subset X,\ |A|=k} \sum_{b \in X} \min_{a \in A} d^{\alpha}(a, b), \tag{2.11}$$

where $\alpha = 1$ for k-median, and $\alpha = 2$ for k-means.

For the general metric case, it is NP-hard to approximate k-median and k-means within a factor of $1 + \frac{2}{e}$ and $1 + \frac{3}{e}$, respectively [70, 80]. Local search has been amply studied for metric k-median: it was shown to yield constant-factor bi-criteria approximations [89, 34], and finally Arya et al. [10] proved that it gives a $(3+\varepsilon)$-approximation, and that this bound is tight.[10] For metric k-means, Gupta and Tangwongsan [72] showed that local search achieves a $(25+\varepsilon)$-approximation.

For the Euclidean case with variable dimension, neither problem admits a PTAS [73, 14]. Whether Euclidean k-means is APX-hard was an open problem for a long time, recently answered positively by Awasthi et al. [14]. Moreover, Kanungo et at. [81] proved that local search offers the improved ratio of $(9+\varepsilon)$ for Euclidean k-means, and this is the current best approximation ratio for the problem.

Finally, if the dimension is fixed, k-median is known to admit PTASs [75, 76, 87]. The question of whether the same is true for k-means remained open for many years. In a recent development, Cohen-Addad et al. [40] finally provided a PTAS for fixed-dimension Euclidean k-means. This is achieved by local search with a neighborhood of size $1/\varepsilon^{O(1)}$. In fact, they prove that the same algorithm provides a PTAS for k-median and for the general problem (2.11) with any fixed $\alpha \geq 1$. In independent and simultaneous work, Friggstad et al. [61] obtained similar results, extending the PTAS even to k-means over general metric distances with bounded doubling dimension (see [71]).

[9] An introduction to this technique can be found in Section 4.2

[10] Better approximation ratios with other techniques are currently known, see [93, 26]

3 MSD via convex programming

3.1 Chapter overview

In this chapter, we consider the max-sum dispersion problem (MSD) exclusively over distances of negative type, and we describe for it approximation algorithms obtained with the classical approach of *randomized rounding*, introduced by Raghavan and Thompson [107].

In the case of a linear objective function over X, and very basic linear constraints, the idea behind randomized rounding is simple enough: define the optimization problem in terms of a Boolean linear program; relax it into a linear program over the unit cube; solve this relaxation to obtain a fractional solution x^* with a higher objective value than the optimal integral solution; and finally, use the component values of x^* to define an independent randomized rounding procedure – for each $a \in X$, independently sample a with probability x^*_a. The sampled set $S \subset X$ has the same objective value as x^* in expectation, and each constraint is satisfied or almost satisfied with high probability, due to concentration bounds. If the number of constraints is low, then the union bound is applied to guarantee that no constraint is violated by a large margin, with good probability.

This “solve the relaxation, then round" technique is standard in combinatorial optimization, and readily yields algorithms with good approximation ratios for a large number of problems. Furthermore, there has been a continuous effort in the literature to extend its applicability. For instance, the union bound will not give meaningful results in the case of a large (exponential) number of constraints; a corresponding alternative is that of dependent rounding: the fractional solution x^* is pushed towards an integral point, usually in an iterative process, so that its feasibility is guaranteed along each step. Yet, if this process is performed deterministically, the concentration bounds are lost, and hence if the rounding process fails to guarantee the satisfiability of a single constraint, the rounded solution can be arbitrarily far from feasible. In the case of a matroid constraint, Chekuri, Vondrák and Zenklusen [35, 36] present a *dependent randomized rounding* framework. This procedure outputs a random integral point which is guaranteed to lie in the matroid polytope, and with a distribution that still observes concentration bounds, allowing to tackle extra constraints.

If the objective function is non-linear, this framework does not extend in general, as it may be NP-hard to solve the corresponding relaxation of the problem. Nevertheless, if the relaxed problem corresponds to the maximization of a concave function over a convex region, then it is known as a *convex program*, and it is tractable. The extended dispersion function $d(x)$, which is quadratic, is neither concave nor convex, but if it comes from a negative-type distance, we will prove that the relaxed problem can be *convexified* – that is, it can be restated and solved in terms of convex programming.

Contribution and organization

Recall that the general max-sum dispersion (MSD) problem is $\max\{d(A) : A \in \mathscr{F}\}$, where $d(A) = \frac{1}{2}\sum_{a,b\in A}$ and $\mathscr{F} \subset 2^X$ is a collection of feasible subsets of X. In this chapter, we deal with a collection $\mathscr{F}$ defined by linear constraints in general, and matroid and knapsack constraints in particular. We focus exclusively on distances of negative type.

In Section 3.2, we relax the problem into a quadratic program, and then convexify it. More concretely, we prove that by solving convex sub-programs, we are able to find a fractional feasible solution with a higher objective value than that of the optimal integral solution.

In Section 3.3, for the case of a matroid constraint of rank k, we complement the point above with a deterministic rounding procedure, reminiscent of pipage rounding. The resulting algorithm provides an approximation ratio of $1 - O(\frac{\log k}{k})$, and hence implies a PTAS for the problem. We also prove that the result immediately extends to a combination with linear functions in the objective (MSD + l).

Finally, in Section 3.4, we randomize the previous deterministic rounding procedure, to obtain an algorithm with the same approximation guarantee in expectation, and offering concentration bounds as well. Consequently, we obtain a PTAS for the case of a matroid constraint and a constant number of knapsack constraints.

3.2 A relaxation for general linear constraints

A *quadratic program* is an optimization problem of the form

$$\min\left\{\frac{1}{2}x^T Qx + c^T x : \ x \in \mathbb{R}^n, \ Mx \le b\right\}, \tag{3.1}$$

where $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $c \in \mathbb{R}^n$, $M \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. The objective function $f(x) = \frac{1}{2}x^T Qx + c^T x$ is convex if and only if Q is positive semidefinite, in which case the problem is a *convex quadratic program*.

There are several efficient algorithms related to convex quadratic programs. In particular, the *ellipsoid method* [83] can be used to solve this problem in polynomial time (see e.g. [90]). This remains true even if the set of constraints defined by $Mx \le b$ is not explicitly given, but the separation problem over the polyhedron defined by $Mx \le b$ can be solved in polynomial time

(see [69]). In this case, the running time is polynomial in the input size of Q and c and the largest binary encoding length of a coefficient on M or b.

A matroid polytope, for instance, in general cannot be represented in a compact way under the form $Mx \leq b$, but its separation problem can be solved efficiently, provided that an independence oracle is given [69]. In that case, the largest encoding length of the numbers in the previously mentioned description of the matroid polytope is $O(\log n)$. Hence, a convex quadratic program over a matroid polytope con be solved in polynomial time.

Problem formulation and relaxation

We consider the negative-type MSD problem with general linear constraints. This means that the family $\mathscr{F} \subset 2^X$ of feasible solutions can be represented as $\mathscr{F} = \{S \subset X : M(\mathbb{1}^S) \leq b\}$, for some $M \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We assume that the separation problem over $Mx \leq b$ can be solved in polynomial time, and that the encoding lengths of coefficients in M and b are small.[1] This scenario covers the cases of a matroid constraint, a cardinality constraint (uniform matroid), knapsack constraints, among others. Let $O \in \mathscr{F}$ the optimal solution, with value $d(O) = opt$. If D is the distance matrix, we can use the extended dispersion function $d(x) = \frac{1}{2}x^T Dx$ (see Section 2.3.2), to write the problem as

$$\max\left\{\frac{1}{2}x^T Dx : \ x \in \{0,1\}^X, \ Mx \leq b\right\}. \tag{3.2}$$

We now remove the integrality constraints, and relax it into a quadratic program:

$$\max\left\{\frac{1}{2}x^T Dx : \ x \in [0,1]^X, \ Mx \leq b\right\}. \tag{3.3}$$

Notice that the set of feasible solutions in (3.3) contains all feasible solutions in (3.2), so the value of the optimal solution in (3.3) can only increase. Our (naive) plan of attack is to solve this quadratic program *exactly*, and obtain a feasible fractional point x^* with $d(x^*) \geq opt$. A rounding procedure can then be applied to it, to produce a feasible integer point with guaranteed value.

Convexification

Unfortunately, (3.3) is not a convex quadratic program, as D is not negative semidefinite.[2] However, we know from Remark 2.5 that D is conditionally negative semidefinite, with exactly one positive eigenvalue. This means that, even though we cannot solve (3.3) exactly,[3] we can easily convexify it by restricting its domain to certain hyperplanes. Hence, we can cover up all integer points in (3.3) with *slices*, and solve these convex sub-programs exactly. This will be enough to compute a (suboptimal) point x^* in (3.3) that still has the property $d(x^*) \geq opt$.

[1] If the encoding lengths are large, in many cases our framework will still work, at a cost of a small decrease in the approximation ratios obtained.

[2] A maximization problem that is written as in (3.1), where Q is negative semidefinite, is equivalent to a convex quadratic program, via a standard transformation.

[3] A non-convex program as (3.1) is NP-hard, even if Q has a single negative eigenvalue [101].

Lemma 3.1. *If (X,d) is of negative type, then $d(x) = \frac{1}{2}x^T Dx$ is a concave function over the domain $\{x \in \mathbb{R}^X : \mathbb{1}^T x = \alpha\}$, for each fixed $\alpha \in \mathbb{R}$.*

Proof. The statement is equivalent to saying that, for any two points $x, y \in \mathbb{R}^X$ such that $\mathbb{1}^T x = \mathbb{1}^T y$, the function $d(\cdot)$ is concave over the line connecting x and y. Or equivalently, that for any point $x \in \mathbb{R}^X$ and vector v with $\mathbb{1}^T v = 0$, the function $f(\lambda) := d(x + \lambda v)$ is concave over $\lambda \in \mathbb{R}$. As $\mathbb{1}^T v = 0$ and the distance is of negative type, we have $d(v) \leq 0$. With the use of properties (2.5), the function $f(\lambda)$ can be written as

$$f(\lambda) = d(x + \lambda v) = d(x) + \lambda d(x, v) + \lambda^2 d(v),$$

and hence its second derivative is $\frac{d^2}{(d\lambda)^2} f(\lambda) = 2d(v) \leq 0$. This proves the statement. □

We remark that convexifications have proved useful before in the design of approximation algorithms, see for example [116]. It is also worth mentioning that Tamir [119] proves a statement similar to Lemma 3.1 for weighted tree distances (which are of negative type, see Proposition 2.12), and uses this result to show that MSD_k has an exact algorithm for these distances.

Theorem 3.2. *Consider the negative-type* MSD *problem* (3.2) *with general linear constraints for which the separation problem can be solved efficiently. One can compute a fractional point x^* that is feasible in the relaxation* (3.3)*, and such that $d(x^*) \geq opt$, in time polynomial in the input size and the maximal binary encoding length of any coefficient in M and b.*

Proof. Notice that for any integer point in (3.3), $\mathbb{1}^T x$ will be an integer smaller than n. Using the ellipsoid method, we solve each of the following n convex quadratic programs exactly.[4]

$$\max\left\{\frac{1}{2}x^T Dx : \; x \in [0,1]^X, \; Mx \leq b, \; \mathbb{1}^T x = \alpha\right\}, \qquad \text{for } \alpha = 1, \cdots, n.$$

We obtain the optimal solutions $x^1, \cdots, x^n$, and we define $x^* = \text{argmax}\{d(x^\alpha) : \; 1 \leq \alpha \leq n\}$. Clearly, x^* is feasible in (3.3). For the optimal integral solution O, we know that $\mathbb{1}^O$ is feasible for the program with $\alpha = |O|$, hence $opt = d(\mathbb{1}^O) \leq d(x^\alpha) \leq d(x^*)$. □

Remark 3.3. At this point, for the cardinality-constrained case (MSD_k), a standard randomized rounding technique [107] readily provides a PTAS, as we now explain. The point x^* obtained in the previous theorem satisfies the constraint $\mathbb{1}^T x^* \leq k$, and from it we sample a set $A \subset X$ by selecting each element $a \in X$ independently with probability $(1-\varepsilon)x_a$. The resulting set has an expected value $\mathbb{E}[d(A)] = (1-\varepsilon)^2 d(x^*) > (1-2\varepsilon)opt$, and its cardinality will be sharply concentrated around its expected value $\mathbb{E}[|A|] = (1-\varepsilon)\mathbb{1}^T x^* \leq (1-\varepsilon)k$, due to Chernoff-type concentration bounds. Hence, for large enough k, A will be feasible with high probability. However, this technique fails for more complex linear constraints, such as a matroid constraint.

[4] See [90, 69] for details on why exact solutions can be obtained here, without an additive error that is typical for many convex optimization techniques.

3.3 Deterministic rounding for a matroid constraint

We consider now the case of negative-type MSD constrained by a matroid. Building up from Theorem 3.2, we describe a deterministic rounding procedure, which will lead to a PTAS. This procedure has similarities with pipage rounding [5, 27] and swap rounding [36], in the sense that it iteratively modifies at most two components of the fractional point, until an integer point is obtained. However, it differs substantially from these procedures, as we deal with a quadratic objective function, and we must accept a certain loss in the objective value due to rounding.[5] By carefully selecting the two components to be modified in each iteration, and using once again the properties of negative-type distances, we manage to establish a small bound for this loss. Relevant to our scenario is also a procedure by Makarychev et al [95], which is based on swap rounding, and provides concentration bounds for polynomial objective functions. However, these bounds are not strong enough for our purposes.

Recall that for a vector $x \in \mathbb{R}^X$ and a set $S \subset X$, we define the restricted vector $x^S \in \mathbb{R}^X$ by $x^S_a = x_a$ if $a \in S$, and 0 otherwise. The input of our problem comprises a negative-type distance matrix D, and a matroid $(X, \mathscr{I})$ of rank k, which is assumed to be given by an independence oracle. The feasible polytope in the relaxation (3.3) corresponds to the matroid polytope

$$P(\mathscr{I}) = \left\{ x \in \mathbb{R}^X_+ : \ \mathbb{1}^T x^S \leq r(S), \ \forall S \subset X \right\},$$

where $r(\cdot)$ is the matroid rank function. As the extended dispersion function $d(x)$ is monotone, the optimal fractional solution x^* must be on the base polytope[6] $P_B(\mathscr{I}) = P(\mathscr{I}) \cap \{x : \mathbb{1}^T x = k\}$, so Theorem 3.2 will actually find this exact point by solving a single convex quadratic program. We describe now a deterministic iterative rounding algorithm, that takes x^* as input, and outputs an integral point x^0 on the base polytope, with $\frac{d(x^*)-d(x^0)}{d(x^*)} = O(\frac{\log k}{k})$. It will imply the following result, which in turn implies a PTAS (see Remark 2.1).

Theorem 3.4. *There exists a deterministic algorithm for negative-type* MSD *with a matroid constraint of rank k, that outputs in polynomial time a basis S with $d(S) \geq \left(1 - \frac{4+2\ln k}{k}\right) opt$. Therefore, this problem admits a polynomial-time approximation scheme.*

3.3.1 The rounding procedure

In the remainder of this section, for any vector $x \in P(\mathscr{I})$ we ignore the elements $a \in X$ with $x_a = 0$, and we assume without loss of generality that x has no zero components. We call an element $a \in X$ *integral* or *fractional* (with respect to x), if $x_a = 1$ or $x_a < 1$, respectively; and we call a set $S \subset X$ *tight* or *loose*, respectively, if $\mathbb{1}^T x^S = r(S)$ or $\mathbb{1}^T x^S < r(S)$. We will need the following result about faces of the matroid polytope, which is a well-known consequence of combinatorial uncrossing (see [64], or [114, Section 44.6c in Volume B]).

[5] Pipage and swap rounding are typically applied in settings where the objective value is preserved in expectation.

[6] Using standard techniques from matroid optimization (see [114, Volume B]), for any point $y \in P(\mathscr{I})$ one can find a point $z \in P_B(\mathscr{I})$ satisfying $z \geq y$ component-wise. And in this case it is clear that $d(z) \geq d(y)$. Hence the optimal solution must be on the base polytope.

Lemma 3.5. *Let $x \in P_B(\mathscr{I})$ be a vector with no zero components, and let*

$$\emptyset = S_0 \subsetneq S_1 \subsetneq \cdots \subsetneq S_p = X$$

be an inclusion-wise maximal chain of tight sets with respect to x. Then, the polytope

$$P(\mathscr{I}) \cap \left\{ y : \mathbb{1}^T y^{S_l} = r(S_l) \ \text{ for } l = 1, \ldots, p \right\}$$

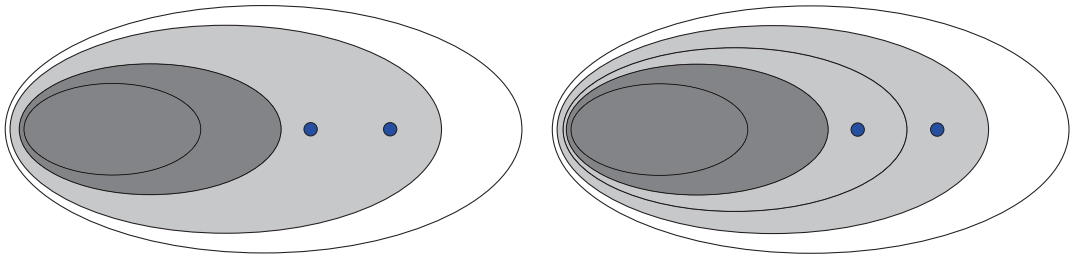
defines the minimal face of $P(\mathscr{I})$ that contains x. In other words, all other tight sets with respect to x are implied by the ones in the chain.

Given a point $x \in P_B(\mathscr{I})$, one can efficiently find a maximal chain of tight sets $(S_l)_{l=1}^p$ as described above. The algorithm will run in iterations; in each iteration it will change two components of x in such a way that x does not leave the minimal face of the matroid polytope on which it lies. This condition is ensured by preserving the structure of the chain. Moreover, in each step the change in x will reduce the dimension of this minimal face. That in turn ensures that x is ultimately rounded into a vertex of the polytope, in a linear number of steps. And finally, the careful selection of the two changing coefficients will bound the total loss.

Consider a point x on the base polytope, and a maximal chain of tight sets $(S_l)_{l=1}^p$. For each $l = 1, \cdots, p$, we define the set $R_l = S_l \setminus S_{l-1}$ – we call these sets *rings*. The rings form a partition of X, their weights $\mathbb{1}^T x^{R_l} = r(S_l) - r(S_{l-1})$ are strictly positive integers whose sum is k, and each ring R_l consists either of a single integral element, or of at least 2 elements, all fractional. This is because whenever $a \in R_l$ is integral, the set $S_{l-1} + a$ is tight, hence it can be added to the chain. We call the rings integral or fractional, accordingly.

We start with $x = x^*$, a chain as above, and a corresponding partition of X into rings. We perform the following process in iterations, and stop when all elements are integral. Among all fractional rings, and all pairs of fractional elements within the same ring, select the pair a, b that minimizes the term $x_a x_b d(a, b)$. We perturb vector x by adding to x_a and subtracting from x_b a certain quantity ε. The dispersion $d(x) = \frac{1}{2} x^T D x$ is linear in ε except for the term $x_a x_b d(a, b)$, hence we can select the sign of ε so that the value of $d(x) - x_a x_b d(a, b)$ does not decrease. We assume without loss of generality that this choice is $\varepsilon > 0$, so x_a is increasing and x_b decreasing. Notice that the weights of all the rings stay constant in this process, and thus all sets in the chain stay tight. And by Lemma 3.5, all tight sets stay tight. We increment ε until a new tight constraint appears. If the constraint corresponds to x_b becoming zero, we erase that element and end the iteration step. Otherwise, a previously loose set $S \subset X$ becomes tight, and S must contain a but not b, as otherwise its weight $\mathbb{1}^T x^S$ would not increase during this process. If the ring containing a and b is $R_l = S_l \setminus S_{l-1}$, then the set $S' = (S \cup S_{l-1}) \cap S_l$ is also tight,[7] and it also contains a but not b, so $S_{l-1} \subsetneq S' \subsetneq S_l$ (see Figure 3.1). We add S' to the chain, update the list of rings, and end the iteration step.

[7] This follows from the uncrossing property: if A and B are tight sets, then $A \cap B$ and $A \cup B$ are tight as well. This property is a consequence of the submodularity of the matroid rank function r.



(a) The fractional ring containing the pair a, b that minimizes $x_a^m x_b^m d(a,b)$.

(b) Elements a and b are separated by a new tight set that fits in the chain structure.

Figure 3.1: The refinement of a fractional ring in an iteration of the rounding procedure.

Analysis

We now analyze this algorithm, and prove Theorem 3.4. At any stage of the algorithm, if q is the number of fractional rings, f is the number of fractional elements, and m is the number of iterations remaining, then $f - q \geq m$. This is because the value $f - q$ can never be negative, and it decreases in each iteration. Either f decreases, or q increases, or q decreases by 1 but f decreases by 2 (any disappearing fractional ring has 2 disappearing fractional elements). This implies in particular that the total number of iterations is at most n, hence the algorithm runs in polynomial time.

Suppose there are M iterations, enumerated in reverse order by m. We add a superscript m to all variables to signify their value at this stage. This way, x^0 is the integral output vector, x^1 is the vector at the beginning of the last iteration, and so on until $x^M = x^*$. It is clear that all vectors x^m stay inside $P(\mathscr{I})$, and their weights $\mathbb{1}^T x^m$ remain unchanged, hence they are all on the base polytope. Furthermore, x^0 is integer-valued, so it is the characteristic vector of a basis in the matroid. For $1 \leq m \leq M$, define $\text{loss}^m = d(x^m) - d(x^{m-1})$, hence the total additive loss incurred in the rounding algorithm is $\sum_{m=1}^{M} \text{loss}^m$. We postpone for a moment the proof of the following claim.

Lemma 3.6. *The loss in iteration m is bounded by*

$$\text{loss}^m \leq \min\left\{\frac{2}{m^2}, \frac{2}{km}\right\} \cdot d(x^m).$$

The total additive loss incurred by the algorithm is

$$\begin{aligned} d(x^*) - d(x^0) = \sum_{m=1}^{M} \text{loss}^m &\leq \sum_{m=1}^{M} \min\left\{\frac{2}{m^2}, \frac{2}{km}\right\} \cdot d(x^m) \\ &\leq \sum_{m=1}^{\infty} \min\left\{\frac{2}{m^2}, \frac{2}{km}\right\} \cdot d(x^*) = \left(\sum_{m=1}^{k} \frac{1}{km} + \sum_{m>k} \frac{1}{m^2}\right) 2d(x^*) \\ &\leq \left(\frac{1+\ln k}{k} + \frac{1}{k}\right) 2d(x^*) = \frac{4+2\ln k}{k} d(x^*), \end{aligned}$$

where we used the inequalities $\sum_{m=1}^{k} \frac{1}{m} \leq 1+\ln k$ and $\sum_{m>k} \frac{1}{m^2} \leq \frac{1}{k}$. In summary, the algorithm finds a basis with dispersion

$$d(x^0) \geq \left(1 - \frac{4+2\ln k}{k}\right) d(x^*) \geq \left(1 - \frac{4+2\ln k}{k}\right) opt.$$

This completes the proof of Theorem 3.4.

Proof of Lemma 3.6. We fix a value for m and analyze the respective iteration, and we skip the superscripts m to simplify notation. Let $F \subset X$ be the set of fractional elements for the current point x. Therefore, $x^F = \sum_R x^R$, where the sum is over all fractional rings R. We apply inequality (2.7) multiple times over this decomposition of x^F, to obtain

$$\frac{d(x^F)}{\mathbb{1}^T x^F} \geq \sum_R \frac{d(x^R)}{\mathbb{1}^T x^R}.$$

If the pair $a, b \in X$ of fractional elements is chosen during this iteration, then $\text{loss} \leq x_a x_b d(a,b)$, and because of the way the pair is chosen, we have that $d(x^R) \geq \binom{|R|}{2}\text{loss} > (|R|-1)^2 \frac{\text{loss}}{2}$, for every fractional ring R. Thus,

$$\begin{aligned} d(x^F) &\geq \left(\mathbb{1}^T x^F\right) \sum_R \frac{d(x^R)}{\mathbb{1}^T x^R} > \frac{\text{loss}}{2} \left[\sum_R \mathbb{1}^T x^R\right] \left[\sum_R \frac{(|R|-1)^2}{\mathbb{1}^T x^R}\right] \\ &\geq \frac{\text{loss}}{2} \left[\sum_R (|R|-1)\right]^2 = \frac{\text{loss}}{2}(f-q)^2 \geq \frac{m^2}{2}\text{loss}; \end{aligned}$$

where in the second line we used a Cauchy-Schwarz inequality. We obtain the bound

$$\text{loss} \leq \frac{2}{m^2} d(x^F).$$

The first claimed inequality now follows because $d(x^F) \leq d(x)$. For the second one, we start by noticing that the weight $\mathbb{1}^T x^F$ decreases by at most 1 in each iteration, which means that $\mathbb{1}^T x^F \leq m$. Using once again inequality (2.7), we have

$$\frac{d(x)}{\mathbb{1}^T x} \geq \frac{d(x^F)}{\mathbb{1}^T x^F},$$

and so $d(x^F) \leq \frac{\mathbb{1}^T x^F}{\mathbb{1}^T x} d(x) \leq \frac{m}{k} d(x)$. This proves the second inequality. □

Notice that, towards the end of the proof of Theorem 3.4, we used Lemma 3.6, and the fact that $d(x^m) \leq d(x^*)$ for all m, because x^* is the optimal point on the base polytope $P_B(\mathcal{I})$. We argue below that this last condition is not needed. In other words, if we perform the rounding procedure starting from an arbitrary fractional solution in $P_B(\mathcal{I})$, the bound on the loss in the objective value still holds. This observation will be of use when we deal with knapsack constraints.

Lemma 3.7. *Starting from any point $x^* \in P_B(\mathscr{I})$, and for a distance of negative type, the previous rounding procedure returns an integral point $x^0 \in P_B(\mathscr{I})$ such that*

$$d(x^*) - d(x^0) \leq \frac{4 + 2\ln k}{k} d(x^*).$$

Proof. Let $x^* = x^M, x^{M-1}, \cdots, x^1, x^0$ be the distinct values taken by point x in the rounding iterations, with labels as before. We use the short-hand $\lambda^m = \min\{\frac{2}{m^2}, \frac{2}{km}\}$. Let m' be the lowest index such that $d(x^{m'}) \geq d(x^*)$. If $m' = 0$, then $d(x^*) - d(x^0) \leq 0$, and the claim follows trivially. Otherwise, by Lemma 3.6,

$$d(x^{m'-1}) = d(x^{m'}) - loss^{m'} \geq (1 - \lambda^{m'}) d(x^{m'}) \geq (1 - \lambda^{m'}) d(x^*).$$

Therefore, $d(x^*) - d(x^{m'-1}) \leq \lambda^{m'} d(x^*)$. Now, for all lower indexes $0 < m < m'$, we have

$$d(x^m) - d(x^{m-1}) = \text{loss}^m \leq \lambda^m d(x^m) \leq \lambda^m d(x^*),$$

by Lemma 3.6 and the definition of m'. Hence,

$$d(x^*) - d(x^0) \leq \sum_{m=1}^{m'} \lambda^m d(x^*) \leq \sum_{m=1}^{\infty} \lambda^m d(x^*),$$

and the proof continues as in Theorem 3.4. □

3.3.2 Integrality gap

To complement our approximation result, we remark that the integrality gap of the convex quadratic program $\max\{\frac{1}{2} x^T D x : \ x \in P_B(\mathscr{I})\}$ considered above is bounded by $1 - \frac{1}{k}$, a bound that almost matches the approximation ratio of our algorithm. Consider the matrix D with all off-diagonal entries equal to 1, which defines a negative-type distance by Lemma 2.10, and a uniform matroid constraint corresponding to the polytope $\{x \in [0,1]^X : \ \mathbb{1}^T x = k\}$. Any k-set is an optimal integral solution with value $opt = \binom{k}{2}$; but the fractional point $x^* = \frac{k}{n}\mathbb{1}$ is feasible and has value $\frac{k^2}{n^2}\binom{n}{2}$. Hence,

$$\frac{opt}{d(x^*)} = \frac{n^2\binom{k}{2}}{k^2\binom{n}{2}} = \frac{k-1}{k}\frac{n}{n-1} \to 1 - \frac{1}{k} \quad \text{as } n \to \infty.$$

3.3.3 Combination with a linear function

We also remark that the previous approximation easily extends to the combination of the dispersion function with a linear function, i.e., to MSD + l. The new objective function can be written as

$$g(x) = \frac{1}{2} x^T D x + w^T x,$$

for a non-negative weight vector w. The extra linear term does no change the concavity of the objective function, hence Lemma 3.1 and Theorem 3.2 are still valid for this problem. Moreover, $g(x)$ is still monotone, which means that the optimal fractional point x^* will be on

the base polytope $P_B(\mathscr{I})$, and we can find it exactly. In each iteration of this section's rounding algorithm, $g(x) - x_a x_b d(a,b)$ is linear in ε, so we can bound the loss of value of $g(x)$ during this iteration by $x_a x_b d(a,b)$, as before. Hence, the previous analysis still holds and shows that the total loss is very small, even when compared to $d(x^*)$ and ignoring the linear contribution to the objective $g(x^*)$. Therefore, the same approximation ratio holds for this more general problem.

3.4 Randomized rounding for further constraint types

We present in this section a natural randomization of the previous rounding procedure. A randomized rounding algorithm can deal with further constraint through concentration bounds. In particular, this will lead to a randomized PTAS, i.e., a PRAS, for negative-type MSD constrained by a matroid and a constant number of knapsack constraints.

3.4.1 The modified rounding procedure

We define a randomized version of the rounding algorithm in Section 3.3.1 for the same framework, namely a matroid constraint. This is a standard randomization of pipage rounding that is known in contexts with linear objective functions (see [35, 36]). Given an input fractional point x^* in the base polytope, the new algorithm returns a random point x^0, which is the characteristic vector of a basis, whose expected objective value has the same guarantee as before. And in addition, it will observe Chernoff-type concentration bounds.

The setup for the iterations remains unchanged. Let $x \in P_B(\mathscr{I})$ be the current fractional point. We select, among all fractional rings and all pairs within the same ring, the pair a, b that minimizes the term $x_a x_b d(a,b)$. Define the vector $v = \mathbb{1}^a - \mathbb{1}^b$ and the coefficients

$$\varepsilon_+ = \max\{\varepsilon \in \mathbb{R} : x + \varepsilon v \in P(\mathscr{I})\} \quad \text{and} \quad \varepsilon_- = \max\{\varepsilon \in \mathbb{R} : x - \varepsilon v \in P(\mathscr{I})\}.$$

With prob. $\frac{\varepsilon_-}{\varepsilon_+ + \varepsilon_-}$, update $x \leftarrow x + \varepsilon_+ v$; else, with prob. $\frac{\varepsilon_+}{\varepsilon_+ + \varepsilon_-}$, update $x \leftarrow x - \varepsilon_- v$.

In words, the point x will move along the line parallel to v, just as before, but the direction is chosen at random. Coefficients ε_+ and ε_- represent the amounts by which it will move in either direction. Notice that both coefficients are strictly positive, because x is moving inside the minimal face of $P(\mathscr{I})$ containing it, and by the definition of this face, x starts in its interior. On the other hand, we have the upper bounds $\varepsilon_+ \le x_b$ and $\varepsilon_- \le x_a$ due to the non-negativity constraints. The probabilities are chosen so that the marginals of the new random point are given by x. Indeed, if the new point is x',

$$\mathbb{E}[x'] = \frac{\varepsilon_-}{\varepsilon_+ + \varepsilon_-}(x + \varepsilon_+ v) + \frac{\varepsilon_+}{\varepsilon_+ + \varepsilon_-}(x - \varepsilon_- v) = x. \tag{3.4}$$

By linearity of expectation, this implies for the output point x^0 that $\mathbb{E}[x^0] = x^*$.

We study now the expected loss incurred in the objective value during an iteration. For convenience, we pre-multiply it by $-(\varepsilon_+ + \varepsilon_-)$.

$$\begin{aligned}
-(\varepsilon_+ + \varepsilon_-)\mathbb{E}[\text{loss}] &= \varepsilon_-[d(x+\varepsilon_+ v) - d(x)] + \varepsilon_+[d(x-\varepsilon_- v) - d(x)] \\
&= \varepsilon_-[d(x, \varepsilon_+ v) + d(\varepsilon_+ v)] + \varepsilon_+[d(x, -\varepsilon_- v) + d(-\varepsilon_- v)] \\
&= \varepsilon_-[\varepsilon_+ d(x,v) + \varepsilon_+^2 d(v)] + \varepsilon_+[-\varepsilon_- d(x,v) + \varepsilon_-^2 d(v)] \\
&= (\varepsilon_+ + \varepsilon_-)\varepsilon_+\varepsilon_- d(v),
\end{aligned}$$

where we used the properties in (2.5). For the term $d(v)$ we have the identity

$$d(v) = d(\mathbb{1}^a - \mathbb{1}^b) = -d(\mathbb{1}^a, \mathbb{1}^b) = -d(a,b),$$

hence

$$\mathbb{E}[\text{loss}] = -\varepsilon_+\varepsilon_- d(v) = \varepsilon_+\varepsilon_- d(a,b) \le x_a x_b d(a,b).$$

In each iteration, the expected loss in the objective value is bounded by the same amount that was used as bound in the analysis of the deterministic algorithm. Therefore, using linearity of expectation on the sum of losses over all iterations, we can prove the same guarantee in expectation.

Lemma 3.8. *Consider negative-type* MSD *constrained by a matroid* $(X, \mathscr{I})$ *of rank* k*. If we apply the aforementioned randomized rounding procedure to a point* x^* *in the matroid base polytope* $P_B(\mathscr{I})$*, the output is a random characteristic vector* x^0 *of a basis, with*

$$\mathbb{E}[d(x^0)] \ge \left(1 - \frac{4 + 2\ln k}{k}\right) d(x^*), \quad \textit{and}$$

$$\mathbb{P}\left[d(x^*) - d(x^0) \ge c\left(\frac{4+2\ln k}{k}\right) d(x^*)\right] \le \frac{1}{c}, \quad \textit{for any } c \ge 1.$$

Proof. The first part follows from the argument above and Lemma 3.7.

Now, define the variable $z = \max\{0, d(x^*) - d(x^0)\}$. The proof of Lemma 3.7 can be easily adapted to show that the expected value of this non-negative variable is $\mathbb{E}[z] \le \left(\frac{4+2\ln k}{k}\right) d(x^*)$. Then, the second statement is simply an application of Markov's inequality over z. □

3.4.2 Concentration bounds for linear constraints

Chekuri, Vondrák and Zenklusen [36] prove that this randomized pipage rounding procedure, whose marginals observe identity (3.4), produces a random variable with *negatively correlated* components. This means that, if x^0 is the output point for input x^*, then for any set $S \subset X$ we have

$$\mathbb{P}\left[\prod_{a\in S} x_a^0 = 1\right] \le \prod_{a\in S} x_a^* \quad \text{and} \quad \mathbb{P}\left[\prod_{a\in S}(1-x_a^0) = 1\right] \le \prod_{a\in S}(1-x_a^*).$$

And therefore, x^0 fulfills the Chernoff-type concentration bounds.

Lemma 3.9 (see [36])**.** *Let x^0 be the random characteristic vector of a matroid basis, obtained from an input point $x^* \in P_B(\mathscr{I})$ by the previous randomized pipage rounding procedure. For a fixed vector of coefficients $m \in [0,1]^X$, define the variable $Y = m^T x^0 = \sum_{a \in X} m_a x^0_a$, whose expectation is $\mathbb{E}[Y] = m^T x^*$ because the procedure preserves marginals in expectation. The following bounds hold for Y.*
If $\delta \geq 0$ and $\mu \geq \mathbb{E}[Y]$, then

$$\mathbb{P}[Y \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu,$$

which, for $\delta \in [0,1]$, can be simplified to

$$\mathbb{P}[Y \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}.$$

And if $\delta \in [0,1]$ and $\mu \leq \mathbb{E}[Y]$, then

$$\mathbb{P}[Y \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}.$$

These last two lemmas show that this procedure is likely to return a solution with large objective value, that does not violate knapsack constraints by much. This is already sufficient to deal with knapsack constraints that are *soft*, where a slight violation of constraints is acceptable. In the following, we show that a proper PRAS can be obtained for a constant number of knapsack constraints along with one matroid constraint.

3.4.3 Dealing with knapsack constraints

As our randomized rounding procedure is a special case of pipage rounding, we can follow existing approaches to deal with knapsack constraints. It is shown in [35] that a procedure that observes concentration bounds can deal with knapsack constraints via a pre-processing step. This step guesses a subset $G \subset O$ of constant size containing “valuable" elements of the optimal solution O, and then removes from the ground set a subset $Q \subset X \setminus G$ of elements with large knapsack weights.

To obtain a strong approximation guarantee, one needs to prove that the potential contribution of the deleted subset Q towards the optimal solution is small. For linear (and more generally submodular) objective functions, this is easily achieved by making sure that the guessed set G contains the elements of O with highest objective value. As we deal with a quadratic function, this intuition becomes opaque, because there is no intrinsic objective value in individual elements. Still, it is not difficult to adapt this idea to our dispersion function, as we prove in the next lemma.

Lemma 3.10. *For any distance space (X,d), let S be an arbitrary subset of X, and let $q \in \mathbb{Z}_+$ and $0 < \varepsilon < 1$. Then, there exists a set $G \subset S$ of size at most $\lceil \frac{2q}{\varepsilon} \rceil$, such that for any set $Q \subset S \setminus G$ of size at most q we have*

$$d(S \setminus Q) \geq (1-\varepsilon)\, d(S).$$

Algorithm 3.1: Randomized rounding over a matroid constraint and knapsack constraints.

Set the constants $q = \lceil \frac{9r}{\varepsilon^2} \ln \frac{r}{\varepsilon} \rceil$ and $\kappa = \left\lceil \frac{2q}{\varepsilon} \right\rceil$.
Find (via exhaustive search) the best feasible set S' of size $\leq \kappa + q + \lceil \frac{1}{\varepsilon} \rceil$.
Guess a set $G \subset O$ of size at most κ that fulfills the properties of Lemma 3.10 for $S = O$.[8]
for $1 \leq i \leq r$, let $b_i = 1 - \sum_{a \in G} m_a^i$ be the remaining capacity of knapsack i, after subtracting the weights of the guessed elements in G. Define $b = (b_1 \cdots, b_r)^T$.
Remove from the ground set all elements $a \in X \setminus G$ such that there exists at least one knapsack i with $m_a^i \geq \frac{r}{q} b_i$. Let $N \subset X \setminus G$ be the set of these discarded elements.
Compute the optimal solution x^* of the following convex quadratic program

$$\max \left\{ \frac{1}{2} x^T D x : \; x \in P(\mathscr{I}),\; x^G = \mathbb{1}^G,\; x^N = 0,\; M x^{X \setminus (G \cup N)} \leq (1-\varepsilon) b,\; \mathbb{1}^T x \in \mathbb{Z} \right\}. \tag{3.5}$$

Let $(X, \mathscr{I}')$ be the matroid obtained from $(X, \mathscr{I})$ by truncating at rank $k' = \mathbb{1}^T x^*$. Use the randomized rounding procedure described above to round x^* to a base S of this matroid.
if S is feasible, **return** the better of S and S'; **else**, **return** S'.

Proof. We assume that $|S|$ is of size at least $\lceil \frac{2q}{\varepsilon} \rceil$, for otherwise the statement holds trivially by setting $G = S$. Let G consist of the $\lceil \frac{2q}{\varepsilon} \rceil$ elements $a \in S$ with highest cross-dispersion value $d(S, a)$. Then, for any $Q \subset S \setminus G$ of size at most q, we have

$$\begin{aligned} d(S) - d(S \setminus Q) &= d(S, Q) - d(Q) \leq d(S, Q) \\ &\leq \frac{|Q|}{|G|} d(S, G) \leq \frac{\varepsilon}{2} d(S, G) \\ &\leq \frac{\varepsilon}{2} d(S, S) = \varepsilon d(S), \end{aligned}$$

where the second inequality follows from the definition of G. This completes the proof. □

Consider a constant number r of knapsack constraints given by

$$\sum_{a \in X} m_a^i x_a \leq 1, \quad \forall i = 1, \cdots, r$$

where all coefficients satisfy $0 < m_a^i \leq 1$, and we assume without loss of generality that the coefficients on the right-hand side are all 1, since we can scale the constraints. For brevity, sometimes we write these constraints in matrix form $Mx \leq \mathbb{1}$, where $M = [m^1, \cdots, m^r]^T$.

Let $0 < \varepsilon \leq \frac{1}{3}$ be an error parameter. In the remainder of the section, we prove that Algorithm 3.1 returns with probability $1 - \tilde{O}(\frac{\varepsilon^2}{r})$ a feasible solution of value at least $(1 - 6\varepsilon) opt$. Here, $\tilde{O}(\cdot)$ hides terms logarithmic in r and $\frac{1}{\varepsilon}$. This immediately implies a PRAS for the problem. We shall favor simplicity in our analysis, and make no attempt to optimize the running time of the algorithm.

[8] This is done by exhaustive search, running the remaining steps of the algorithm for each such subset and returning the best obtained solution.

Theorem 3.11. *The negative-type* MSD *problem, constrained by a matroid and a constant number of knapsacks, admits a polynomial-time randomized approximation scheme.*

Analysis

We will use use the definitions and notation found in Algorithm 3.1. We assume that $|O| \geq \kappa + q + \lceil \frac{1}{\varepsilon} \rceil$; otherwise the algorithm finds and returns the optimal set $S' = O$. We also assume without loss of generality that $|G| = \kappa$.

Lemma 3.12.

$$d(x^*) \geq (1-5\varepsilon)opt.$$

Proof. After guessing G, each knapsack i has capacity b_i, hence it can fit at most $\frac{q}{r}$ elements of weight at least $\frac{r}{q} b_i$. This proves that the number of elements in O that are erased has a bound $|N \cap O| \leq q$. By Lemma 3.10,

$$d(O \setminus N) \geq (1-\varepsilon) d(O) = (1-\varepsilon)opt.$$

Furthermore, it is evident that the point $\mathbb{1}^G + (1-\varepsilon')\mathbb{1}^{O\setminus(G\cup N)}$ is feasible in the quadratic program (3.5), where ε' is the smallest constant larger than ε such that $(1-\varepsilon')|O \setminus (G \cup N)| \in \mathbb{Z}$. For this last set, we have the following bound on its size:

$$|O \setminus (G \cup N)| = |O| - |O \cap N| - |G| \geq |O| - \kappa - q \geq \frac{1}{\varepsilon}.$$

Therefore, $\varepsilon' \leq 2\varepsilon$. We conclude that

$$\begin{aligned} d(x^*) &\geq d\left(\mathbb{1}^G + (1-\varepsilon')\mathbb{1}^{O\setminus(G\cup N)}\right) \geq d\left((1-\varepsilon')\mathbb{1}^{O\setminus N}\right) \\ &= (1-\varepsilon')^2 d(O \setminus N) \geq (1-2\varepsilon)^2(1-\varepsilon)opt \\ &\geq (1-5\varepsilon)opt. \end{aligned}$$

□

Lemma 3.13. *With probability* $1 - \tilde{O}(\frac{\varepsilon^2}{r})$*, the dispersion of set* S *is* $d(S) \geq (1-6\varepsilon)opt$.

Proof. Consider the randomized rounding procedure that starts with x^* and outputs a basis S of the matroid $(X, \mathscr{I}')$. This procedure does not modify integer coordinates, hence $G \subset S$. In particular, if k' is the rank of this matroid, then $k' \geq |G| = \kappa$. We also have that $\kappa \geq \frac{2q}{\varepsilon}$, so $\varepsilon \geq \frac{2q}{\kappa}$. From Lemma 3.12, we obtain

$$\begin{aligned} \mathbb{P}[d(S) \leq (1-6\varepsilon)opt] &\leq \mathbb{P}[d(S) \leq (1-\varepsilon)(1-5\varepsilon)opt] \\ &\leq \mathbb{P}[d(S) \leq (1-\varepsilon)d(x^*)] = \mathbb{P}[d(x^*) - d(S) \geq \varepsilon d(x^*)] \\ &\leq \mathbb{P}\left[d(x^*) - d(S) \geq \frac{2q}{\kappa} d(x^*)\right]. \end{aligned}$$

Lemma 3.8 implies that this last probability is bounded by $\frac{\kappa}{2q}\frac{4+2\ln k'}{k'} = \frac{\kappa}{q} O(\frac{\log k'}{k'}) = \frac{\kappa}{q} O(\frac{\log \kappa}{\kappa}) = O(\frac{\ln \kappa}{q}) = \tilde{O}(\frac{1}{q}) = \tilde{O}(\frac{\varepsilon^2}{r})$. Here, we used the definition of q, and fact that $k' \geq \kappa$. □

Our final task is to give a bound on the probability that all knapsack constraints are satisfied. Concentration bounds guarantee that each individual constraint is satisfied with high probability, and then a union bound is used to guarantee their simultaneous satisfiability. We remark that the matroid constraint is satisfied with probability 1. The following lemma completes the proof of Theorem 3.11.

Lemma 3.14. *With probability at least* $1 - \frac{\varepsilon^2}{r}$, *$S$ satisfies all r knapsack constraints.*

Proof. Fix a knapsack $1 \le i \le r$, and define the vector $\bar{m} = \frac{q}{b_i r}(m^i)^{X\setminus G}$. The i-th knapsack constraint is satisfied by set S if and only if

$$\sum_{a\in S} m_a^i \le 1 \quad \Leftrightarrow \quad \sum_{a\in S\setminus G} m_a^i \le b_i \quad \Leftrightarrow \quad \frac{q}{b_i r}\sum_{a\in S\setminus G} m_a^i \le \frac{q}{r} \quad \Leftrightarrow \quad \bar{m}^T \mathbb{1}^S \le \frac{q}{r}.$$

Observe that the vector of coefficients $\bar{m}$ is in $[0,1]^X$, because $m_a^i \le \frac{r}{q} b_i$ for each $a \in X \setminus G$. Furthermore, since the randomized rounding procedure is unbiased, we have that

$$\mathbb{E}[\bar{m}^T \mathbb{1}^S] = \bar{m}^T x^* = \frac{q}{b_i r}(m^i)^T (x^*)^{X\setminus(G\cup N)} \le \frac{q}{b_i r}(1-\varepsilon) b_i = (1-\varepsilon)\frac{q}{r},$$

where the inequality is ensured by the feasibility of point x^* in the program (3.5). We apply Lemma 3.9 over the vector $\bar{m}$ and the variables $Y = \bar{m}^T \mathbb{1}^S$ and $\mu = (1-\varepsilon)\frac{q}{r}$, to conclude that the probability of violating the i-th knapsack constraint is at most

$$\mathbb{P}[Y \ge (1+\varepsilon)\mu] \le e^{-\frac{\mu\varepsilon^2}{3}} = e^{-(1-\varepsilon)\frac{\varepsilon^2 q}{3r}} \le e^{-(1-\varepsilon)3\ln\frac{r}{\varepsilon}} \le e^{-2\ln\frac{r}{\varepsilon}} = \frac{\varepsilon^2}{r^2},$$

where we used the fact that $\varepsilon \le \frac{1}{3}$. Finally, by the union bound, we find that the probability that any of the r knapsack constraints is violated is at most $\frac{\varepsilon^2}{r}$. This completes the proof. □

4 MSD via local search

4.1 Chapter overview

In this chapter, we focus mostly on distances of negative type, and extend some remarks to the metric case as well. The performance of local search is analyzed for several versions of the max-sum dispersion problem. We prove that these simple and efficient algorithms provide PTASs for negative-type MSD, when the constraint is a general matroid, or even an intersection of two matroids. And we obtain asymptotically optimal $O(1)$-approximations for the combinations of the dispersion function with monotone submodular objectives (MSD + f).

The main result of the chapter is a $(1 - O(\frac{1}{k}))$-approximation for negative-type MSD, constrained by a general matroid of rank k, in time linear in n. Thus, it provides a similar approximation as the algorithm presented in the previous chapter, with a considerably faster running time. The new algorithm is hence suitable for practical applications on very large data sets.

Organization and contribution

In Section 4.2, we give an informal introduction to the technique of local search, for the general scenario of a monotone function being maximized over a matroid constraint. We define a generic algorithm, and present general guidelines for its use. They include conditions that ensure an efficient execution time, and a framework to calculate approximation ratios. More interestingly, we formalize a technique that has been recently used to achieve best-possible approximation ratios for submodular maximization [58, 118]. This technique allows to merge two local-search algorithms with distinct objective functions, into a new algorithm that is tailored to maximize a combination of the objectives. We consider the study and formalization of this procedure to be of independent interest.

In Section 4.3, we present a literature review of the most relevant work related to this chapter. The review consists of the most recent approximability results for the maximization of a monotone submodular function, and for metric MSD, both constrained by a matroid. Furthermore, we provide an improved approximation ratio for the combination of these two objectives (metric MSD + f). The use of the theory and techniques presented in Section 4.2 is highlighted there.

In Section 4.4, we consider the negative-type max-sum dispersion problem, constrained by a matroid of rank k. We prove that the standard local-search algorithm offers an approximation ratio of $1-\frac{4}{k}$, in time $O(nk^2 \log k)$, and hence provides a PTAS. Moreover, for the combination with monotone submodular functions (MSD + f), we present an improved $O(1)$-factor approximation, which is considerably better than the current $\frac{1}{2}$-factor approximation, for negative-type distances.

In Section 4.5, a PTAS is given for the negative-type MSD problem constrained by two matroids. Such a broad result has no parallel on the metric case, and showcases the strength of the negative-type inequalities.

4.2 A local-search toolbox for matroid-constrained maximization

In this section, we present a generic local-search algorithm for the maximization of a monotone function over the independent sets of a matroid. We list some basic properties that are sufficient for the algorithm to provide an approximation ratio. Furthermore, we analyze the scenario where the sum of two functions is to be maximized: if local-search algorithms provide approximation ratios for the individual functions, then they can be merged into an algorithm for the combined function.

We present the basic notions of a local search algorithm, in an informal way. For a more detailed introduction, we direct the reader to [1, 123]. Before we focus on a matroid constraint, we consider a general scenario, where for a finite ground set X, a monotone objective function $f : 2^X \to \mathbb{R}_+$ is to be maximized, over a certain family $\mathscr{F} \subset 2^X$ of feasible solutions (this is the same framework described in Section 2.2). Let $O \in \mathscr{F}$ be the optimal solution.

For each solution $A \in \mathscr{F}$, we need to define a *neighborhood* $N(A) \subset \mathscr{F}$. A solution A is called a *local optimum* if none of its neighbors has a better objective value. We describe the basic idea of the algorithm. Starting at some initial solution, it iteratively performs exhaustive search over the neighborhood of the current solution. It then moves to the best neighbor, and starts a new iteration. The algorithm thus advances from neighbor to neighbor, in a greedy way, until it finds a local optimum. Hence, for the design of an efficient algorithm, one must ensure that neighbors are small, while at the same time making sure that local optima compare well to the global optimum O.

The value $\min\{f(A)/f(O) : A \text{ is a local optimum}\}$, is called the *locality gap*. It gives a theoretical limit to the approximation ratio that the algorithm may offer. The value of the locality gap is usually not known. On the other hand, a known lower bound to this value is called a *locality ratio*. Proving a locality ratio is equivalent to proving an approximation ratio of the same value, for any approximation algorithm that outputs a local optimum.

However, in order to ensure an efficient execution time, the local optimality condition is often relaxed, and a local-search algorithm is set to halt as soon as it finds a solution such that the

objective value improvement offered by any neighbor is bounded by a threshold (we call this solution a near local optimum). There will be a trade-off between the threshold used as halting condition, and a consequent loss in the approximation ratio, with respect to the locality ratio.

Sometimes it is convenient to define an auxiliary potential function F different from f, and run a local-search algorithm returning a (near) local optimum with respect to F. Such an algorithm is called *non-oblivious*. In this case, the locality gap is defined as $\min\{f(A)/f(O) : A \text{ is a local optimum w.r.t } F\}$, and it may be larger than the locality gap of the standard (*oblivious*) local search. Consequently, better locality and approximation ratios can be obtained with non-oblivious algorithms, as we shall see.

4.2.1 Generic algorithm for a matroid constraint

We suppose in this section that the feasible solutions correspond to the independent sets of a matroid $(X, \mathcal{I})$ of rank k. We will describe for this case a generic local-search algorithm. As stated in Section 2.2, we assume that the objective function f and the matroid are given by a value oracle and an independence oracle, respectively, and we are looking for an algorithm that runs in polynomial time. In particular, only polynomially many calls to these oracles can be made.

As f is monotone, we can restrict our attention to solutions that are bases (of cardinality k). Two bases A, B are considered neighbors if $|A \triangle B| \leq 2$.[1] For greater generality, we consider a non-oblivious local-search algorithm for this problem, that maximizes a potential function $F : 2^X \to \mathbb{R}_+$. This function F will be defined in terms of f, and possibly $\mathcal{I}$, and should provide a good locality ratio. In the case that F is equal to (a scalar multiple of) f, this is equivalent to an oblivious algorithm. And for a parameter $\varepsilon > 0$, we establish a halting condition in such a way that the difference between the locality ratio and the approximation ratio of the algorithm is no more than ε.

We present now a list of sufficient conditions for such an algorithm to work successfully. Let O_f and O_F be optimal bases for functions f and F, respectively. We call F *bounded*, if there is a computable number B_F of polynomial size, such that $\frac{F(O_F)}{f(O_f)} \leq B_F$. We also need an initial basis A_0 whose objective value is not too low. We call a basis A_0 *restricted for* F if $\log \frac{F(O_F)}{F(A_0)}$ is polynomially bounded.

An evaluation of F might need a superpolynomial number of evaluations of f to be computed. We call F *approximable* if, for any basis A, and numbers M and $\delta > 0$, an estimate $F'(A)$ of $F(A)$ can be computed in time polynomial in n, δ^{-1} and $\log M$, such that

$$\mathbb{P}\left[|F'(A) - F(A)| \geq \delta F(A)\right] \leq \frac{1}{M}.$$

[1] The notion of neighborhood will be different in Section 4.5, as it deals with a different type of constraint.

Algorithm 4.1: Local search for matroid-constrained maximization, associated to a potential function F and a parameter $\varepsilon > 0$.

Define $\delta = \frac{\varepsilon}{kB_F}$.
Compute a restricted basis A_0 and initialize $A \leftarrow A_0$.
while $\exists$ *pair* $(a,b) \in A \times (X \setminus A)$ *such that* $A - a + b \in \mathscr{I}$ *and* $F(A-a+b) > (1+\delta)F(A)$ **do**
  Find such a pair (a,b) maximizing $F(A-a+b)$.
  Set $A \leftarrow A - a + b$.
return A.

If this is the case, we can choose M large enough, and use the union bound, to prove that with high probability all the necessary evaluations of F' during the algorithm execution will have a multiplicative error within $1 \pm \delta$. Hence, the approximation loss due to estimation errors can be made of the same or smaller order as the loss due to the halting condition. To simplify analysis, whenever F is approximable, we assume that we have an exact value oracle for it.[2]

We assume that the previous conditions are satisfied by a potential function F, and define Algorithm 4.1, associated to F and to a parameter $\varepsilon > 0$, which performs single-element swaps in each iteration. In order to prove a good locality ratio, we will use the following well-known exchange property observed by the bases of a matroid. We shall refer to the bijection defined below as *Brualdi bijections*.

Lemma 4.1 (Brualdi [25])**.** *For any two independent sets $A, B \subset \mathscr{I}$ of equal cardinality, there is a bijection $\pi : A \to B$ such that for any $a \in A$, we have $A - a + \pi(a) \in \mathscr{I}$. In particular, such a bijection satisfies that it is the identity mapping over $A \cap B$.*

Theorem 4.2. *Consider the problem $\max_{A \in \mathscr{I}} f(A)$, for a matroid $(X, \mathscr{I})$ and a monotone function $f : 2^X \to \mathbb{R}_+$, and let O_f be the corresponding optimal basis. Let $F : 2^X \to \mathbb{R}_+$ be a potential function that is bounded, approximable, and for which a restricted basis can be computed efficiently. Moreover, assume that for any two bases A and B and a Brualdi bijection $\pi : A \to B$,*

$$f(A) \geq (1 - \alpha_B) f(B) + \sum_{a \in A} [F(A) - F(A - a + \pi(a))], \tag{4.1}$$

for a coefficient α_B in $[0,1]$ that may depend on B. Then, for the problem above, Algorithm 4.1 associated to F and to a parameter $\varepsilon > 0$ offers an approximation ratio of $(1 - \alpha_{O_f} - \varepsilon)$.

Proof. First we study the approximation ratio. By the way the updates are performed, A will be a basis throughout the execution of the algorithm. Let the output basis be S, and consider a Brualdi bijection $\pi : S \to O_f$. For each $a \in S$, the set $S - a + \pi(a)$ is also a basis, so by the halting condition we have that $(1+\delta)F(S) \geq F(S - a + \pi(a))$. It implies that

$$F(S) - F(S - a + \pi(a)) \geq -\delta F(S) \geq -\frac{\varepsilon}{kB_F} F(S) \geq -\frac{\varepsilon}{k} f(O_f).$$

[2] Technical details about dealing with an approximable potential function can be found in [58].

If we apply inequality (4.1) to this bijection, and use the previous bound for all $a \in A$, we get

$$f(S) \geq (1-\alpha_{O_f})f(O_f) - k\left(\frac{\varepsilon}{k}f(O_f)\right) = (1-\alpha_{O_f}-\varepsilon)f(O_f).$$

Next, we analyze the complexity of the algorithm. The basis A_0 is computed in polynomial time, and indeed in most applications this complexity is low compared to the rest of the algorithm. Each iteration is performed in time $O(nk)$, and in every iteration the value $F(A)$ grows at a multiplicative rate of at least $(1+\delta)$. If the total number of iterations is T, then

$$F(O_F) \geq F(S) \geq (1+\delta)^T F(A_0),$$

and consequently

$$T = O\left(\frac{1}{\delta}\log\frac{F(O_F)}{F(A_0)}\right) = O\left(\frac{kB_F}{\varepsilon}\log\frac{F(O_F)}{F(A_0)}\right).$$

The terms B_F and $\log\frac{F(O_F)}{F(A_0)}$ are polynomially bounded, so the proof is complete. If we consider the complexity N of approximating each evaluation of F, then the overall running time is

$$O\left(\varepsilon^{-1}nk^2B_F N\log\frac{F(O_F)}{F(A_0)}\right).$$

□

In the limit $\varepsilon \to 0$, the previous theorem states that the locality ratio of the algorithm is $(1-\alpha_{O_f})$, which is apparent from inequality (4.1). When maximizing a function under a matroid constraint, it is common in the literature to state the locality ratio with a similar inequality, as we will see in Section 4.3, where α is usually a constant, and F is often just a scalar multiple of f.

4.2.2 Combining objective functions

The following is a formalization of a technique that has been recently used in the maximization of monotone submodular functions [58, 118], to obtain optimal approximation ratios via local search. The result allows to combine the locality ratios of two distinct objective functions, expressed in the form of (4.1), into a locality ratio for a combined objective function.

Theorem 4.3. *Consider a matroid $(X,\mathcal{I})$, and for $i=1,2$, consider a monotone function $f_i : 2^X \to \mathbb{R}_+$, and a corresponding potential function $F_i : 2^X \to \mathbb{R}_+$ that satisfies all conditions of Theorem 4.2, and where in particular inequality (4.1) holds for a coefficient $\alpha_B^{(i)}$. Define the function $g = f_1 + f_2$, consider the problem $\max_{A\in\mathcal{I}} g(A)$, and let O_g be the corresponding optimal basis. Then, $G = F_1 + F_2$ is a potential function for g that satisfies all conditions of Theorem 4.2, and where inequality (4.1) holds for a coefficient*

$$\alpha_B = \frac{f_1(B)}{g(B)}\alpha_B^{(1)} + \frac{f_2(B)}{g(B)}\alpha_B^{(2)}.$$

Consequently, this problem admits an approximation ratio of $(1-\alpha_{O_g}-\varepsilon)$, for any $\varepsilon > 0$.

We remark that α_B corresponds to a convex combination of α_B^1 and α_B^2, so in particular $\alpha_B \leq \max\{\alpha_B^1, \alpha_B^2\}$. Notice also that the claim extends to combinations of more than two functions, and to weighted combinations.

Proof. Consider any bases A and B and any Brualdi bijection $\pi : A \to B$. For potential functions F_1 and F_2, inequality (4.1) reads

$$f_i(A) \geq (1-\alpha_B^i) f_i(B) + \sum_{a \in A} [F_i(A) - F_i(A-a+\pi(a))], \quad \text{for } i = 1,2.$$

If we sum up these two inequalities and use the definitions of g and G, we get

$$\begin{aligned} g(A) &\geq g(B) - \alpha_B^1 f_1(B) - \alpha_B^2 f_2(B) + \sum_{a \in A} [G(A) - G(A-a+\pi(a))] \\ &= \left(1 - \frac{f_1(B)}{g(B)}\alpha_B^1 - \frac{f_2(B)}{g(B)}\alpha_B^2\right) g(B) + \sum_{a \in A} [G(A) - G(A-a+\pi(a))]. \end{aligned}$$

Hence we obtain the desired inequality (4.1) for G. If F_1 and F_2 are approximable, then G is clearly also approximable. G is also bounded because

$$\frac{G(O_G)}{g(O_g)} = \frac{F_1(O_G)}{g(O_g)} + \frac{F_2(O_G)}{g(O_g)} \leq \frac{F_1(O_{F_1})}{g(O_{f_1})} + \frac{F_2(O_{F_2})}{g(O_{f_2})} \leq \frac{F_1(O_{F_1})}{f_1(O_{f_1})} + \frac{F_2(O_{F_2})}{f_2(O_{f_2})} \leq B_{F_1} + B_{F_2}.$$

And finally, if we can efficiently compute a restricted basis A_i for F_i, $i = 1,2$, and, say, $F_1(A_1) \geq F_2(A_2)$, then A_1 is a restricted basis for G because

$$\frac{G(O_G)}{G(A_1)} \leq \frac{F_1(O_G) + F_2(O_G)}{F_1(A_1)} \leq \frac{F_1(O_G)}{F_1(A_1)} + \frac{F_2(O_G)}{F_2(A_2)} \leq \frac{F_1(O_{F_1})}{F_1(A_1)} + \frac{F_2(O_{F_2})}{F_2(A_2)}.$$

□

4.3 Related work

We present a short review of the most relevant results in the literature. They correspond to recent developments in the approximability of monotone submodular maximization [58, 118], using non-oblivious local search, as well as the current best approximation algorithms for metric MSD [23]. We also provide an improved result for the problem of metric MSD + f.

We highlight the use of the theoretical toolbox introduced above. Throughout this section, $(X, \mathscr{I})$ is a fixed matroid, A and B are arbitrary matroid bases, $\pi : A \to B$ is a Brualdi bijection, and $\varepsilon > 0$ is an arbitrary constant. We will state local ratio results in the form of inequality (4.1), and invoke Theorems 4.2 and 4.3 to obtain corresponding approximation guarantees.

4.3.1 Monotone submodular maximization

Consider the problem $\max_{A \in \mathscr{I}} f(A)$, where $f : 2^X \to \mathbb{R}_+^X$ is a submodular monotone function. This problem is first studied in the seminal work of Fisher, Nemhauser and Wolsey [59]. They show that the oblivious local-search algorithm achieves a locality ratio of $\frac{1}{2}$, and an

approximation ratio of $\frac{1}{2}-\varepsilon$. They also provide an example where the locality gap is $\frac{1}{2}$, proving that the analysis is tight. Borodin et al. [23, Lemma 5] state this locality ratio in the form of inequality (4.1):[3]

$$f(A) \geq \left(1-\frac{1}{2}\right) f(B) + \frac{1}{2} \sum_{a \in A} \left[f(A) - f(A-a+\pi(a)) \right]. \tag{4.2}$$

Here, the potential function can be considered to be $F = \frac{1}{2} f$.

However, a different, more careful choice of the potential function can yield better results. Filmus and Ward [58, Theorem 5.1] prove that

$$f(A) \geq \left(1-\frac{1}{e}\right) f(B) + \sum_{a \in A} \left[F(A) - F(A-a+\pi(a)) \right], \tag{4.3}$$

for a potential F that is approximable, bounded with bound $B_F = O(\log k)$, and for which a restricted basis can be computed efficiently.[4] Therefore, by Theorem 4.2, the local search associated to F offers an approximation ratio of $1-\frac{1}{e}-\varepsilon$. This function was found with the technique of factor-revealing LPs, to ensure its optimality. And indeed, this approximation ratio is best possible, as Feige [52] proves that improving the bound of $1-\frac{1}{e}$ is NP-hard, even if f is an explicitly given coverage function. And Nemhauser and Wolsey [100] show that improving upon this bound requires an exponential number of queries in the value oracle model.

Conforti and Cornuéjols [41] define the *curvature* of a monotone submodular function f as

$$c = 1 - \min_{a \in A} \frac{f(X) - f(X-a)}{f(a) - f(\emptyset)}.$$

The property of submodularity implies that $c \geq 0$, while monotonicity implies $c \leq 1$. Hence, the coefficient c is always between 0 and 1, and it measures how close the function is to being linear, where $c = 0$ if and only if f is linear. Sviridenko, Vondrák and Ward [118] build upon Filmus and Ward's result, and optimize the choice of the potential F depending on the curvature of f. More concretely, for a monotone and submodular function f of curvature c, Sviridenko et al. consider the decomposition $f = l + f'$, where

$$l(A) = f(\emptyset) + \sum_{a \in A} [f(X) - f(X-a)], \quad \text{and} \quad f'(A) = f(A) - l(A), \quad \forall A \subset X.$$

They prove that l is linear, f' is submodular, monotone and normalized, and $f'(A) \leq c f(A)$ for each $A \subset X$. It is easy to see that for any linear function l,

$$l(A) = l(\emptyset) + \sum_{a \in A} \left[l(A) - l(A-a+\pi(a)) \right]. \tag{4.4}$$

[3] This result is originally stated in less generality, but its original proof carries on directly for this statement.
[4] The original potential function in [58] is a scalar factor $(1-1/e)$ away from this function.

Hence, if we define a potential function F' for f' as in inequality (4.3), and use it to define $F = l + F'$, it follows from Theorem 4.3 over inequalities (4.3) and (4.4), that F is a potential function of f which satisfies all conditions of Theorem 4.2, and such that

$$\begin{aligned} f(A) &\geq \left(1 - \frac{1}{e}\frac{f'(B)}{f(B)}\right) f(B) + \sum_{a \in A} \left[F(A) - F(A - a + \pi(a))\right] \\ &\geq \left(1 - \frac{c}{e}\right) f(B) + \sum_{a \in A} \left[F(A) - F(A - a + \pi(a))\right]. \end{aligned} \tag{4.5}$$

Thus, in [118] the authors conclude that the local search associated to F offers an approximation ratio of $1 - \frac{c}{e} - \varepsilon$. Moreover, they extend the negative result of [100] to prove that this bound is best possible; namely they show that, for each $c \in [0,1]$, improving upon the bound of $1 - \frac{c}{e}$ requires an exponential number of queries in the value oracle model.

As another applicability example, we perform the same decomposition of $f = l + f'$ as above, and this time define the potential function $F = l + \frac{1}{2} f'$. Then, using Theorem 4.3 over inequalities (4.2) and (4.4), we conclude the following.

Theorem 4.4. *For the matroid-constrained maximization of a submodular monotone function f of curvature c, there is a non-oblivious local-search algorithm that achieves an approximation ratio of $1 - \frac{c}{2} - \varepsilon$, for any $\varepsilon > 0$, with a potential function that can be computed exactly in polynomial time.*

4.3.2 **Metric** MSD

Consider the MSD problem over a metric distance (X, d), constrained by a matroid $(X, \mathscr{I})$ of rank k. We study the results by Borodin et al. [23], for which we present short proofs. This will serve as a starting point for our analysis of the negative-type case considered in Section 4.4.

The first thing we need is a bound on the distances between elements paired by the Brualdi bijection $\pi : A \to B$. This is given in the next lemma.

Lemma 4.5 ([23])**.** *If (X, d) is metric, then for any sets $A, B \subset X$ of equal size $k \geq 3$, and any bijection $\pi : A \to B$,*

$$\sum_{a \in A} d(a, \pi(a)) \leq d(A, B) - d(B).$$

Proof. For any a in A and two distinct elements b, b' in $B - \pi(a)$, the triangle inequality gives

$$d(a, b) + d(a, b') \geq d(b, b').$$

Keeping a fixed and summing these inequalities over all $\binom{k-1}{2}$ pairs in $B - \pi(a)$ yields

$$\begin{aligned} \frac{k-2}{2} d(a, B - \pi(a)) + \frac{k-2}{2} d(a, B - \pi(a)) \geq d(B - \pi(a)), \quad \text{or equivalently} \\ (k-2)\left[d(a, B) - d(a, \pi(a))\right] \geq d(B) - d(B, \pi(a)). \end{aligned}$$

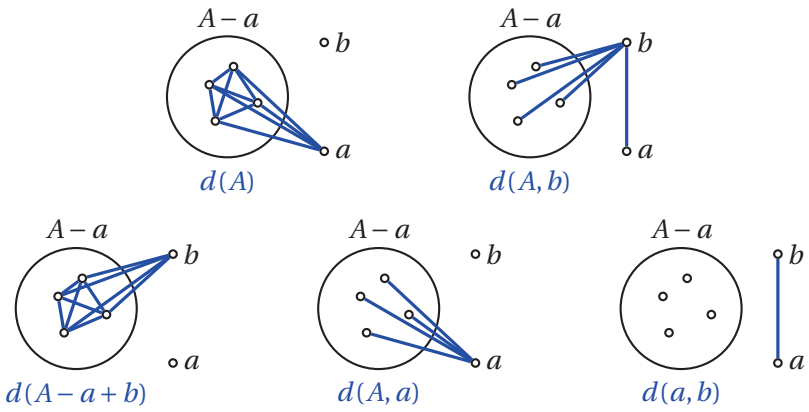


Figure 4.1: Visual proof of identity $d(A)+d(A,b) = d(A-a+b)+d(A,a)+d(a,b)$, assuming that $a \in A$ and $b \notin A-a$. Each edge is counted once on either side of the equation.[5]

Finally, summing over all elements a in A,

$$(k-2)\left[d(A,B) - \sum_{a\in A} d(a,\pi(a))\right] \geq kd(B) - d(B,B) = (k-2)d(B).$$

The result follows after dividing by $(k-2)$ and rearranging the terms. □

In the remainder of the section, we assume for simplicity that the rank is $k \geq 3$; instances with smaller rank can be solved efficiently by exhaustive search.

Lemma 4.6 ([23])**.** *If (X,d) is metric, then for any bases $A, B \in \mathscr{I}$ (of size $k \geq 3$) and any Brualdi bijection $\pi : A \to B$,*

$$d(A) \geq \left(1 - \frac{1}{2}\right) d(B) + \frac{1}{2} \sum_{a\in A} \left[d(A) - d(A-a+\pi(a))\right]. \tag{4.6}$$

Proof. A Brualdi bijection corresponds to the identity when restricted to $A \cap B$. Therefore, for any $a \in A$, we have $\pi(a) \notin A-a$, and the following identity holds (see Figure 4.1).

$$d(A) - d(A-a+\pi(a)) = d(A,a) - d(A,\pi(a)) + d(a,\pi(a)). \tag{4.7}$$

Summing up these terms over all $a \in A$, and using the previous lemma, we get

$$\begin{aligned}\sum_{a\in A} \left[d(A) - d(A-a+\pi(a))\right] &= d(A,A) - d(A,B) + \sum_{a\in A} d(a,\pi(a)) \\ &\leq 2d(A) - d(B).\end{aligned}$$

We obtain the claim after solving for the term $d(A)$ on the right-hand side. □

[5] Image excerpted from [31], with permission from the authors.

It is easy to check that all conditions of Theorem 4.2 are satisfied. Hence, Borodin et al. conclude that an oblivious local search offers an approximation ratio of $\frac{1}{2} - \varepsilon$ for metric MSD constrained by a matroid.

Next, the authors in [23] study metric MSD + f, i.e., for a given submodular and monotone function f, the goal is to maximize the function $g = d + f$ over a basis of the matroid. The sum of inequalities (4.2) and (4.6) immediately gives

$$g(A) \geq \left(1 - \frac{1}{2}\right) g(B) + \frac{1}{2} \sum_{a \in A} \left[g(A) - g(A - a + \pi(a)) \right],$$

and so, by Theorem 4.3, we have that an oblivous local search also offers an approximation ratio of $\frac{1}{2} - \varepsilon$ for metric MSD + f constrained by a matroid.

We highlight that the recent non-oblivious local-search procedures [58, 118] presented in Section 4.3.1 lead to a strengthening of the results of Borodin et al. Assume function f has curvature c. Define for it the potential F as in inequality (4.5) (see [118]), and set $G = \frac{1}{2}d + F$. Then, the sum of inequalities (4.5) and (4.6) yields

$$g(A) \geq \left(1 - \frac{1}{2}\frac{d(B)}{g(B)}\right) g(B) - \frac{c}{e}\frac{f(B)}{g(B)} + \sum_{a \in A} \left[G(A) - G(A - a + \pi(A)) \right].$$

From Theorem 4.3 we conclude the following.

Theorem 4.7. *Consider metric* MSD + f *constrained by a matroid, where f has curvature c. Let $\lambda_d \geq 0$ be such that $\frac{d(O)}{g(O)} \leq \lambda_d$, where O is the optimal basis. Then, Algorithm 4.1 associated to $\varepsilon > 0$ and the potential function G defined above, offers an approximation ratio of*

$$1 - \lambda_d \frac{1}{2} - (1 - \lambda_d)\frac{c}{e} - \varepsilon.$$

4.4 Negative-type MSD with a matroid constraint

We now pass to the MSD problem over a distance of negative type, constrained by a matroid of rank k. In this section, we prove that the oblivious version of the generic Algorithm 4.1 offers an approximation ratio of $1 - O(\frac{1}{k})$, and hence, provides a PTAS. We provide a detailed analysis of the running time of this algorithm. Then, we use the local-search machinery introduced in Section 4.2 to define a non-oblivious algorithm for the mixed-objective problem MSD + f, with a $O(1)$ approximation ratio that is asymptotically optimal.

4.4.1 Statement of locality ratio

We start our analysis in a similar way as for the metric case (Section 4.3.2). We consider a Brualdi bijection between two arbitrary bases, and look for a statement of the locality ratio in the form of inequality (4.1). Once again, the sum of distances between paired elements needs to be bounded.

Lemma 4.8. *If (X, d) is of negative type, then for any sets $A, B \subset X$ of cardinality k, and any*

bijection $\pi : A \to B$,

$$\sum_{a \in A} d(a, \pi(a)) \leq \frac{2}{k} d(A, B).$$

Proof. For any $a \in A$, if $\pi(a) \neq a$, inequality (2.8) gives

$$d(A, a) + d(A, \pi(a)) = d(A, \{a, \pi(a)\}) \geq \frac{2}{k} d(A) + \frac{k}{2} d(a, \pi(a)).$$

And the same inequality is true if $\pi(a) = a$; namely, in this case we have $d(A, a) \geq \frac{1}{k} d(A)$, also stemming from inequality (2.8). The sum of these inequalities over all $a \in A$ gives

$$d(A, A) + d(A, B) \geq 2d(A) + \frac{k}{2} \sum_{a \in A} d(a, \pi(a)).$$

The terms $d(A, A)$ and $2d(A)$ cancel out, and the claim follows. □

We prove now a locality ratio of $1 - \frac{4}{k+2}$.

Lemma 4.9. *For any two bases* $A, B \subset X$*, and any Brualdi bijection* $\pi : A \to B$*,*

$$d(A) \geq \left(1 - \frac{4}{k+2}\right) d(B) + \frac{k}{k+2} \sum_{a \in A} \left[d(A) - d(A - a + \pi(a)) \right]. \tag{4.8}$$

Proof. We assume that the matroid rank k is at least 2, for otherwise the statement follows trivially. Summing up identity (4.7) over all $a \in A$, and using the previous lemma, we get

$$\begin{aligned}
\sum_{a \in A} \left[d(A) - d(A - a + \pi(a)) \right] &= d(A, A) - d(A, B) + \sum_{a \in A} d(a, \pi(a)) \\
&\leq 2d(A) - \left(1 - \frac{2}{k}\right) d(A, B) \\
&\leq 2d(A) - \left(1 - \frac{2}{k}\right) \left[d(A) + d(B) \right] \\
&= \frac{k+2}{k} d(A) - \frac{k-2}{k} d(B),
\end{aligned}$$

where the last inequality comes from inequality (2.8). The claim now follows after solving for the term $d(A)$ on the right-hand side. □

Theorem 4.10. *For negative-type* MSD *with a rank k matroid restriction, the oblivious Algorithm 4.1 offers an approximation ratio of* $(1 - \frac{4}{k+2} - \varepsilon)$*, for any* $\varepsilon > 0$*. Consequently, it provides a polynomial-time approximation scheme to the problem.*

See Remark 2.1 for the implication towards a PTAS. The previous statement is a direct application of Theorem 4.2, for which all necessary conditions are satisfied. In particular, if we take as initial solution a basis A_0 that contains the pair $\{a, b\} \in I$ of maximum distance $d(a, b)$, then it can be checked that $d(A_0) \geq \frac{opt}{k}$, so A_0 is restricted.[6] However, we prove next that the algorithm is highly efficient, even if the initial solution is an arbitrary basis.

[6] Indeed, if $\{a, b\} \subset A_0$, inequality (2.9) gives $d(A_0) \geq \frac{k}{2} d(a, b)$; and we also know that $opt \leq \frac{k^2}{2} d(a, b)$.

4.4.2 Complexity of the algorithm

The following is a mostly standard argument showing exponentially fast convergence of the local search algorithm. We will take advantage of the greedy way in which the improvements are made.

Lemma 4.11. *If the oblivious Algorithm 4.1 starts with an arbitrary basis A_0, and returns a basis A_t after t iterations, then*

$$d(A_t) \geq \left[1 - \frac{4}{k+2} - \left(1 - \frac{k+2}{k^2}\right)^t\right] opt.$$

Proof. Denote by A_i, $i = 0, \cdots, t$, the basis obtained after i iterations of the algorithm. For a fixed i, let $\pi : A_i \to O$ be a Brualdi bijection between A_i and the optimal basis O. A reformulation of inequality (4.8) gives

$$\frac{1}{k} \sum_{a \in A_i} [d(A_i - a + \pi(a)) - d(A_i)] \geq \frac{k-2}{k^2} d(O) - \frac{k+2}{k^2} d(A_i).$$

The left-hand side corresponds to the average improvement in the objective value, obtained by swaps of the paired elements. As A_{i+1} is chosen greedily over all feasible swaps, we learn that

$$d(A_{i+1}) - d(A_i) \geq \frac{k-2}{k^2} d(O) - \frac{k+2}{k^2} d(A_i),$$

and after regrouping terms:

$$\left(1 - \frac{4}{k+2}\right) d(O) - d(A_{i+1}) \leq \left(1 - \frac{k+2}{k^2}\right) \left[\left(1 - \frac{4}{k+2}\right) d(O) - d(A_i)\right], \quad \forall i = 0, \cdots, t-1.$$

If we apply the previous inequality sequentially over all $0 \leq i \leq t-1$:

$$\left(1 - \frac{4}{k+2}\right) d(O) - d(A_t) \leq \left(1 - \frac{k+2}{k^2}\right)^t \left[\left(1 - \frac{4}{k+2}\right) d(O) - d(A_0)\right].$$

And hence,

$$\begin{aligned} d(A_t) &\geq \left(1 - \frac{4}{k+2}\right) \left[1 - \left(1 - \frac{k+2}{k^2}\right)^t\right] d(O) + \left(1 - \frac{k+2}{k^2}\right)^t d(A_0) \\ &\geq \left[1 - \frac{4}{k+2} - \left(1 - \frac{k+2}{k^2}\right)^t\right] d(O). \end{aligned}$$

□

Theorem 4.12. *For negative-type* MSD *with a rank k matroid constraint, it suffices to run the oblivious Algorithm 4.1 for $O(k \log k)$ iterations, starting with an arbitrary basis, to obtain an approximation ratio of $(1 - \frac{4}{k})$. Moreover, this algorithm can be implemented to run in time $O(nk^2 \log k)$, when counting distance evaluations and calls to the independence oracle as unit time.*

Proof. From the previous lemma, if Algorithm 4.1 runs for $t = \left\lceil \frac{k^2}{k+2} \ln \frac{k(k+2)}{8} \right\rceil = O(k \log k)$ iterations, then the approximation ratio is $1 - \frac{4}{k+2} - \frac{8}{k(k+2)} = 1 - \frac{4}{k}$.
To complete the proof, it remains to show that each iteration can be performed in time $O(nk)$. To achieve this, we use the following identity (see Figure 4.1).

$$d(A - a + b) = d(A) - d(A, a) + d(A, b) - d(a, b), \quad \forall a \in A,\ b \in X \setminus A.$$

Consider a fixed iteration, where the current set is A, and suppose that we have a table with the values of $d(A, c)$ for each $c \in X$. Then, for any $(a, b) \in A \times (X \setminus A)$, the evaluation $d(A - a + b)$ can be computed in constant time, using the table and the identity above, and the feasibility of $A - a + b$ is also checked in constant time. Hence, the optimal swap pair can be found in time $O(nk)$. And after the swap, the table can be updated in constant time per entry. □

We remark the high efficiency of the algorithm, which is linear in n. It is thus applicable on many real-life applications, where n is large and k is of medium size. For instance, if $k = 50$, the running time stays low, and the approximation guarantee is over 90%. In Section 5.5, we will discuss the application of core-sets for geometric instances of the problem. It will allow to reduce the complexity of the algorithm even more, and adapt it into streaming and distributed models of computation, to tackle instances with huge values of n.

4.4.3 Locality gap

We prove that the previous approximation ratio almost matches the locality gap of the local-search algorithm, even in the case of a uniform matroid, and even if several elements are swapped in each iteration.

Theorem 4.13. *For negative-type* MSD_k*, the locality gap of a local-search algorithm that swaps a sublinear number of elements per iteration is at most* $1 - \frac{1}{2k} + o(\frac{1}{k})$.

Proof. Consider a local-search algorithm that in each iteration removes at most c elements of the current set and adds at most c new elements, where $c = o(k)$. We define the following distance space (X, d), which will be of negative type by Lemma 2.10. Let $n = 2k$, and let X be partitioned into two k-sets $X = A \cup B$. The distances are: $1 + \frac{c}{2k^2}$ for all pairs within A, $1 + \frac{1}{2k}$ for all distances within B, and 1 for all pairs across A and B. Then, $d(A)/d(B) = 1 - \frac{1}{2k} + o(\frac{1}{k})$. The proof is complete once we show that A is a local optimum. If we swap c elements in A for c elements in B, the gain is $\binom{c}{2}(1 + \frac{1}{2k}) + c(k - c)$; and the loss is $\binom{c}{2}(1 + \frac{c}{2k^2}) + c(k - c)(1 + \frac{c}{2k^2})$. Thus, the total gain is

$$\begin{aligned}
\binom{c}{2}\left(\frac{1}{2k} - \frac{c}{2k^2}\right) - c(k-c)\frac{c}{2k^2} &\le \frac{c^2}{2}\left(\frac{1}{2k} - \frac{c}{2k^2}\right) - \frac{c^2}{2k^2}(k-c) \\
&= \frac{c^2}{4k^2}\left[(k-c) - 2(k-c)\right] \\
&= -\frac{c^2(k-c)}{4k^2} < 0.
\end{aligned}$$

□

4.4.4 Combination with a monotone submodular function

We consider now the negative-type MSD + f problem. That is, we are given a distance (X, d) of negative type, a matroid $(X, \mathscr{I})$ of rank k, and additionally a function $f : 2^X \to \mathbb{R}_+$ that is monotone and submodular, and we search for the basis that maximizes the mixed function $g = d + f$. We assume moreover that the curvature of f is c.

We define for f the potential function F as in inequality (4.5) (see [118]), and use it to define $G = \frac{k}{k+2} d + F$. Then, for any bases A, B, and any Brualdi bijection $\pi : A \to B$ between them, the sum of inequalities (4.5) and (4.8) yields

$$g(A) \geq \left(1 - \frac{4}{k+2}\frac{d(B)}{g(B)} - \frac{c}{e}\frac{f(B)}{g(B)}\right) g(B) + \sum_{a \in A} \left[G(A) - G(A - a + \pi(a))\right].$$

From Theorem 4.3 we conclude the following.

Theorem 4.14. *Consider negative-type* MSD + f *constrained by a matroid of rank k, where f has curvature c. Let $\lambda_d = \frac{d(O)}{g(O)}$ and $\lambda_f = \frac{f(O)}{g(O)}$, where O is an optimal basis. Then, Algorithm 4.1 associated to $\varepsilon > 0$ and the potential function G defined above, offers an approximation ratio of*

$$1 - \lambda_d \frac{4}{k+2} - \lambda_f \frac{c}{e} - \varepsilon \geq 1 - \max\left\{\frac{4}{k+2}, \frac{c}{e}\right\} - \varepsilon.$$

We make some observations about this last result. In the case $c = 0$, the algorithm offers a PTAS for the linear case (MSD + l). This greatly improves upon the known $\frac{1}{2}$-approximation [18, 23], for negative-type distances. On the other hand, if k is large enough, the result yields an approximation ratio of $1 - \frac{c}{e} - \varepsilon$, which is known to be optimal even for the special case of maximizing a monotone submodular function with curvature c over a matroid constraint [118].

4.5 Negative-type MSD with a matroid-intersection constraint

We extend our results in the previous section, and prove that a local-search algorithm also offers a PTAS in the case of a matroid-intersection constraint. The PTAS is mostly of theoretical interest, as its complexity will be large. For clarity of exposition, we do not attempt to provide a sharp bound on its complexity. We consider a distance space (X, d) and two matroids $(X, \mathscr{I}_1)$ and $(X, \mathscr{I}_2)$ over the same ground set X, and our goal is to maximize the dispersion of a common independent set $A \in \mathscr{I}_1 \cap \mathscr{I}_2$.

4.5.1 Algorithm definition

Our Algorithm 4.2 for this problem is similar to Algorithm 4.1, except that a larger number of elements is exchanged per iteration. For a parameter $p \geq 2$, at most p elements from the current set A are removed, and at most $p - 1$ new elements are added to it. This algorithm is also very similar to a procedure suggested by Lee, Sviridenko and Vondrák [92], designed for the maximization of a monotone submodular function subject to multiple matroid constraints.

Algorithm 4.2: Local search for a matroid-intersection constraint with exchange parameter p and error parameter ε.

Define $\delta = \frac{\varepsilon}{12k}$.
Compute a restricted set $A_0 \in \mathscr{I}_1 \cap \mathscr{I}_2$ and initialize $A \leftarrow A_0$.
while $\exists P \subset X$ *with*

1. $|P \cap A| \leq p$, $|P \setminus A| \leq p-1$,
2. $A \triangle P \in \mathscr{I}_1 \cap \mathscr{I}_2$, *and*
3. $d(A \triangle P) > (1+\delta)d(A)$,

do
  Find such a set P maximizing $d(A \triangle P)$.
  Set $A \leftarrow A \triangle P$.
Find a maximal set $S \in \mathscr{I}_1 \cap \mathscr{I}_2$ containing A.
return S.

In this section, $k = \max\{|A| : A \in \mathscr{I}_1 \cap \mathscr{I}_2\}$ will be the maximum cardinality of a common independent set. It is well known that the cardinality of any (inclusion-wise) maximal common independent set is at least $\frac{k}{2}$ (see, e.g. [123]). We will use this property in our analysis. A possibility to define the initial solution A_0 is to find the pair $\{a,b\} \in \mathscr{I}_1 \cap \mathscr{I}_2$ with maximum distance $d(a,b)$, and then compute any maximal set containing the pair. Then, inequality (2.9) gives $d(A_0) \geq \frac{|A_0|}{2} d(a,b) \geq \frac{k}{4} d(a,b)$; while for the optimal solution we know that $d(O) \leq \binom{k}{2} d(a,b) \leq \frac{k^2}{2} d(a,b)$. Thus, we have the bound $d(A_0) \geq \frac{1}{2k} d(O)$.

The algorithm runs in polynomial time. Each iteration is performed in time $n^{O(p)}$, where p is a constant. And in each iteration, the value of $d(A)$ grows at a multiplicative rate of at least $(1+\delta)$. Hence, by an argument as in Theorem 4.2, the total number of iterations is

$$T = O\left(\frac{1}{\delta} \log \frac{d(O)}{d(A_0)}\right) = O\left(\frac{k}{\varepsilon} \log k\right).$$

4.5.2 Exchange property for matroid intersection

In the single matroid case, the analysis of our local-search algorithm had Brualdi's exchange property at its core (Lemma 4.1). For the intersection of two matroids, we will heavily rely on a similar, recently developed exchange property, where for any two common independent sets $A, B \in \mathscr{I}_1 \cap \mathscr{I}_2$, subsets of A are paired up with subsets of B, in such a way that swapping any subset of A with its paired subset in B leads again to a common independent set. The following lemma was shown in [37], building up on previous work [92, 36]. We present a simplified version of the lemma, leaving out some properties that we do not need.

Lemma 4.15 ([37, Lemma 3.3])**.** *For an integer $p \geq 2$, and two common independent sets $A, B \in \mathscr{I}_1 \cap \mathscr{I}_2$ with $|A| = |B|$, there exists a family of nonempty sets $P_1, \ldots, P_m \subseteq A \Delta B$, with $|P_i \cap A| \leq p$ and $|P_i \cap B| \leq p-1$ for $1 \leq i \leq m$, and coefficients $\lambda_1, \cdots, \lambda_m > 0$, such that*

1. *$A \Delta P_i \in \mathscr{I}_1 \cap \mathscr{I}_2$, for $1 \leq i \leq m$, and*
2. *$\sum_{i=1}^m \lambda_i \mathbb{1}^{P_i} = \frac{p}{p-1} \mathbb{1}^{A \setminus B} + \mathbb{1}^{B \setminus A}$.*

We recall again that $\mathbb{1}^A \in \mathbb{R}^X$ represents the characteristic vector of a set $A \subset X$. In the previous statement, the condition $|A| = |B|$ is in fact not necessary, even though it is important in the original lemma in [37] to achieve further properties. For completeness, we state the lemma without this requirement, and provide a short proof for it.

Lemma 4.16. *For an integer $p \geq 2$, and two common independent sets $A, B \in \mathscr{I}_1 \cap \mathscr{I}_2$, there exists a family of nonempty sets $P_1, \ldots, P_m \subseteq A \Delta P_i$, with $|P_i \cap A| \leq p$ and $|P_i \cap B| \leq p-1$ for $1 \leq i \leq m$, and coefficients $\lambda_1, \cdots, \lambda_m > 0$, such that*

1. *$A \Delta P_i \in \mathscr{I}_1 \cap \mathscr{I}_2$, for $1 \leq i \leq m$, and*
2. *$\sum_{i=1}^m \lambda_i \mathbb{1}^{P_i} = \frac{p}{p-1} \mathbb{1}^{A \setminus B} + \mathbb{1}^{B \setminus A}$.*

Proof. In the case that $A, B \in \mathscr{I}_1 \cap \mathscr{I}_2$ have different sizes, we will "lift" them into larger sets A' and B' of the same size, that are common independent sets of two auxiliary matroids $\mathscr{I}'_1$ and $\mathscr{I}'_2$. Then, the application of Lemma 4.15 over these auxiliary matroids will imply the claim.
Recall that k is the maximum cardinality of a common independent set. Define $X' = X \cup Y$, where Y is an auxiliary k-set that is disjoint from X. And define the auxiliary matroids $(X', \mathscr{I}'_j)$, for $j = 1, 2$, where $\mathscr{I}'_j = \{S' \subset X' : S' \cap X \in \mathscr{I}_j\}$. These are indeed matroids, as they correspond to the direct sum of the matroid $(X, \mathscr{I}_j)$ with the free matroid $(Y, 2^Y)$ (see, e.g., [114, volume B]). Finally, for sets $A, B \in \mathscr{I}_1 \cap \mathscr{I}_2$, let $A', B' \in \mathscr{I}'_1 \cap \mathscr{I}'_2$ be any sets such that $|A'| = |B'| = k$, $A = A' \cap X$, and $B = B' \cap X$.
We can now apply Lemma 4.15 to sets A' and B', with respect to matroids $\mathscr{I}'_1$ and $\mathscr{I}'_2$, to obtain a family of sets $P'_1, \cdots, P'_m \subset X'$ and coefficients $\lambda_1 \cdots, \lambda_m$, satisfying the properties guaranteed by the lemma. To complete the proof, it is enough to observe that the family $P_i = P'_i \cap X$ for $1 \leq i \leq m$ also satisfies the claimed properties (after removing empty sets). These properties follow from the definitions of the auxiliary matroids, which in particular imply that for any $S' \in \mathscr{I}'_1 \cap \mathscr{I}'_2$, we have $S' \cap X \in \mathscr{I}_1 \cap \mathscr{I}_2$. □

4.5.3 Statement of locality ratio

As in the previous sections, when we use the exchange property in the analysis, we will need to bound the dispersion of the paired elements. We do that in the next lemma.

Lemma 4.17. *For $p \geq 2$ and $A, B \subset \mathscr{I}_1 \cap \mathscr{I}_2$, let $P_i \subset X$ and $\lambda_i > 0$ (for $1 \leq i \leq m$) be a family of sets and coefficients satisfying the conditions of Lemma 4.16. If (X, d) is of negative type, then*

$$\frac{|A|}{2p-1} \sum_{i=1}^m \lambda_i d(P_i) \leq \frac{2p}{p-1} d(A) + d(A, B).$$

Proof. For any $1 \le i \le m$, inequality (2.8) gives

$$d(\mathbb{1}^A, \mathbb{1}^{P_i}) = d(A, P_i) \ge \frac{|A|}{|P_i|} d(P_i) \ge \frac{|A|}{2p-1} d(P_i),$$

where we used $|P_i| \le 2p-1$. Multiplying by λ_i and summing over all i, we obtain

$$d\left(\mathbb{1}^A, \sum_{i=1}^m \lambda_i \mathbb{1}^{P_i}\right) \ge \frac{|A|}{2p-1} \sum_{i=1}^m \lambda_i d(P_i).$$

Now, by Lemma 4.16, the left-hand side of the inequality above is

$$\begin{aligned} d\left(\mathbb{1}^A, \sum_{i=1}^m \lambda_i \mathbb{1}^{P_i}\right) &= d\left(\mathbb{1}^A, \frac{p}{p-1}\mathbb{1}^{A\setminus B} + \mathbb{1}^{B\setminus A}\right) \\ &= \frac{p}{p-1} d(A, A\setminus B) + d(A, B\setminus A) \\ &\le \frac{p}{p-1} d(A, A) + d(A, B) \\ &= \frac{2p}{p-1} d(A) + d(A, B). \end{aligned}$$

The claim now follows. □

In the next lemma, we state a locality ratio for the local-search Algorithm 4.2. It has the same basic form as inequality 4.1, except that we express it in terms of the square root of the dispersion, $\sqrt{d(A)}$. This framework turns out to be more convenient in terms of clarity of the analysis. Also, it is evident in the following inequality that a PTAS is achieved when p and k/p are large enough.

Lemma 4.18. *Consider the same hypotheses as in Lemma 4.17, where in addition A is a maximal set. Define $\lambda = \frac{4p-2}{k}$, and assume that $k \gg p \gg 1$, so that $1 - 2\lambda - \frac{1}{p} > 0$. Then*

$$\sqrt{d(A)} \ge \left(1 - 2\lambda - \frac{1}{p}\right)\sqrt{d(B)} + \frac{1 - \lambda - \frac{1}{p}}{2\sqrt{d(A)}} \sum_{i=1}^m \lambda_i [d(A) - d(A \triangle P_i)].$$

Proof. For a fixed i, the following identity can be verified by a diagram similar to Fig. 4.1:

$$\begin{aligned} d(A) - d(A \triangle P_i) &= d(A, P_i \cap A) - d(A, P_i \cap B) + d(P_i \cap A, P_i \cap B) - d(P_i \cap A) - d(P_i \cap B) \\ &\le d(A, P_i \cap A) - d(A, P_i \cap B) + d(P_i). \end{aligned}$$

We multiply the previous inequality by λ_i, and sum over all indices i:

$$\sum_{i=1}^m \lambda_i [d(A) - d(A \triangle P_i)] \le d\left(\mathbb{1}^A, \sum_{i=1}^m \lambda_i \mathbb{1}^{P_i \cap A}\right) - d\left(\mathbb{1}^A, \sum_{i=1}^m \lambda_i \mathbb{1}^{P_i \cap B}\right) + \sum_i \lambda_i d(P_i).$$

For the sum on the left-hand side, we use the short-hand $\sum$, and we analyze the three terms on the right-hand side. From Lemma 4.16, we know that $\sum_i \lambda_i \mathbb{1}^{P_i \cap A} = \frac{p}{p-1}\mathbb{1}^{A\setminus B}$; hence the

first term is $\frac{p}{p-1}d(A, A \setminus B)$. Similarly, the second term is $d(A, B \setminus A)$. And for the third term, we use the previous lemma, and we also use the fact that A is maximal, so $2|A| \geq k$ and $\lambda = \frac{4p-2}{k} \geq \frac{2p-1}{|A|}$. Hence,

$$\begin{aligned}
\sum &\leq \frac{p}{p-1}d(A, A \setminus B) - d(A, B \setminus A) + \lambda\left(\frac{2p}{p-1}d(A) + d(A, B)\right) \\
&\leq \frac{p}{p-1}d(A, A) - d(A, B) + \lambda\left(\frac{2p}{p-1}d(A) + d(A, B)\right) \\
&= (1+\lambda)\frac{2p}{p-1}d(A) - (1-\lambda)d(A, B) \\
&\leq (1+\lambda)\frac{2p}{p-1}d(A) - (1-\lambda)\left[\frac{|B|}{|A|}d(A) + \frac{|A|}{|B|}d(B)\right],
\end{aligned}$$

where in the second line we added and removed multiples of the expression $d(A, A \cap B)$, with a positive net addition, and in the last line we used inequality (2.8). Now, let $q = \frac{|B|}{|A|}$, and consider the expression $qd(A) + \frac{1}{q}d(B)$. For fixed values of $d(A)$ and $d(B)$, the coefficient q that minimizes this expression is $q = \sqrt{\frac{d(B)}{d(A)}}$, which corresponds to the value $2\sqrt{d(A) \cdot d(B)}$. Thus,

$$\begin{aligned}
\sum &\leq (1+\lambda)\frac{2p}{p-1}d(A) - 2(1-\lambda)\sqrt{d(A) \cdot d(B)} \\
&= 2\sqrt{d(A)}\left[(1+\lambda)\frac{p}{p-1}\sqrt{d(A)} - (1-\lambda)\sqrt{d(B)}\right] \\
&\leq 2\sqrt{d(A)}\left[\frac{1}{1-\lambda-\frac{1}{p}}\sqrt{d(A)} - (1-\lambda)\sqrt{d(B)}\right].
\end{aligned}$$

The claim follows after solving for the term $\sqrt{d(A)}$ inside the brackets, and using the inequality $(1-\lambda)(1-\lambda-1/p) \geq 1-2\lambda-1/p$. □

Finally, we prove that the PTAS follows from the previous statement of locality ratio by setting k and p large enough.

Theorem 4.19. *Negative-type* MSD *over a matroid-intersection constraint admits a polynomial-time approximation scheme.*

Proof. Suppose that we want Algorithm 4.2 to achieve a $(1-\varepsilon)$-approximation ratio, for a constant $\varepsilon > 0$. We set the exchange parameter p large enough, so that $\frac{1}{p} \leq \frac{\varepsilon}{8}$, i.e., $p = \lceil \frac{8}{\varepsilon} \rceil = \theta(\varepsilon^{-1})$. We also assume k (the maximum size of a common independent set) to be large enough, so that $\lambda = \frac{4p-2}{k} \leq \frac{\varepsilon}{8}$. If this is not the case, then the cardinality of the optimal solution is bounded by a constant $O(\varepsilon^{-2})$, and we can find it efficiently via exhaustive search.
Let $S \in \mathscr{I}_1 \cap \mathscr{I}_2$ be the output solution of the algorithm. We apply Lemma 4.16 over S and the optimal solution O, to obtain a family of sets $P_i \subset X$, and coefficients $\lambda_i > 0$, for $1 \leq i \leq m$. S is a maximal set, hence we can apply Lemma 4.18 to obtain

$$\sqrt{d(S)} \geq \left(1 - 2\lambda - \frac{1}{p}\right)\sqrt{d(O)} + \frac{1 - \lambda - \frac{1}{p}}{2\sqrt{d(S)}} \sum_{i=1}^{m} \lambda_i [d(S) - d(S \triangle P_i)].$$

The coefficient that multiplies $\sqrt{d(O)}$ is bounded from below by $1 - 3\varepsilon/8$, because of our bounds on p and λ. Now we study the term inside the sum. By the halting condition in the algorithm, for each set P_i we know that $d(S \triangle P_i) - d(S) \leq \delta d(S)$. On the other hand, we can bound the sum of coefficients λ_i by

$$\sum_{i=1}^{m} \lambda_i \leq \left\| \sum_{i=1}^{m} \lambda_i \mathbb{1}^{P_i} \right\|_1 = \left\| \frac{p}{p-1} \mathbb{1}^{A \setminus B} + \mathbb{1}^{B \setminus A} \right\|_1 \leq 2|A \setminus B| + |B \setminus A| \leq 3k,$$

where we used the fact that $p \geq 2$. Therefore,

$$\begin{aligned}
\sqrt{d(S)} &\geq \left(1 - \frac{3\varepsilon}{8}\right)\sqrt{d(O)} - \frac{1}{2\sqrt{d(S)}} \left(\sum_i \lambda_i\right) \delta d(S) \\
&\geq \left(1 - \frac{3\varepsilon}{8}\right)\sqrt{d(O)} - \frac{3k\delta}{2}\sqrt{d(S)} \\
&= \left(1 - \frac{3\varepsilon}{8}\right)\sqrt{d(O)} - \frac{\varepsilon}{8}\sqrt{d(S)} \\
&\geq \left(1 - \frac{\varepsilon}{2}\right)\sqrt{d(O)}.
\end{aligned}$$

Finally, we conclude that $d(S) \geq \left(1 - \frac{\varepsilon}{2}\right)^2 d(O) \geq (1 - \varepsilon) d(O)$. This completes the proof. □

5 MSD via core-sets

5.1 Chapter overview

This chapter is dedicated to the problem of cardinality-constrained max-sum dispersion (MSD_k), over distance spaces that are Euclidean-squared, or that are induced by an arbitrary (but fixed) norm in $\mathbb{R}^q$, and where the embedding dimension q is assumed to be a low constant. We present a PTAS for all of these instances, by means of a single algorithm.

Using the convexity of the norm function, and the concept of subgradients, we prove the existence of a solution with a very simple structure, and whose dispersion is arbitrarily close to optimal. The algorithm starts by computing a core-set of the input that is guaranteed to contain this approximate solution. Then, thanks to the known structure of the latter, the algorithm is able to find it by exhaustive search in polynomial time.

The algorithm thus provides an approximation ratio of $(1-\varepsilon)$, in time $O(Mn\log k + Mk^M)$, where M is a constant that depends on ε, the dimension q, and the norm. We highlight the linear dependence on n. The implementation is very simple, and fits into streaming and distributed models of computation in a straight-forward way. Our core-set also compares favorably to other core-sets for MSD_k recently proposed in the research community.

Related work

When the dimension is part of the input, this general framework is as hard as metric MSD_k, because any metric distance can be embedded into the ℓ_∞ norm (Proposition 2.2). However, surprisingly little is known for the case of fixed dimension, despite the fact that geometric instances of low dimension constitute some of the most natural applications of the problem, e.g. in facility location.

As we mentioned in Section 2.4, Fekete and Meijer [56] present a PTAS for the problem over Manhattan distances of any constant dimension, and they remark that their result implies a $(\frac{1}{\sqrt{2}}-\varepsilon)$-approximation for the two-dimensional Euclidean case, for any $\varepsilon > 0$. For all other norms and dimensions, no approximation ratios better than $\frac{1}{2}$ were previously known. As noted in [108, 56], the NP-hardness status of the geometric MSD_k problem on fixed dimension remains open, for any norm (see Table 2.1).

Organization and contribution

In Section 5.2, we present a PTAS for Euclidean-squared distances in fixed dimension. Although a PTAS for this distance class is already implied by the $(1-o(1))$-approximations in the previous chapters, this result is a useful complement, for instances where n is large but k is not large enough with respect to the error parameter. Furthermore, the geometric properties of these distances allow for a particularly elegant proof, that is simpler than for the norm-induced cases. Hence, the study of this PTAS for Euclidean-squared distances is justified, if only as a didactic tool.

In Section 5.3, we extend the PTAS to Euclidean distances, by exploiting the convexity of the Euclidean norm function, and the concept of subgradients; and in Section 5.4, we further extend the analysis to an arbitrary norm. We reiterate that it is a single algorithm that works for all of these distance classes, up to a precision parameter that depends on the dimension, norm, and error tolerance.

Finally, in Section 5.5, we discuss the implementation of the algorithm in the streaming and distributed models. We also give a proper introduction to the notion of core-sets, and study the properties of the one used in our algorithm. Our core-set can work in conjunction with the local-search algorithm presented in Chapter 4, to achieve a high-quality and highly efficient algorithm for instances of MSD_k that are both of fixed dimension and of negative type.

5.2 The Euclidean-squared case

Consider a Euclidean-squared distance space that is represented with an embedding. That is, the input consists of finite set $X \subset \mathbb{R}^q$, where q is a low constant, and $d(x,y) = \|y-x\|_2^2$ for all points $x, y \in \mathbb{R}^q$. We present a $(1-\varepsilon)$-approximation algorithm for MSD_k that runs in time $O\left(Mn\log k + Mk^M\right)$, for a constant $M = O(\varepsilon^{-q/2})$.

5.2.1 Centroids and a geometric property of the optimal solution

The dispersion function has very particular geometric properties for the class of Euclidean-squared distances, related to the concept of *centroids*.[1] The centroid of a finite and non-empty set $A \subset \mathbb{R}^q$ is defined as

$$c_A = \frac{1}{|A|}\sum_{a\in A} a.$$

This is the point that minimizes the cross-dispersion $d(A, c_A)$; and moreover, when computing $d(A,x)$ as a function of x, the set A may be treated like a single point placed at c_A of mass $|A|$. We state these properties formally in the following lemma.

Lemma 5.1. *For any finite set $A \subset \mathbb{R}^q$ with centroid $c_A = \frac{1}{|A|}\sum_{a\in A} a$, we have*

$$d(A,x) = |A|d(c_A,x) + d(A,c_A) \quad \forall x \in \mathbb{R}^q, \quad \text{and} \tag{5.1}$$

$$d(A) = |A|d(A,c_A). \tag{5.2}$$

[1] We already mentioned some of these properties in Remark 2.14.

Proof. We consider the cross-dispersion $d(A,x)$, for any point $x \in \mathbb{R}^q$:

$$\begin{aligned} d(A,x) &= \sum_{a\in A} \|x-a\|_2^2 = \sum_{a\in A} \|(x-c_A)+(c_A-a)\|_2^2 \\ &= \sum_{a\in A} \left[\|x-c_A\|_2^2 + \|c_A-a\|_2^2 + 2(x-c_A)^T(c_A-a)\right] \\ &= \sum_{a\in A} d(c_A,x) + \sum_{a\in A} d(a,c_A) + 2(x-c_A)^T\left(|A|c_A - \sum_{a\in A} a\right) \\ &= |A|d(c_A,x) + d(A,c_A). \end{aligned}$$

The last term vanishes, because the expression in parenthesis is zero by definition of the centroid. This proves identity (5.1). Now, we use the identity $d(A) = \frac{1}{2}d(A,A)$, together with (5.1), to obtain

$$\begin{aligned} d(A) &= \frac{1}{2}d(A,A) = \frac{1}{2}\sum_{a\in A} d(A,a) \\ &= \frac{1}{2}\sum_{a\in A} [|A|d(c_A,a) + d(A,c_A)] \\ &= \frac{1}{2}[|A|d(A,c_A) + |A|d(A,c_A)] = |A|d(A,c_A). \end{aligned}$$

□

We would like to understand how the dispersion function behaves, when all but one of the points are fixed. To that effect, for a fixed non-empty set A, we study $d(A,x)$ as a function of x. From identity (5.1), we see that its value depends uniquely on the distance from x to c_A. Hence, the level sets of the function correspond to concentric spheres centered at c_A. This leads to the following observation.

Lemma 5.2. *The optimal k-set O is equal to $O = X \cap (\cup_{o\in O} H_o)$, where H_o is the half-space*

$$H_o = \{y \in \mathbb{R}^q : \ (y-o)^T(o-c_{O-o}) \geq 0\}.$$

Proof. For any elements $o \in O$ and $a \in X \setminus O$, the optimality of the set O implies that

$$0 \leq d(O) - d(O-o+a) = d(O-o,o) - d(O-o,a) = (k-1)[d(c_{O-o},o) - d(c_{O-o},a)],$$

where we used identity (5.1). Hence, $d(c_{O-o},a) \leq d(c_{O-o},o)$. This implies that each point in $X \setminus O$ must be inside the sphere centered at c_{O-o} and touching o; or equivalently, that each point of X that is strictly outside of this sphere must be in O.
The half-space H_o described in the statement is the unique half-space that intersects with this sphere exactly at point o. By the previous argument, all points of X contained in H_o must be part of O, so $o \in (X \cap H_o) \subset O$. Consequently, we have that $X \cap (\cup_{o\in O} H_o)$ both contains and is contained in O, hence it must be equal to O. This completes the proof. □

The previous lemma describes a quality of hollowness of the optimal set O. For instance, it says that no point in O is contained in the convex hull of $X \setminus O$. The main idea of our PTAS is to define a polynomially bounded collection of sets with a similar geometric structure, and then simply perform an exhaustive search over it. Continuing with the analysis of the set O, we remark that it is completely determined by sets of the form $X \cap H$, where H is a half-space, and that each of these sets $X \cap H$ is in turn determined by a direction $v \in \mathbb{R}^q$ in space and a cardinality m. This motivates the following definition.

Let $\mathbb{S}^{q-1}$ be the unit sphere in space $\mathbb{R}^q$. Given a unit vector $v \in \mathbb{S}^{q-1}$ and a positive integer m, we define the m-set $X(v,m) \subset X$ as follows: project X into the line spanned by v, and add the m highest points into $X(v,m)$. Equivalently, this set contains the m points $a \in X$ with highest value of $v^T a$.[2]

It is clear that for any half-space H, the set $X \cap H$ can be written as a set $X(v,m)$, where $0 \le m \le k$. However, there are still too many sets of this form to execute an exhaustive search. Fortunately, as we show below, reducing the search to only a small number of directions $v \in \mathbb{S}^{q-1}$ will be enough to a find a k-set O' whose dispersion is arbitrarily close to optimal.

5.2.2 θ-coverings and algorithm

For a fixed angle θ, we say that a set $V \subset \mathbb{S}^{q-1}$ of unit vectors is a *θ-covering* of $\mathbb{S}^{q-1}$, if for any $w \in \mathbb{S}^{q-1}$ there is a $v \in V$, such that the angle between w and v is at most θ.[3] The use of θ-coverings is common in the implementation of geometric algorithms, and it is known that for any $\theta > 0$, a θ-covering V of size $M = O(\theta^{-q})$ can be constructed efficiently (see, e.g., [122, Lemma 5.2]). Notice that M is a constant whenever θ is constant.

We will fix a θ-covering $V = \{v_1, \cdots, v_M\}$, and build sets of the form $X(v_j, m_j)$, with $v_j \in V$ and $0 \le m_j \le k$. Then, we will approximate the optimal k-set O by a union of such sets, i.e., by a set of the form $\cup_{j=1}^{M} X(v_j, m_j)$. We delay for a moment the proof of the following existence result, which immediately implies the PTAS.

Theorem 5.3. *If $V = \{v_1, \cdots, v_M\}$ is a θ-covering of $\mathbb{S}^{q-1}$, then there exists a k-set $O' \subset X$ which can be written as $O' = \cup_{j=1}^{M} X(v_j, m_j)$, for some list of coefficients $x_1, \cdots, x_M \in \{0, \cdots, k\}$, and such that*

$$d(O') \ge (1 - 3\theta^2) d(O).$$

Theorem 5.4. *For any $\varepsilon > 0$, and setting $\theta = \sqrt{\varepsilon/3}$, Algorithm 5.1 offers an approximation ratio of $(1-\varepsilon)$, in time $O(Mn \log k + Mk^M)$ and space $O(Mk)$, where $M = O(\varepsilon^{-q/2})$, assuming that distance evaluations and inner products are performed in unit time. Therefore, it is a polynomial-time approximation scheme for fixed-dimension Euclidean-squared* MSD_k.

[2] As we will work with only a small, fixed set of unit vectors v, we may perturb the input by an infinitesimal amount to avoid ties. In particular, we may assume that no two input points share a common position.

[3] θ-coverings may be equivalently defined in terms of distances, and are usually called ε-coverings or ε-nets.

Algorithm 5.1: Exhaustive search over sets of the form $\cup_{j=1}^{M} X(v_j, m_j)$, for a θ-covering V.

Let $V = \{v_1, \cdots, v_M\}$ be a θ-covering of $\mathbb{S}^{q-1}$.
for $j = 1, \cdots, M$ **do**
    Compute the set $X(v_j, k)$, with elements a ordered by the value of $v_j^T a$.
Initialize $A \leftarrow \emptyset$.
for *each list* $(m_1, \cdots, m_M) \in \{0, k\}^M$ *such that* $|\cup_{j=1}^{M} X(v_j, m_j)| = k$ **do**
    if $d(\cup_{j=1}^{M} X(v_j, m_j)) > d(A)$ **then**
        Set $A \leftarrow \cup_{j=1}^{M} X(v_j, m_j)$.
return A.

Proof. The algorithm is bound to find the set O' whose existence is guaranteed by Theorem 5.3, so

$$d(A) \geq d(O') \geq (1 - 3\theta^2) d(O) = (1 - \varepsilon) d(O).$$

In terms of complexity, a θ-covering V of size $M = O(\theta^{-q}) = O(\varepsilon^{-q/2})$ can be found efficiently. We split the algorithm in two phases, corresponding to the two **for** loops, that we call respectively the list-building and exhaustive-search phases.

The list-building phase consists of M individual processes, where each process computes an ordered list of the k best input points, with respect to a certain linear function.[4] Each process can be run in time $O(n \log k)$ and space $O(k)$, using a heap-sort algorithm, which moreover only passes through the set X once. Thus, this phase runs in time $O(Mn \log k)$ and space $O(Mk)$.

Assume now that these sorted lists are available. There are $O(k^{M-1})$ ways to build a k-set of the form $\cup_{j=1}^{M} X(v_j, m_j)$; and if these sets are explored in an order so that only one element changes from a set to the next, then their dispersions can be computed in linear time $O(Mk)$ per set, as was described in the proof of Theorem 4.12. Thus, the exhaustive-search phase runs in time $O(Mk^M)$ and space $O(Mk)$. This completes the proof. □

5.2.3 Proof of existence

In this section, we present the proof of Theorem 5.3. It will be constructive, by means of an algorithm that starts with O and outputs a set O' with the required properties. But first, we present some needed lemmas. The first one is a property of the optimal set O, which holds for a general distance space (X, d).

Lemma 5.5. *If O is the optimal k-set, then for any other k-set B, and any bijection $\pi : O \to B$ that corresponds to the identity mapping over $O \cap B$, we have*

$$d(O, B) \leq 2d(O) + \sum_{o \in O} d(o, \pi(o)).$$

[4] The union of all points in these M lists provides a core-set. Such core-set will be discussed in Section 5.5.

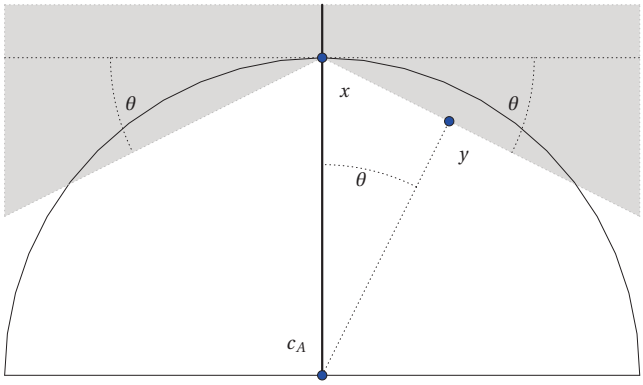


Figure 5.1: Inside the shaded region, point y minimizes the distance to point c_A.

Proof. Since O is optimal, for any $o \in O$ we must have (see Figure 4.1)

$$0 \leq d(O) - d(O - o + \pi(o)) = d(O, o) - d(O, \pi(o)) + d(o, \pi(o)).$$

And the claim follows when we sum over all $o \in O$:

$$0 \leq 2d(O) - d(O, B) + \sum_{o \in O} d(o, \pi(o)).$$

□

For a fixed set A, we consider again the expression $d(A, x)$ as a function of x. And for a point x, we define the half-space $H_x = \{y \in \mathbb{R}^q : (y - x)^T (x - c_A)\}$. As we saw in the proof of Lemma 5.2, if x moves anywhere within H_x, the value of $d(A, x)$ can only increase. On the other hand, we bound the decrease of $d(A, x)$ in the case that x moves marginally outside of H_x.

Lemma 5.6. *For any finite set $A \subset \mathbb{R}^q$, and any points $x, y \in \mathbb{R}^q$, if the angle between vectors $(x - c_A)$ and $(y - x)$ is at most $\frac{\pi}{2} + \theta$, for some $0 \leq \theta \leq \frac{\pi}{2}$, then*

$$d(A, x) - d(A, y) \leq \theta^2 d(A, x).^5$$

Proof. From identity (5.1), we have

$$d(A, x) - d(A, y) = |A| \left[d(c_A, x) - d(c_A, y) \right].$$

Thus, if the set A and point x are fixed, this difference is maximal whenever the distance from c_A to y is minimal. This occurs precisely when the points c_A, x and y form a right triangle, with angle θ at c_A (see Figure 5.1). In this case, by the definition of the cosine function,

$$\cos^2 \theta = \frac{\|y - c_A\|_2^2}{\|x - c_A\|_2^2} = \frac{d(c_A, y)}{d(c_A, x)}.$$

[5] As the zero vector is perpendicular to all other vectors, the statement holds (trivially) even if $x = c_A$ or $x = y$.

Algorithm 5.2: Approximation of a k-set $A \subset X$ by a set B of the form $\cup_{j=1}^{M} X(v_j, m_j)$, using the notion of centroids.

Let $V = \{v_1, \cdots, v_M\}$ be a θ-covering of $\mathbb{S}^{q-1}$.
Initialize $A^0 \leftarrow A$, $B^0 \leftarrow \emptyset$, $m_j^0 \leftarrow 0$ for $1 \le j \le M$.
for $i = 1, \cdots, k = |A|$ **do**
  Let a^i be any point in A^{i-1}.
  Let $v_{j'} \in V$ be such that the angle between $v_{j'}$ and $(a^i - c_{A^{i-1} \cup B^{i-1} - a^i})$ is at most θ (if $a^i = c_{A^{i-1} \cup B^{i-1} - a^i}$, select any $v_{j'} \in V$).
  Let b^i be the point in $X \setminus B^{i-1}$ maximizing $v_{j'}^T b^i$.
  if $b^i \in A^{i-1}$, **then**
    Set $a^i \leftarrow b^i$ (forget previous value of a^i).
  Set $A^i \leftarrow A^{i-1} - a^i$, $B^i \leftarrow B^{i-1} + b^i$.
  Set $m_{j'}^i$ to smallest value such that $X(v_j, m_{j'}^i)$ contains b_i; and $m_j^i \leftarrow m_j^{i-1}$ for $j \neq j'$.
return $B = B^k$.

And therefore,

$$d(A,x) - d(A,y) \le (1 - \cos^2\theta)|A| d(c_A, x) = \sin^2\theta |A| d(c_A, x) \le \theta^2 |A| d(c_A, x).$$

Finally, again by (5.1), we have that $|A| d(c_A, x) \le d(A, x)$. This completes the proof. □

We are ready to present the proof of Theorem 5.3, in the form of Algorithm 5.2 and Lemma 5.7.

Lemma 5.7. *If Algorithm 5.2 receives as input a k-set $A \subset X$ and a θ-covering $V = \{v_1, \cdots, v_M\}$, then at each iteration $0 \le i \le k$ we have the identity $B^i = \cup_{j=1}^{M} X(v_j, m_j^i)$. Moreover, if $A = O$ is the optimal k-set, and the output set is $B = B^k$, then*

$$d(O) - d(B) \le 3\theta^2 d(O).$$

Proof. Notice that each iteration removes an element from A^i, and adds an element to B^i, so that $|A^i| = k - i$, $|B^i| = i$, and moreover A^i and B^i remain disjoint, for each $0 \le i \le k$. Using the same labels that are ultimately given by the algorithm to the elements of A and B, we have that $A = \{a^1, \cdots, a^k\}$, $A^i = \{a^{i+1}, \cdots, a^k\}$, $B = \{b^1, \cdots, b^k\}$, and $B^i = \{b^1, \cdots, b^i\}$, where $a^i = b^i$ whenever $b^i \in A$.
Next, we prove that $B^i = \cup_{j=1}^{M} X(v_j, m_j^i)$ by induction on i, where the base case $i = 0$ holds trivially. Suppose it is true for $i-1$, and let vector $v_{j'}$ be chosen during the i-th iteration. Then

$$B^i = B^{i-1} + b^i \subset B^{i-1} \cup X(v_{j'}, m_{j'}^i) = \cup_{j=1}^{M} X(v_j, m_j^i).$$

It only remains to argue that $X(v_{j'}, m_{j'}^i) \subset B^i$. If this is not the case, there is an element $b \in X \setminus B^{i-1}$, different from b^i, such that $v_{j'}^T b > v_{j'}^T b^i$. But this contradicts the way element b^i is chosen.
We consider now the difference in dispersion between sets A and B. We look at the evolution of the k-set $A^i \cup B^i$ throughout the algorithm, and prove that its loss in dispersion in each

iteration is bounded. Fix an iteration i, and again let the vector $v_{j'}$ be chosen during this round. If $b^i \in A^{i-1}$, then $a^i = b^i$ and $A^{i-1} \cup B^{i-1} = A^i \cup B^i$, so there is no loss. Otherwise, the fact that $b^i \neq a^i$ implies that $v_{j'}^T b^i \geq v_{j'}^T a^i$, which means that the angle between $v_{j'}$ and $(b^i - a^i)$ is at most $\frac{\pi}{2}$. And by the choice of $v_{j'}$, the angle between $v_{j'}$ and $(a^i - c_{A^i \cup B^{i-1}})$ is at most θ. Therefore, the angle between $(b^i - a^i)$ and $(a^i - c_{A^i \cup B^{i-1}})$ is at most $\frac{\pi}{2} + \theta$, and we can apply Lemma 5.6 to obtain

$$\begin{aligned} d(A^{i-1} \cup B^{i-1}) - d(A^i \cup B^i) &= d(A^i \cup B^{i-1}, a^i) - d(A^i \cup B^{i-1}, b^i) \\ &\leq \theta^2 d(A^i \cup B^{i-1}, a^i) \\ &= \theta^2 \left[\sum_{j:i<j} d(a^i, a^j) + \sum_{j:i>j} d(a^i, b^j) \right]. \end{aligned}$$

Hence, the total loss in dispersion incurred in the algorithm is

$$\begin{aligned} d(A) - d(B) &= \sum_{i=1}^{k} \left[d(A^{i-1} \cup B^{i-1}) - d(A^i \cup B^i) \right] \\ &\leq \theta^2 \left[\sum_{i<j} d(a^i, a^j) + \sum_{i>j} d(a^i, b^j) \right] \\ &\leq \theta^2 \left[d(A) + \sum_{i \neq j} d(a^i, b^j) \right] \\ &= \theta^2 \left[d(A) + d(A, B) - \sum_{i=1}^{k} d(a^i, b^i) \right]. \end{aligned}$$

And finally, if A corresponds to the optimal solution O, we simply replace $d(O, B)$ by the bound stated in Lemma 5.5, to obtain $d(O) - d(B) \leq 3\theta^2 d(O)$. □

5.3 The Euclidean case

In this section, the MSD_k instance is given by a finite set $X \subset \mathbb{R}^q$, where $\mathbb{R}^q$ is equipped with the Euclidean distance: $d(x, y) = \|y - x\|_2$ for all $x, y \in \mathbb{R}^q$, and q is a low constant. We will prove that Algorithm 5.1 also provides a PTAS in this framework. In terms of complexity the only difference will be a worse dependency between θ and ε. We will exploit the convexity of the norm function, and the notion of subgradients.

5.3.1 Subgradients and the norm function

A well-known result in convex analysis (see e.g. [110]) is that any convex and continuous function $f(x)$ on $\mathbb{R}^q$ has a *subgradient* v_x at each point x. A subgradient at a point x is a vector v_x, in general not unique, with the property that

$$f(y) - f(x) \geq (y - x)^T v_x, \quad \text{for all } y \in \mathbb{R}^q. \tag{5.3}$$

For a fixed point a, consider the function $d_a(x) = d(a, x) = \|x - a\|_2$ over all $\mathbb{R}^q$. It corresponds to the ℓ_2 norm function, translated by a. It is convex and continuous, and differentiable every-

where except on a, and its gradient is the unit vector $\nabla d_a(x) = \frac{x-a}{\|x-a\|_2}$. If we abuse notation and extend the function $\nabla d_a(x)$ by defining $\nabla d_a(a) = 0$, then it provides a subgradient of $d_a(x)$ over all $x \in \mathbb{R}^q$.

Next, for a fixed finite set $A \subset \mathbb{R}^q$, we define the functions $d_A(x) = d(A,x)$ and $\nabla d_A(x) = \sum_{a\in A} \nabla d_a(x)$. Since $d_A(x)$ is the sum of convex functions, it is itself convex, and it is easy to check that $\nabla d_A(x)$ provides a corresponding subgradient, with $\|\nabla d_A(x)\|_2 \leq |A|$ for all $x \in \mathbb{R}^q$. Inequality (5.3) applied to these functions gives

$$d(A,y) - d(A,x) \geq (y-x)^T \nabla d_A(x), \quad \text{for all } x, y \in \mathbb{R}^q. \tag{5.4}$$

Let us consider the level sets of the function $d_A(x)$. They are not perfect spheres anymore, as in the previous section, and they are in general not smooth either, but they are the boundary of convex regions. At each point x, $\nabla d_A(x)$ is perpendicular to the corresponding level set at x, and it defines as well a half-space $H_x = \{y \in \mathbb{R}^q : (y-x)^T \nabla d_A(x) \geq 0\}$, with the property that each point y in it has a higher value $d_A(y)$ than $d_A(x)$. Inequality 5.4 gives a bound to this value gain. We state now a property of the optimal solution O, similar to Lemma 5.2.

Lemma 5.8. *The optimal k-set O is equal to $O = X \cap (\cup_{o\in O} H_o^*)$, for the sets*

$$H_o^* = \{o\} \cup \{y \in \mathbb{R}^q : (y-o)^T \nabla d_{O-o}(o) > 0\}.$$

Proof. For any elements $o \in O$ and $a \in X \setminus O$, the optimality of the set O implies

$$0 \geq d(O-o+a) - d(O) = d(O-o,a) - d(O-o,o) \geq (a-o)^T \nabla d_{O-o}(o),$$

where we used inequality (5.4). As the inequality above is violated for all points in $H_o^* - o$, then the points of X that lie in this set must be part of O. This completes the proof. □

We remark that Lemma 5.8 is slightly weaker than Lemma 5.2, in the sense that we use open half-spaces instead of closed ones. This is due to the fact that the function $d_A(x)$ might not be strictly convex anymore.[6] On the other hand, Lemma 5.8 will remain true for distances induced by any norm in $\mathbb{R}^q$.

As in the previous section, this hollow quality of the optimal solution motivates us to approximate it by a set of the form $\cup_{j=1}^M X(v_j, m_j)$, defined exactly as before, where $V = \{v_1, \cdots, v_M\}$ is a θ-covering of the unit sphere $\mathbb{S}^{q-1}$. Thus, once again we perform the exhaustive search described in Algorithm 5.1. All that we need now is a corresponding existence result, as in Theorem 5.3. We state it now, and delay its proof momentarily.

Theorem 5.9. *If $V = \{v_1, \cdots, v_M\}$ is a θ-covering of S^{q-1}, then there exists a k-set $O' \subset X$ which can be written as $O' = \cup_{j=1}^M X(v_j, m_j)$, for some list of coefficients $(x_1, \cdots, x_M) \in \{0, \cdots, k\}$, and such that*

$$d(O') \geq (1-4\theta) d(O).$$

[6] For the Euclidean norm, it can be verified that $d_A(x)$ is strictly convex as long as points in A are not collinear.

Theorem 5.10. *For any $\varepsilon > 0$, and setting $\theta = \varepsilon/4$, Algorithm 5.1 offers an approximation ratio of $(1-\varepsilon)$, and runs in time $O(Mn\log k + Mk^M)$ and space $O(Mk)$, where $M = O(\varepsilon^{-q})$, assuming that distance evaluations and inner products are performed in unit time. Therefore, it is a polynomial-time approximation scheme for fixed-dimension Euclidean* MSD_k.

Proof. The proof is virtually identical to that of Theorem 5.4. □

5.3.2 Proof of existence

Once again, the proof of Theorem 5.9 will be constructive, and based on an algorithm very similar to Algorithm 5.2. We present first some required lemmas. The first one is a property that holds for any metric distance (norm-induced distances are always metric).

Lemma 5.11. *Let (X,d) be a metric distance, and let $O \subset X$ be the k-set of largest dispersion. Then, for any k-set $B \subset X$, and any bijection $O \to B$ that corresponds to the identity mapping in $O \cap B$, we have*

$$(k-1)\sum_{o\in O} d(o,\pi(o)) \le 4d(O).$$

Proof. For any elements $o, o' \in O$, the triangle inequality gives

$$d(o,\pi(o)) \le d(o,o') + d(o',\pi(o)).$$

And summing up over all o and o' in O, we obtain

$$k\sum_{o\in O} d(o,\pi(o)) \le 2d(O) + d(O,B).$$

Finally, if we replace the term $d(O,B)$ by the bound given in Lemma 5.5, the claimed inequality follows. □

Lemma 5.12. *For any finite set $A \subset \mathbb{R}^q$, and any points $x, y \in \mathbb{R}^q$, if the angle between vectors $(y-x)$ and $\nabla d_A(x)$ is at most $\frac{\pi}{2}+\theta$, for some $0 \le \theta \le \frac{\pi}{2}$, then*

$$d(A,x) - d(A,y) \le \theta |A| d(x,y).^{7}$$

Proof. From inequality (5.4), we have

$$\begin{aligned} d(A,x) - d(A,y) &\le -(y-x)^T \nabla d_A(x) \\ &\le -\|y-x\|_2 \cdot \|\nabla d_A(x)\|_2 \cos\left(\frac{\pi}{2}+\theta\right) \\ &= d(x,y)\|\nabla d_A(x)\|_2 \sin\theta \\ &\le \theta |A| d(x,y) \end{aligned}$$

where we used the inequalities $\|\nabla d_A(x)\|_2 \le |A|$ and $\sin\theta \le \theta$. □

[7] Since the zero vector is perpendicular to all vectors, the statement holds (trivially) even if $x = y$ or $\nabla d_A(x) = 0$.

Algorithm 5.3: Approximation of a k-set $A \subset X$ by a set B of the form $\cup_{j=1}^{M} X(v_j, m_j)$, using the notion of subgradients.

Let $V = \{v_1, \cdots, v_M\}$ be a θ-covering of $\mathbb{S}^{q-1}$.
Initialize $A^0 \leftarrow A$, $B^0 \leftarrow \emptyset$, $m_j^0 \leftarrow 0$ for $1 \leq j \leq M$.
for $i = 1, \cdots, k = |A|$ **do**
  Let a^i be any point in A^{i-1}.
  Let $v_{j'} \in V$ be such that the angle between $v_{j'}$ and $\nabla d_{A^{i-1} \cup B^{i-1} - a^i}(a^i)$ is at most θ (if $\nabla d_{A^{i-1} \cup B^{i-1} - a^i}(a^i) = 0$, select any $v_{j'} \in V$).
  Let b^i be the point in $X \setminus B^{i-1}$ maximizing $v_{j'}^T b^i$.
  if $b^i \in A^{i-1}$, **then**
    Set $a^i \leftarrow b^i$ (forget previous value of a^i).
  Set $A^i \leftarrow A^{i-1} - a^i$, $B^i \leftarrow B^{i-1} + b^i$.
  Set $m_{j'}^i$ to smallest value such that $X(v_j, m_{j'}^i)$ contains b_i; and $m_j^i \leftarrow m_j^{i-1}$ for $j \neq j'$.
return $B = B^k$.

We present now the proof of Theorem 5.9, in the form of Algorithm 5.3 and Lemma 5.13. Notice that Algorithm 5.3 is virtually identical to Algorithm 5.2, except that we use the notion of subgradients, instead of that of centroids.

Lemma 5.13. *If Algorithm 5.3 receives as input a k-set $A \subset X$ and a θ-covering $V = \{v_1, \cdots, v_M\}$, then at each iteration $0 \leq i \leq k$ we have that $B^i = \cup_{j=1}^{M} X(v_j, m_j^i)$. Moreover, if $A = O$ is the optimal k-set, and the output set is $B = B^k$, then*

$$d(O) - d(B) \leq 4\theta d(O).$$

Proof. The proof that $B^i = \cup_{j=1}^{M} X(v_j, m_j^i)$ for each $0 \leq i \leq k$ is the same as in Lemma 5.7. We use the labels ultimately given by the algorithm to the elements of A and B, so that $A^i = \{a^{i+1}, \cdots, a^k\}$ and $B^i = \{b^1, \cdots, b^i\}$ for each i, where $a^i = b^i$ whenever $b^i \in A$.
Now, we look at the evolution of the k-set $A^i \cup B^i$ throughout the algorithm. Fix in iteration i and let $v_{j'}$ be the vector selected during this round. If $b^i \in A^{i-1}$, then $a^i = b^i$ and $A^{i-1} \cup B^{i-1} = A^i \cup B^i$. Otherwise, the fact that b^i was selected over a^i by the algorithm implies that the angle between $(b^i - a^i)$ and $v_{j'}$ is at most $\frac{\pi}{2}$. And we know that the angle between $v_{j'}$ and $\nabla d_{A^i \cup B^{i-1}}(a^i)$ is at most θ. Hence, the angle between $(b^i - a^i)$ and $\nabla d_{A^i \cup B^{i-1}}(a^i)$ is at most $\frac{\pi}{2} + \theta$, and we can apply Lemma 5.12 to obtain

$$d(A^{i-1} \cup B^{i-1}) - d(A^i \cup B^i) = d(A^i \cup B^{i-1}, a^i) - d(A^i \cup B^{i-1}, b^i) \leq \theta(k-1) d(a^i, b^i).$$

Hence, the total loss in dispersion incurred in the algorithm is

$$d(A) - d(B) = \sum_{i=1}^{k} \left[d(A^{i-1} \cup B^{i-1}) - d(A^i \cup B^i) \right] \leq \theta(k-1) \sum_{i=1}^{k} d(a^i, b^i).$$

And if $A = O$ is the optimal k-set, Lemma 5.11 implies that $d(O) - d(B) \leq 4\theta d(O)$. □

5.4 The case of a general norm

Finally, we consider the MSD_k problem, given by a finite set $X \subset \mathbb{R}^q$, where $\mathbb{R}^q$ is equipped with an arbitrary but fixed norm $\|\cdot\|_*$, and q is a low constant. The distances are thus given by $d(x,y) = \|y-x\|_*$, for all $x, y \in \mathbb{R}^q$. We prove that the PTAS for Euclidean norm is easily extended. Once again, in terms of complexity, the only difference will be a somewhat worse dependency between θ and ε.

5.4.1 One-dimensional case

We start with the case $q = 1$. The space $\mathbb{R}$ has a unique norm (up to scalar multiples), so we can think of $\|\cdot\|_*$ as being the ℓ_2 norm. Let the optimal k-set be $O = \{o_1, \cdots, o_k\}$, with points enumerated by increasing order on the line. For any $1 \le i \le k$, it can be verified that

$$\nabla d_{O-o_i}(o_i) = 2i - k - 1.$$

In this case, Lemma 5.8 implies that O must contain the $\lfloor \frac{k}{2} \rfloor$ left-most and $\lfloor \frac{k}{2} \rfloor$ right-most points in X. And if k is odd, the extra point can be chosen arbitrarily.

This trivial solution was pointed out by Tamir [120], in a comment to a paper by Ravi et al. [108], where a non-trivial, efficient algorithm was given for this framework. We notice also that this description of O is true whenever the points in X are collinear and contained in any $\mathbb{R}^q$, equipped with any norm.

5.4.2 General case and equivalence of norms

Now we consider the general case $q \ge 2$. A basic result in analysis is that all norms in $\mathbb{R}^q$ are *equivalent*. This means that there exist constants $C \ge c > 0$ such that

$$c\|x\|_2 \le \|x\|_* \le C\|x\|_2 \quad \text{for all } x \in \mathbb{R}^q. \tag{5.5}$$

In particular, if $\|\cdot\|_*$ is the ℓ_p norm, for $1 \le p \le \infty$, the corresponding coefficients are such that $\frac{C}{c} \le \sqrt{q}$.

We study the norm function $\|x\|_*$. It must be convex and continuous, but not necessarily differentiable. In any case, at each point x it must have a subgradient v_x, in general not unique. We will need to bound the ℓ_2 norm of such a subgradient.

Lemma 5.14. *If v_x is a subgradient of the norm function $\|x\|_*$ at point x, then*

$$\|v_x\|_2 \le C.$$

Proof. We define $y = x + v_x$. The subadditivity of the norm function gives

$$\|y\|_* = \|x + v_x\|_* \le \|x\|_* + \|v_x\|_*.$$

By the definition of subgradient (inequality 5.3),

$$\|v_x\|_* \geq \|y\|_* - \|x\|_* \geq (y-x)^T v_x = v_x^T v_x = \|v_x\|_2^2.$$

And therefore, applying inequality 5.5, we obtain

$$\|v_2\|_2 \leq \frac{\|v_2\|_*}{\|v_2\|_2} \leq C.$$

□

We fix such a subgradient of the norm at each point in $\mathbb{R}^q$, to define a function that we denote by $\nabla d_0(x)$ (by abuse of notation). Now, if we fix a point a and define the functions $\nabla d_a(x) = d(a,x)$ and $\nabla d_a(x) = \nabla d_0(x-a)$, it is clear that $\nabla d_a(x)$ provides a subgradient for $d_a(x)$ point by point. And if we now fix a finite set A and define the functions $d_A(x) = d(A,x)$ and $\nabla d_A(x) = \sum_{a\in A} \nabla d_a(x)$, again $\nabla d_A(x)$ provides a subgradient for $d_A(x)$ at each point $x \in \mathbb{R}^q$. And by the previous result, we have the bound $\|\nabla d_A(x)\|_2 \leq C|A|$.

As in the Euclidean case, these subgradients define a half-space over which the function $d_A(x)$ can only increase. In the next lemma, which is the counterpart to Lemma 5.12, we bound the loss in the case that x moves to a new point that is marginally outside of this half-space.

Lemma 5.15. *For any finite set $A \subset \mathbb{R}^q$, and any points $x, y \in \mathbb{R}^q$, if the angle between vectors $(y-x)$ and $\nabla d_A(x)$ is at most $\frac{\pi}{2}+\theta$, for some $0 \leq \theta \leq \frac{\pi}{2}$, then*

$$d(A,x) - d(A,y) \leq \frac{C}{c}\theta|A|d(x,y).$$

Proof. By the definition of subgradient (inequality (5.4)), we have

$$\begin{aligned} d(A,x) - d(A,y) &\leq -(y-x)^T \nabla d_A(x) \\ &\leq \|y-x\|_2 \cdot \|\nabla d_A(x)\|_2 \sin\theta \\ &\leq \left(\frac{1}{c}\|y-x\|_*\right)(C|A|)\theta \\ &= \frac{C}{c}\theta|A|d(x,y), \end{aligned}$$

where we used the previous bound on $\|\nabla d_A(x)\|_2$, and inequality (5.5). □

We are now ready to present an existence result, which uses Algorithm 5.3, and which in turn immediately shows the correctness of Algorithm 5.1.

Theorem 5.16. *If $V = \{v_1, \cdots, v_M\}$ is a θ-covering of $\mathbb{S}^{q-1}$, then there exists a k-set $O' \subset X$ which can be written as $O' = \cup_{j=1}^M X(v_j, m_j)$, for some list of coefficients $(x_1, \cdots, x_M) \in \{0, \cdots, k\}^M$, and such that*

$$d(O') \geq \left(1 - 4\frac{C}{c}\theta\right) d(O).$$

In particular, such a set O' is provided by Algorithm 5.3, for input set O.

Proof. The proof is exactly the same as in Lemma 5.13, except that it invokes Lemma 5.15 in place of Lemma 5.12. □

Theorem 5.17. *Consider the* MSD_k *problem over the space* $\mathbb{R}^q$ *equipped with a norm* $\|\cdot\|_*$ *such that* $c\|x\|_2 \leq \|x\|_* \leq C\|x\|_2$. *For any constant* $\varepsilon > 0$, *and setting* $\theta = \frac{c}{4C}\varepsilon$, *Algorithm 5.1 offers an approximation ratio of* $(1-\varepsilon)$, *and runs in time* $O(Mn\log k + Mk^M)$ *and space* $O(Mk)$, *where* $M = O\left(\left(\frac{C}{\varepsilon c}\right)^q\right)$, *assuming that distance evaluations and inner products are performed in unit time. Therefore, it is a polynomial-time approximation scheme for fixed-dimension norm-induced* MSD_k.

The proof of the previous theorem is virtually identical to that of Theorem 5.4. We recall that in the case of an ℓ_p norm, for $1 \leq p \leq \infty$, we have $\frac{C}{c} \leq \sqrt{q}$, so the size of the θ-covering will be $M = O\left(\left(\sqrt{q}/\varepsilon\right)^q\right)$.

5.5 Applications

In this section, we highlight some applications and adaptations of the results of this chapter. The existence results seen in the previous sections give rise to core-sets of size linear in k offering arbitrarily good approximations for these geometric instances of MSD_k. Furthermore, the simplicity of our algorithm makes it compatible with streaming and distributed models of computation for very large datasets.

Streaming and distributed models

Problems of diversity maximization, such as MSD, find many important applications in the analysis of massive amounts of data. As the size of the input increases, approximation algorithms need to address several specific challenges, while keeping a high-quality approximation ratio. For instance, a superlinear dependence on the size of the input becomes prohibitive. And the local resources of any available processor become limited, as well.

The *streaming* model [78] considers a processor with very limited memory space, which receives large data volumes as a stream. Thus, only a small portion of the input can be stored and accessed at any given moment. The stream may be read only one time (single pass), or several times. The quality of a streaming algorithm is measured by: a) the number of required passes of the stream, b) the space complexity required, and c) the time complexity required per item of the stream (update time).

The *distributed model* of computation considers a scenario where a massive amount of processors working in parallel is available, but each unit has limited resources. The input is partitioned and distributed among these units (or in many scenarios, data is originally produced and stored in this distributed manner); and each unit performs a task in parallel, based solely on its input. A single processor then merges their outputs, or another round of distributed computing may then take place. The quality of a distributed algorithm is measured by: a) the time and space complexity required per processor, and b) the overall time complexity required. A popular model for distributed computing is MapReduce [82, 103].

Core-sets

A *core-set* [4], with respect to a given objective to be maximized, is a subset X' of the input X, which contains a good approximation to the optimal solution of the entire input. Ideally, the size of the core-set should be much smaller, or even independent from the size of the input, and close to the size of the output. Recently, the use of core-sets for diversity problems in general, and the metric MSD_k in particular, has received attention in the literature [79, 6, 30]. Core-sets help adapt a readily available, sequential approximation algorithm, into the streaming and distributed models of computation: assuming that the computation of the core-set X' fits these models, and that the size of X' is small enough to be read and stored by a single processor, the sequential approximation algorithm can then be applied over it.

For MSD_k, a core-set $X' \subset X$ has an approximation ratio of $(1-\beta)$, if it is guaranteed to contain a k-set of value at least $(1-\beta)opt$. If an algorithm for MSD_k with an approximation ratio of $(1-\alpha)$ is then applied over X', the combination offers a ratio of $(1-\alpha)(1-\beta) \geq (1-\alpha-\beta)$.

A *composable core-set* [79] is a collection of core-sets for an arbitrary partition of the input set, such that the union of the core-sets is a core-set for the whole input set. A stronger concept is that of *core-preserving* core-sets [127], also called *mergeable* core-sets [3], with the additional property that taking a core-set of a union of core-sets yields a core-set with the same size and approximation factor; i.e., a sequential composition of core-set reductions affects neither the size nor the approximation ratio of the result. These are key properties that facilitate the construction of core-sets in streaming and distributed settings.

Composable core-sets with constant approximation ratios have been introduced for metric MSD_k [79, 6]. More recently, Cecarello et al. [30] study this problem over metric distances with bounded doubling dimension q' (see [71]). We remark that all norm-induced distances of dimension q have a doubling dimension $q' = \theta(q)$. For this framework, they provide a $(1-\varepsilon)$-approximation composable core-set, of size $O(\varepsilon^{-q'}k^2)$, which can be computed in a single-pass streaming process; and of size $O(\varepsilon^{-q'}k)$ if two passes are taken.

Our results

Our Algorithm 5.1 is composed of two separate processes, corresponding to the two **for** loops, that we call the list-building phase and the exhaustive-search phase. The existence results of the chapter (Theorems 5.3, 5.9 and 5.16) prove that, if $X' \subset X$ is the union of all points contained in the M lists built in the first phase, then X' is a $(1-\varepsilon)$-factor core-set of size $O(Mk)$. Here, M depends on the distance class, but in general $M = O(\varepsilon^{-q})$.

The list-building phase consists of M independent processes, each keeping an ordered list of the best k points from the input, with respect to a certain linear objective. Using heap-sort, each lists can be built and maintained in a single-pass streaming model, in space $O(k)$ and update time $O(\log k)$. Furthermore, if the input points are distributed over several machines, and the same θ-covering is used in each machine, the construction of these lists can be easily

parallelized and merged. Thus, this core-set is composable and core-preserving.

Theorem 5.18. *For* MSD_k *on norm-induced distances and Euclidean-squared distances of fixed dimension* q*, and for any* $\varepsilon > 0$*, there exists a core-set with an approximation ratio of* $(1-\varepsilon)$ *and size* $O(\varepsilon^{-q}k)$*. This core-set is composable and core-preserving, hence it can be computed in a distributed system. Furthermore, it can be computed and maintained in a single-pass streaming model, in space* $O(\varepsilon^{-q}k)$*, and update time* $O(\varepsilon^{-q}\log k)$*. This is, assuming that inner products are performed in unit time.*

This core-set offers similar a dependence on ε and q as the best current core-set, by Cecarello et al. [30]. However, it compares favorably to the latter, as it requires only a single-pass in the streaming model to reach a size linear in k; and its core-preserving property makes its construction more adaptable to diverse settings.

We also remark that the exhaustive-search phase of Algorithm 5.1 can be easily parallelized, by an arbitrary number of processors, each with a space requirement of only $O(\varepsilon^{-q}k)$. Therefore, the PTAS presented in this chapter fits in the distributed model of computation.

Finally, we consider an MSD_k instance that is both geometric of fixed dimension, and of negative type. For such an instance, we can first compute the core-set described above, and then perform the local-search algorithm presented in Chapter 4. The following statement is a direct consequence of Theorems 5.18 and 4.12.

Theorem 5.19. *Consider the* MSD_k *problem over the Euclidean-squared, Euclidean or Manhattan distances over* $\mathbb{R}^q$*, for fixed* q*. Then, for any* $\varepsilon > 0$*, the problem admits an approximation ratio of* $(1-\frac{4}{k}-\varepsilon)$*, in time* $O(\varepsilon^{-q}(n+k^3)\log k)$ *and space* $O(\varepsilon^{-q}k)$*, assuming that distance evaluations and inner products are performed in unit time.*

6 Conclusions and open questions

Max-sum dispersion (MSD) is one of the most prominent diversity-maximization problems. Prior to out results, the literature on approximation algorithms for this problem had focused almost exclusively on the metric case. As a consequence, the approximability of metric MSD was very well understood, even on special cases such as matroid constraints, and a combination with a submodular function in the objective. In contrast, surprising little was known about some the most natural and relevant geometric instances, such as Euclidean distances.

In this thesis we provide new approximation algorithms for these geometric instances of MSD. We obtain results that are much stronger than anything previously known for them, and for further special cases including knapsack and matroid-intersection constraints.

In particular, we propose the study of the negative-type condition on the distance space, as an alternative to the metric condition. This new distance class is general enough to contain some of the most important instances seen in real-life applications (including some non-metric ones), yet strong enough to provide a PTAS for this problem. This result is obtained by two different techniques: convex optimization is able to handle very general linear constraints but is relatively slow, and local search has a more restricted use but is highly efficient.

We also provide a PTAS for distances that are induced by an arbitrary norm, in fixed dimension. The algorithm can be executed in the streaming and distributed models of computation, to handle very large instances. A component of this algorithm is a core-set, with properties that fare well compared to other core-sets recently proposed for this problem. The combination of this core-set followed by local search results in an extremely fast and accurate algorithm for geometric instances of negative type and fixed dimension.

Open questions

Consider the MSD_k problem over distances induced by the q-dimensional ℓ_p norm. The standard local search achieves a $\frac{1}{2}$-approximation ratio for all values of p and q, and a PTAS whenever $1 \le p \le 2$ or $1 \le q \le 2$ (see Propositions 2.8 and 2.12). Both of these results are tight. What is the precise approximation ratio achieved by this algorithm, for specific values of p and q? Or rather, how far from negative-type is the q-dimensional ℓ_p norm?

The metric condition is a standard assumption in the analysis of problems dealing with distances, such as diversity and clustering problems. The special geometric cases of Euclidean and Manhattan distances are also widely studied, oftentimes under the assumption of fixed dimension. On the other hand, the class of negative-type distances, and in particular its definition in terms of negative-type inequalities (see Lemma 2.13), has received little attention in these areas of combinatorial optimization. We raise the question as to whether the study of this class can provide better algorithms, or simplify the analysis of current ones, for other diversity and clustering problems.

One can draw several interesting parallels between the class of submodular monotone functions f, and the dispersion function d over distances of negative type. For starters, for the maximization of these two functions, local search works well, and even allows for natural combinations of the functions. Next, the extension of d into $[0,1]^n$ is convex along the line parallel to $\mathbb{1}$, and concave on its orthogonal hyperplane; while the multilinear extension of f is concave along the line parallel to $\mathbb{1}$, and *almost* convex on its orthogonal hyperplane.[1] This difference can be appreciated when comparing our rounding algorithm for d (Theorem 3.4) with the continuous greedy algorithm for f [27]. In the former the loss comes only from the rounding procedure, and in the latter it comes only from the choice of the fractional solution.

Regarding the previous comment, and in the same spirit as Borodin et al. [22], we ask for the definition of a class of monotone set functions, as broad as possible and containing both of the aforementioned classes of functions, for which local search (Algorithm 4.1) offers high-quality approximations for the constrained maximization problem. On the other hand, and also in regards to our comment above, we conjecture that our rounding algorithm given by Theorem 3.4 can be combined with the continuous greedy algorithm, to yield good approximations for the mixed-objective problem MSD + f, under general linear constraints.

For negative-type MSD_k, we leave it as an open question whether the standard greedy algorithm matches the $(1-O(1/k))$-approximation ratio offered by local search (Theorem 4.12). This would further reduce the complexity of the problem, and could open the door to a greedy-based approximation algorithm in the streaming model. Such an algorithm is available, for instance, for monotone submodular maximization [15]. Similarly, for the cardinality-constrained, mixed-objective problem ($\mathrm{MSD}_k + f$), on both metric and negative-type distances, it would be of practical interest to develop a high-quality approximation in the streaming model, that works with space $O(k)$.

Another challenge is to extend our PTAS for MSD_k over norm-induced distances of fixed dimension (Algorithm 5.1), to general metric distances of bounded doubling dimension [71]. Our core-set (Thm. 5.18) might also be extended to these distances. And in terms of complexity, the question remains open as to whether MSD_k is NP-hard over distances induced by any norm in fixed dimension (see Table 2.1).

[1] The multilinear extension is convex along the line spanned by $v = \mathbb{1}^a - \mathbb{1}^b$, for any $a, b \in X$ [27].

Bibliography

[1] E. Aarts and J. K. Lenstra. *Local search in combinatorial optimization*. Princeton University Press, 1997.

[2] Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 32–40. ACM, 2013.

[3] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems (TODS)*, 38(4):26, 2013.

[4] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[5] A. A. Ageev and M. I. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(23):307–328, 2004.

[6] S. Aghamolaei, M. Farhadi, and H. Zarrabi-Zadeh. Diversity maximization via composable coresets. In *Proceedings of the 27th Canadian Conference on Computational Geometry*, 2015.

[7] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[8] N. Alon, S. Arora, R. Manokaran, D. Moshkovitz, and O. Weinstein. Inapproximability of densest κ-subgraph from average case hardness. *Unpublished manuscript*, 2011.

[9] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293. ACM, 1995.

[10] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.

[11] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[12] S. Ağca, B. Eksioglu, and J. B. Ghosh. Lagrangian solution of maximum dispersion problems. *Naval Research Logistics*, 47(2):97–114, 2000.

[13] D. Avis and M. Deza. The cut cone, L_1 embeddability, complexity, and multicommodity flows. *Networks*, 21(6):595–617, 1991.

[14] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop. The hardness of approximation of Euclidean k-means. In *31st International Symposium on Computational Geometry (SoCG)*, volume 34, pages 754–767. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

[15] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680. ACM, 2014.

[16] K Ball. Isometric embedding in ℓ_p-spaces. *European Journal of Combinatorics*, 11(4):305–311, 1990.

[17] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 201–210. ACM, 2010.

[18] S. Bhattacharya, S. Gollapudi, and K. Munagala. Consideration set generation in commerce search. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, pages 317–326. ACM, 2011.

[19] B. Birnbaum and K. J. Goldman. An improved analysis for a greedy remote-clique algorithm using factor-revealing LPs. *Algorithmica*, 55(1):42–59, 2009.

[20] L. M. Blumenthal. *Theory and Applications of Distance Geometry*, volume 347. Oxford, 1953.

[21] A. Bookstein. Information retrieval: A sequential learning process. *Journal of the American Society for Information Science*, 34(5):331–342, 1983.

[22] A. Borodin, D. T. M. Le, and Y. Ye. Weakly submodular functions. *arXiv preprint arXiv:1401.6697*, 2014.

[23] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st Symposium on Principles of Database Systems*, pages 155–166, 2012.

[24] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[25] R. A. Brualdi. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1(02):161–167, 1969.

[26] J. Byrka, T. Pensyl, B. Rybicki, A.ravind Srinivasan, and K. Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 737–756. SIAM, 2015.

[27] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[28] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[29] R. Carrasco, A. Pham, M. Gallego, F. Gortázar, R. Martí, and A. Duarte. Tabu search for the max–mean dispersion problem. *Knowledge-Based Systems*, 85:256–264, 2015.

[30] M. Ceccarello, A. Pietracaprina, G. Pucci, and E. Upfal. Mapreduce and streaming algorithms for diversity maximization in metric spaces of bounded doubling dimension. *arXiv preprint arXiv:1605.05590*, 2016.

[31] A. Cevallos, F. Eisenbrand, and R. Zenklusen. Local search for max-sum diversification. *arXiv preprint arXiv:1607.04557*, 2016.

[32] A. Cevallos, F. Eisenbrand, and R. Zenklusen. Max-sum diversity via convex programming. In *Proceedings of the 32nd Symposium on Computational Geometry (SoCG)*, pages 26:1–26:14, 2016.

[33] B. Chandra and M. M. Halldórsson. Approximation algorithms for dispersion problems. *Journal of algorithms*, 38(2):438–465, 2001.

[34] M. S. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM Journal on Computing*, 34(4):803–824, 2005.

[35] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding for matroid polytopes and applications. *arXiv preprint arXiv:0909.4348*, 2009.

[36] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 575–584, 2010.

[37] C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the 21st Annual ACM -SIAM Symposium on Discrete Algorithms (SODA)*, pages 1080–1097, 2011.

[38] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436. ACM, 2006.

[39] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[40] V. Cohen-Addad, P. N. Klein, and C. Mathieu. Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics. *arXiv preprint arXiv:1603.09535*, 2016.

[41] M. Conforti and G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.

[42] A. Das Sarma, S. Gollapudi, and S. Ieong. Bypass rates: reducing query abandonment using negative inferences. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 177–185. ACM, 2008.

[43] M. R. Q. de Andrade, P. M. F. de Andrade, S. L. Martins, and A. Plastino. GRASP with path-relinking for the maximum diversity problem. In *International Workshop on Experimental and Efficient Algorithms*, pages 558–569. Springer, 2005.

[44] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer-Verlag, Berlin, 1997.

[45] M. M. Deza and H. Maehara. Metric transforms and euclidean embeddings. *Transactions of the American Mathematical Society*, 317(2):661–671, 1990.

[46] L. E. Dor. Potentials and isometric embeddings in l_1. *Israel Journal of Mathematics*, 24(3):260–268, 1976.

[47] A. Duarte and R. Martí. Tabu search and GRASP for the maximum diversity problem. *European Journal of Operational Research*, 178(1):71–84, 2007.

[48] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[49] E. Erkut, T. Baptie, and B. Von Hohenbalken. The discrete p-maxian location problem. *Computers & Operations Research*, 17(1):51–61, 1990.

[50] E. Erkut and S. Neuman. Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40(3):275–291, 1989.

[51] E. Erkut and S. Neuman. Comparison of four models for dispersing facilities. *INFOR: Information Systems and Operational Research*, 29(2):68–86, 1991.

[52] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[53] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[54] U. Feige and M. Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41(2):174–211, 2001.

[55] U. Feige and M. Seltser. *On the densest k-subgraph problem*. Citeseer, 1997.

[56] S. P. Fekete and H. Meijer. Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, 38(3):501–511, 2004.

[57] B. Fichet. l_p spaces in data analysis. *Classification and related methods of data analysis, HH Bock ed., North Holland*, pages 439–444, 1988.

[58] Y. Filmus and J. Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014.

[59] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

[60] M. Fréchet. Les dimensions d'un ensemble abstrait. *Mathematische Annalen*, 68(2):145–168, 1910.

[61] Z. Friggstad, M. Rezapour, and M. R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *arXiv preprint arXiv:1603.08976*, 2016.

[62] M. Gallego, A. Duarte, M. Laguna, and R. Martí. Hybrid heuristics for the maximum diversity problem. *Computational Optimization and Applications*, 44(3):411–426, 2009.

[63] J. B. Ghosh. Computational aspects of the maximum diversity problem. *Operations research letters*, 19(4):175–181, 1996.

[64] F. R. Giles. *Submodular Functions, Graphs and Integer Polyhedra*. PhD thesis, University of Waterloo, 1975.

[65] F. Glover, K. Ching-Chung, and K. S. Dhir. A discrete optimization model for preserving biological diversity. *Applied mathematical modelling*, 19(11):696–701, 1995.

[66] F Glover, CC Kuo, and KS Dhir. Integer programming and heuristic approaches to the minimum diversity problem. *Journal of Business and Management*, 4(1):93–111, 1996.

[67] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pages 381–390. ACM, 2009.

[68] J. C. Gower and P. Legendre. Metric and euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3(1):5–48, 1986.

[69] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[70] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999.

[71] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 534–543. IEEE, 2003.

[72] A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *arXiv preprint arXiv:0809.2554*, 2008.

[73] V. Guruswami and P. Indyk. Embeddings and non-approximability of geometric problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 3, pages 537–538, 2003.

[74] P. Hansen and I. D. Moon. Dispersing facilities on a network. *Cahiers du GERAD*, 1995.

[75] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300. ACM, 2004.

[76] S.l Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 126–134. ACM, 2005.

[77] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Operations Research Letters*, 21(3):133–137, 1997.

[78] M. R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical report, 1998.

[79] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM Symposium on Principles of Database Systems*, pages 100–108, 2014.

[80] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 731–740. ACM, 2002.

[81] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the eighteenth annual Symposium on Computational Geometry (SoCG)*, pages 10–18. ACM, 2002.

[82] H. Karloff, S. Suri, and S. Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 938–948. Society for Industrial and Applied Mathematics, 2010.

[83] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1097, 1979.

[84] S. Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

[85] R. K. Kincaid. Good solutions to discrete noxious location problems via metaheuristics. *Annals of Operations Research*, 40(1):265–281, 1992.

[86] G. Kochenberger and F. Glover. Diversity data mining. 1999.

[87] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-median problem. *SIAM Journal on Computing*, 37(3):757–782, 2007.

[88] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 692–701. IEEE, 1993.

[89] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of algorithms*, 37(1):146–188, 2000.

[90] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.

[91] M. J. Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987.

[92] J. Lee, M. Sviridenko, and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

[93] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *SIAM Journal on Computing*, 45(2):530–547, 2016.

[94] Q. Lv, M. Charikar, and K. Li. Image similarity search with compact data structures. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 208–217. ACM, 2004.

[95] K. Makarychev, W. Schudy, and M. Sviridenko. Concentration inequalities for nonlinear matroid intersection. *Random Structures & Algorithms*, 46(3):541–571, 2015.

[96] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[97] J. Matoušek. Lecture notes on metric embeddings. kam.mff.cuni.cz/~matousek/ba-a4.pdf, 2013.

[98] M. W. Meckes. Positive definite metric spaces. *Positivity*, 17(3):733–757, 2013.

[99] I. D. Moon and S. S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30(3):290–307, 1984.

[100] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[101] P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.

[102] E. Pękalska and R. P. W. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.

[103] A. Pietracaprina, G. Pucci, M. Riondato, F. Silvestri, and E. Upfal. Space-round tradeoffs for mapreduce computations. In *Proceedings of the 26th ACM international conference on Supercomputing*, pages 235–244. ACM, 2012.

[104] D. Pisinger. Upper bounds and exact algorithms for p-dispersion problems. *Computers & operations research*, 33(5):1380–1398, 2006.

[105] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692. ACM, 2006.

[106] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM, 2008.

[107] P. Raghavan and C. D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[108] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[109] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[110] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.

[111] G. Salton and M. J. MacGill. Introduction to modern information retrieval. *McGraw-Hill computer science series*, 1983.

[112] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, pages 811–841, 1938.

[113] I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938.

[114] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[115] G. C. Silva, L. S. Ochi, and S. L. Martins. Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In *International Workshop on Experimental and Efficient Algorithms*, pages 498–512. Springer, 2004.

[116] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *Journal of the ACM*, 48(2):206–242, 2001.

[117] A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 181–191. Springer, 1998.

[118] M. Sviridenko, J. Vondrák, and J. Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1134–1148, 2015.

[119] A. Tamir. Obnoxious facility location on graphs. *SIAM Journal on Discrete Mathematics*, 4(4):550–567, 1991.

[120] A. Tamir. Comments on the paper: 'Heuristic and special case algorithms for dispersion problems' by S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. *Operations Research*, 46(1):157–158, 1998.

[121] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *2008 IEEE 24th International Conference on Data Engineering*, pages 228–236. IEEE, 2008.

[122] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[123] J. Ward. *Oblivious and non-oblivious local search for combinatorial optimization*. PhD thesis, University of Toronto, 2012.

[124] R. R. Weitz and S. Lakshminarayanan. An empirical comparison of heuristic methods for creating maximally diverse groups. *Journal of the operational Research Society*, 49(6):635–646, 1998.

[125] D. J. White. The maximal dispersion problem and the "first point outside the neighbourhood" heuristic. *Computers & operations research*, 18(1):43–50, 1991.

[126] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *2009 IEEE 25th International Conference on Data Engineering*, pages 1299–1302. IEEE, 2009.

[127] Hamid Zarrabi-Zadeh. Core-preserving algorithms. In *Proc. 20th Canad. Conf. Computat. Geom. (CCCG)*. Citeseer, 2008.

[128] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. ACM, 2003.

[129] C. X. Zhai and J. Lafferty. A risk minimization framework for information retrieval. *Information Processing & Management*, 42(1):31–55, 2006.

[130] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

Alfonso Cevallos

Chemin de Rionza 5, 1020 Renens, Switzerland.
alfonsoc@gmail.com – (+41) 787596172

Research Interests

I am interested in developing fast approximation algorithms for combinatorial problems, and their applications to operations research, computational geometry and information retrieval. I study the strengths and limitations of algorithms, and the inherent hardness of a problem (complexity theory).

My current areas of focus are diversification and clustering problems, distance embeddings and the analysis of the geometric structure of datasets, submodular optimization and the combinatorial properties of matroids, and the use of core-sets in streaming algorithms.

I am generally captivated by problems of geometric and graph theoretical natures, such as vehicle routing, facility location and network design problems, as well as clique, coloring, and matching problems.

Education

École Polythecnique Fédérale de Lausanne, Switzerland

2012 – present

PhD candidate in Mathematics, to defend in November 2016.
Discrete optimization chair. Advisor: Friedrich Eisenbrand.
Thesis: *Approximation algorithms for geometric dispersion*.

Université Bordeaux I, France & **Universiteit Leiden**, The Netherlands

2009 – 2011

M.Sc. in Mathematics.
Erasmus Mundus ALGANT program with mobility scheme.
Thesis in cryptography (robust secret sharing).

Universidad San Francisco de Quito, Ecuador

2004 – 2009

B.Sc. in Mathematics, Magna Cum Laude.
Minors in Physics and Computer Engineering.

University of Illinois at Urbana-Champaign, USA

2007 – 2008

University exchange year. Scholarship granted by USFQ.

Publications

- A. Cevallos, F. Eisenbrand, and R. Zenklusen. *Local search for max-sum diversification*. To appear in SODA 2017.
- A. Cevallos, F. Eisenbrand, and R. Zenklusen. *Max-sum diversity via convex programming*. 32nd International Symposium on Computational Geometry–SoCG, 2016.
- M. Aprile, A. Cevallos, and Y. Faenza. *On vertices and facets of combinatorial 2-level polytopes*. 4th International Simposium on Combinatorial Optimization–ISCO, 2016.
- A. Cevallos, S. Fehr, R. Ostrovsky, and Y. Rabani. *Unconditionally-secure robust secret sharing with compact shares.* Advances in Cryptology–EUROCRYPT, 2012.

Teaching Experience

École Polythecnique Fédérale de Lausanne, Switzerland

September 2012 – present

Teaching assistant. Mathématiques générales, optimisation discrète, algèbre linéaire avancée I & II, computer algebra.

Project supervisor. Maximization of submodular and diversity functions, implementation of the RSA cryptosystem, optimization of a hydropower plant, the orienteering problem on planar graphs.

Universidad San Francisco de Quito, Ecuador

January – May 2012

Lecturer. Cálculo, álgebra linear.

August 2008 – May 2009

Teaching Assistant. Cálculo, ecuaciones diferenciales.

Relevant Skills

Language skills. Native Spanish, fluent English, proficient French, conversant Italian.

Software skills. LaTeX, Matlab, C++, Java, Python.